M. Sc. Engineering Thesis

DYNAMIC ADAPTIVE CONTENT DELIVERY USING GENETIC ALGORITHM

by

MOHAMMAD MAKSUD HOSSAIN

Submitted to

Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering.

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka - 1000, Bangladesh

September 2009

The thesis titled "DYNAMIC ADAPTIVE CONTENT DELIVERY USING GENETIC ALGORITHM" submitted by Mohammad Maksud Hossain, Student No: 040405035P, Session: April 2004, to the department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of **Master of Science in Computer Science and Engineering** and approved as to its style and contents. Examination held on September 30, 2009.

Board of Examiners

Dr. Md. Mostofa Akbar Associate Professor Department of CSE, BUET, Dhaka.	Chairman (Supervisor)
Dr. Md. Saidur Rahman Professor and Head Department of CSE, BUET, Dhaka.	Member (Ex-officio)
Dr. M. Kaykobad Professor Department of CSE, BUET, Dhaka.	Member
Dr. Mohammad Sohel Rahman Assistant Professor Department of CSE, BUET, Dhaka.	Member
Dr. Sayed Faisal Hasan Assistant Professor Department of CSE, University of Dhaka	Member (External)

Candidate's Declaration

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Mohammad Maksud Hossain

To the Almighty

To my family

To my first new born son Jarif Tahmid Hossain (Raem)

ACKNOWLEDGEMENT

All praise to Allah, the most benevolent and the Almighty, for His boundless grace in successful completion of this thesis.

I would like to express my sincere respect and gratitude to my thesis supervisor, Dr. Md. Mostofa Akbar, Associate Professor of the Department of Computer Science and Engineering (CSE), Bangladesh University of Engineering and Technology (BUET), Dhaka, for his thoughtful suggestions, constant guidance and encouragement throughout the progress of this research work. He was always more confident than I was about my being able to complete this thesis. The ever-interesting part of working with him was: whenever I almost devised a solution, he was there to find a small new patch to the original problem; although the solution part was hardly ever patchable and I had to endure the fun of doing re-research.

ABSTRACT

In this thesis a new framework for dynamic adaptive content delivery is presented, which is suitable for diversified mobile devices. The proposed framework can dynamically adapt itself for diversified web contents available at the numerous content delivery sites around the globe. Our approach differs form previous works as it is not only based on adapting single type of content in static predefined way, but also capable to adapt multiple types of content dynamically on population changes. Every type of content is different from the others by different attributes they have and even different attribute values.

The adaptive content delivery problem considered here is an NP hard problem with exponential time complexity. We introduce Genetic Algorithm for the dynamic learning at the initial phase and at the time when the environment changes due to introduction of new clients. In the proposed framework the Dynamic Content Adaptation has been established by using Genetic Algorithm to identify the Majority Supported Capability Set at the learning engine in the learning phase using the information from client historical base. The current client environment can be easily identified using the client historical base information and the change in the client environment can also be identified in real-time.

We show that Dynamic Adaptive Content Delivery (DACD) can minimize the limitations of existing content adaptation techniques and also add new scope to the current research directions. The framework is verified using real telecom network data with help of WURFL repository. Results indicate that the DACD framework can efficiently identify the MSCS which can deliver content that closely matches the capability of the population and reduces the variety of content significantly.

The proposed framework has been compared with the existing research on content adaptation. It is found that the solution of the proposed framework performs better in terms of real-time content adaptation capability and maximization of server resource utilization.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

API: Application Programming Interface

CHTML: Compact Hyper Text Markup Language

DACD: Dynamic Adaptive Content Delivery

GA: Genetic Algorithm

HTML: Hyper Text Markup Language

JSP: Java Server Page

MSCS: Majority Supported Capability Set

PHP: Hypertext Pre Processor

RDF: Resource Description Framework

SMIL: Synchronized Multimedia Integration Language

TS: Test Set

WALL: Wireless Abstraction Library

WAP: Wireless Access Protocol

WCML: Web Composition Markup Language

WML: Wireless Markup Language

WS: Working Set

WURFL: Wireless Universal Resource File.

XHTML: Extensible Hyper Text Markup Language

XML: Extensible Markup Language

XSL: Extensible Style Sheet Language

List of Symbols

$A_j = j^{th}$ attribute of the attribute set.

$c_i = i^{th}$ member of Working Set.

$c_x = x^{th}$ member of *WS* (Working Set)

comply= Linear function which returns 1 when the first argument complied with the second.

f_{min} = Minimum fitness value that can be considered for learning.

fitness = Function finds the fitness of its parameter against the Test Set.

I_{max} = Maximum number of iteration allowed before terminate.

m = Total numbers of attributes in the Attribute Set

$M_m = m^{th}$ member of MSCS.

n = Total size of Test Set

S_{min} = Minimum acceptable MSCS size.

$TS_i = i^{th}$ member of *TS* (Test Set)

Chapter 1. Introduction

1.1. Motivation

The enormous expansion of the use of internet as the most important information source in desktop computers as well as in recent devices like PDA, handheld computers, PocketPCs and SmartPhones has increased the diversity and heterogeneity of content to meet the device capability. The content includes media like images [2, 4, 6], audio, video [5], application or even text [4] which vary from device to device. People desire to access Internet contents anywhere, anytime with any devices but the increasing diversity and heterogeneity of contents, client devices with individual preferences make "*one content fits all needs*" impossible. All of the content may not be suitable for every device and sometimes that becomes compatible for limited number of device. Adaptation of the content is required for effective delivery in heterogeneous environment. In a typical content delivery network clients of diversified capabilities request content from content server and the server delivers the contents without checking the capability of the client, as a result in many cases the content may not be usable by the client due to capability limitation. The following issues regarding content delivery systems motivate the investigation of new framework:

- Identifying client capability for the content support to deliver the appropriate content.
- Compilation of contents according to the client's capability.
- Limitations of resources in the server to pre compile or comply on the fly all combinational types of contents.
- The change of the content repository and content delivery strategy with the change of device environment.

Figure 1-1 depicts a typical content delivery system.

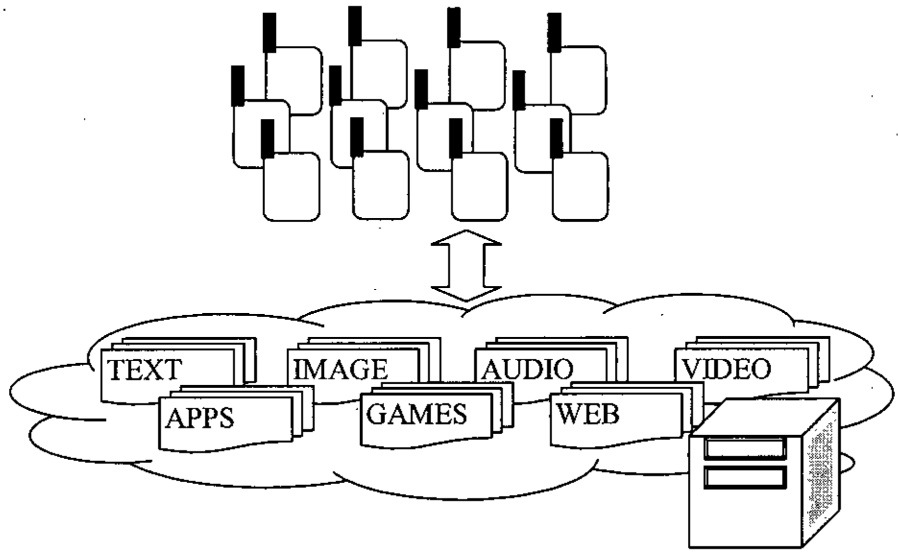


Figure 1-1 A typical content delivery system.

1.2. Problem Definition

In this thesis a new framework is proposed that addresses the problem of adaptation of requested contents to the diversified clients in a content delivery system. The following points describe the problem addressed by the framework.

- *Contents residing in content servers*: A variety of contents can be hosted by content server. The content can have different attributes that varies from client to client in case of the content support capability of the client.
- *Requests of the clients:* The clients request for contents which are stored in the content server. The server without verifying client capability for the requested content serve the client request by sending the content as the response against the request over the network. The content server will not reject to serve a client even if the content is not capable to satisfy the content support capability of that client.

The possible ways to address these problems are:

- *Adaptation of contents*: Multiple alternative versions of content can either be pre converted or compiled on the fly and served according to the client capabilities to a client to provide more reliable service which will increase the client's satisfaction.

- *Maximization of content support*: The adaptation of contents at content servers is done in such a way so as to maximize the majority support of all the clients' capabilities along with the consideration of server's resource limitation.
- *Dynamic adaptation on environment change*: The environment of the system changes with the inclusion of new clients and removal of old clients along with the change of content support the clients have. The content server is configured to adapt with this changing environment along with the contents it stores.

1.3. Objective and Scope of the Thesis

The objective of this thesis is to support effective and optimized delivery of content in heterogeneous environment by implementing some artificial intelligence technique to identify optimized set of Majority Supported Contents and implement a Dynamic Adaptive Content Delivery Framework.

The dynamic adaptive content delivery framework presented in this thesis is solved by using Genetic Algorithm [13]. The Genetic Algorithm [13] will not always provide an optimal solution in all scenarios. However the solution, whether it is optimal or not will be generated in polynomial time, the time complexity of the algorithm does not even depend on the number of clients or contents in the environment.

The framework for an exact solution is out of scope of the thesis. The exact solution is an NP hard problem and it is practically impossible to solve the problem for larger data sets. That is why the proposed algorithm is compared with infopyramid [1] based solution as well as brute force technique. Also to determine the performance of the proposed solution, the proposed framework has been simulated with lots of changing parameters of Genetic Algorithm [13]. A detailed study of the performance of the framework has been discussed later in this thesis.

1.4. Outline

The remaining part of the thesis is organized as follows.

Chapter 2 discusses about the content and content capabilities. It also gives a brief description on Genetic Algorithm [13]. An overview of the content adaptation problem, its applications, algorithms and current works along with limitations are also presented in this chapter.

The proposed new framework is described in Chapter 3, along with its mathematical formulation. Section 3.3 is devoted to the description of the proposed dynamic adaptive content delivery framework. A worst case complexity analysis of the framework will be found in Section 3.7.

In Chapter 4, an analysis of the new algorithm is presented compared to the existing adaptation techniques.

Chapter 5 concludes the thesis providing some directions for further research in this field.

Chapter 2. Literature Review

2.1. Multimedia Content

Every website consists of several type of multimedia elements including image, audio, text, video, complex object like flash document, java script, java applet and many more. It is also represented by some structural markup language like HTML, WML, xHTML. Some other auxiliary component such as document header is also available with it. All these elements are Web Content that the client device has to interpret properly to display some meaningful information to user. .

Every Web Content has some attribute attached with it, which defines the actual representation of the content. For example image can have the attributes like height, width, color-depth, gray scale, type etc. The following table represents some contents and their attributes with possible values.

Table 2-1 Content and attributes.

Content	Attribute	Values
Image	Type	Wbmp, bmp, epoc_bmp, ota_bitmap, gif, gif_animated, jpg, png, tiff, svgt_1_1, svgt_1_1_plus
	Width	96, 120, 128, 160, 320, 600, ...
	Height	96, 120, 128, 160, 320, 600, ...
	Size limit	100K, 200K, 500K, 1M, 10M, ...
	Gray Scale	2/4 bit
	Color Depth	16, 256, 65K, ...
	Resizable	Yes, No
Audio	Type	wav, mmf, smf, mld, midi, sp_midi, rmf, xmf, compactmidi, digiplug, nokia_ringtone, imelody, au, amr, awb, aac, mp3, voices, qcelp, evrc.
	Tone	Monophonic, Poly-phonic, voice

	Time	30sec, 1min, 5min, ...
	Size limit	100K, 200K, 500K, 1M, 10M, ...
	Chord	16 bit, 64 bit
Text	Type	SMS, MMS, WAP Push, Markup
	Encoding	utf8, ascii, iso8859

2.2. Client Capability

With diversified growth of modern technology along with mobile device, lots of varieties of client devices are currently available. The capability of these devices varies from device to device. For example if we consider the images PC can support many type of images with different dimension, Smart Phone may be limited to few type with fixed dimension, text based terminal or mobiles does not support any picture. A comparative representation can be found in Table 2-2.

Table 2-2 Content Capability Support.

Device	Image	Height	Width	Type	Color	Size
PC	Yes	Any	Any	Any	Any	Any
PDA	Yes	Any	Any	Limited	Max 16 bit	Any
Handheld	Yes	Any	Any	Limited	Max 16 bit	Any
Nokia 6030	Yes	Max 800	Max 800	Jpg, gif	Max 8 bit	1 M
Moto C-80	No	N/A	N/A	N/A	1 bit	N/A
Nokia N70	Yes	Any	Any	Limited	Max 16 bit	10M

2.3. Client Capability Identification

There is an open source project for identification of mobile device capabilities along with their Wireless Universal Resource File (WURFL) [14]. It is part of a FOSS (Free and Open Source Software) community effort focused on the problem of presenting content on the wide variety of wireless devices. The WURFL itself is an XML [18] configuration file which contains information about device capabilities and features for a variety of mobile devices. Device information is contributed by developers around the world and the WURFL is updated frequently reflecting new wireless devices coming in the market. This project is maintains a huge XML data file which contains the capability information for most of the mobile devices. WURFL usually stores the user agent string of the client's request header sent with the content request. The user agent string can be considered as a unique code to represent a client. WURFL maintains the capability information along with other metadata against each user agent string. In our framework we will use the WURFL data source for client capability identification.

The following example shows sample template of capability XML data in WURFL for a dummy client ClientX.

```
<device user_agent="ClientX-Mozilla/X.X (compatible; MSIE X.YY; Windows ZZ; Smartphone;
MMxNN)" fall_back="ms_mobile_browser_ver1" id="clientx_ver1">
        <group id="group_a">
                <capability name="capability_m" value="XXX"/>
                <capability name=" capability_n" value="YYY"/>
                <capability name=" capability_o" value="ZZZ"/>
        </group>
        <group id="display">
                <capability name=" capability_p" value="1234"/>
                <capability name=" capability_q" value="3.14"/>
                <capability name=" capability_r" value="AA"/>
                <capability name=" capability_s" value="BB"/>
                <capability name=" capability_t" value="CC"/>
                <capability name=" capability_u" value="XYZ"/>
        </group>
</device>
```

2.3.1. History of WURFL Development

At the end of 1999 the first WAP [20] phone was launched in Europe, followed by many others in the following months. By 2001 it became clear that WAP devices exhibited significant differences in the way they handled WAP content. The implication of this was that mobile developers found it difficult to support the increasing numbers of devices; the cost of application development and cost of testing made WAP development expensive as compared to web development. Eventually, some developers realized that they could leverage the open-source model for their efforts. Luca Passani and Andrea Trasatti joined forces to build a community around a shared repository of device capability information, which they named WURFL [14]. Over the years the project has gained followers and supporters from different geographical regions and with different backgrounds. The first basic API was in Perl. Java and PHP versions of the libraries appeared shortly afterward, soon followed by a better .NET Framework, Perl version, Ruby, and, more recently, Python, XSLT and C++. These API versions help developers to incorporate the WURFL support in their applications using their native programming languages.

2.3.2. The Problem of Device Fragmentation in Wireless Environment

Content written as HTML can be expected to be visible to most users of a web-based channel via one of the standard browsers like Internet Explorer, Mozilla Firefox, Safari, Opera, and so on. Software updates for desktop browsers are frequently made and widely distributed.

Unlike the desktop web-channel, there is a tremendous amount of fragmentation in the mobile device-channel. Markup can be WML, HTML, HDML, XHTML Mobile Profile, etc. In addition, unlike a standard desktop web-channel, a wireless-device channel will vary on screen size, ability to support client side scripting, ability to support various image formats, and even color. As the markup is generally sent directly to the phone, there is no opportunity for a central server to "fix" or adapt to browser limitations or defects. Software updates for mobile browsers are rare.

2.3.3. Solution Approaches

There have been several approaches to this problem, including developing very primitive content and hoping it works on a variety of devices, limiting support to a small subset of devices or bypassing the browser solution altogether and developing a Java ME or BREW client application.

WURFL [14] solves this by allowing development of content pages using abstractions of page elements (buttons, links and textboxes for example). At run time, these are converted to the appropriate, specific markup types for each device. In addition, the developer can specify other content decisions be made at runtime based on device specific capabilities and features (which are all in the WURFL).

2.3.4. WALL, Wireless Abstraction Library

WALL (Wireless Abstraction Library) is a JSP tag library that lets a developer to author mobile pages similar to plain HTML without the thinking of the device capabilities and final output format. It can deliver WML, C-HTML and XHTML based on the Mobile Profile of the device from which the HTTP request originates, depending on the actual capabilities of the device itself. Device capabilities are queried dynamically using the WURFL [14] API.

2.4. Genetic Algorithm

A genetic algorithm (GA) [13] is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms (EA) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover.

2.4.1. Genetic Algorithm Methodology

Genetic algorithms [13] are implemented in a computer simulation. In GA a population of abstract representations (called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) is used. It is an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.

The evolution usually starts from a population of randomly generated individuals and continues in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

Genetic algorithms find application in bioinformatics, phylogenetics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields. A typical genetic algorithm requires a *genetic representation* of the solution domain, and a *fitness function* to evaluate the solution domain.

A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the knapsack problem one wants to maximize the total value of objects that can be put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution is the sum of values of all objects in the knapsack if the representation is valid or 0 otherwise. In some problems, it is hard or even impossible to define the fitness expression; in these cases, interactive genetic algorithms are used.

Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, and then improve it through repetitive application of mutation, crossover, and inversion and selection operators.

2.4.2. Initialization

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space). Occasionally, the solutions may be biased towards the areas where optimal solutions are likely to be found.

2.4.3. Selection

Selection is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (recombination or crossover). There are several generic selection algorithms, such as tournament selection and fitness proportionate selection (also known as roulette-wheel selection). The latter may be implemented as follows:

- The fitness function is evaluated for each individual, providing fitness values, which are then normalized. Normalization means multiplying the fitness value of each individual by a fixed number, so that the sum of all fitness values equals 1.
- The population is sorted by descending fitness values.
- Accumulated normalized fitness values are computed (the accumulated fitness value of an individual is the sum of its own fitness value plus the fitness values of all the previous individuals). The accumulated fitness of the last

individual should of course be 1 (otherwise something went wrong in the normalization step!).

- A random number R between 0 and 1 is chosen.
- The selected individual is the first one whose accumulated normalized value is greater than R.

There are other selection algorithms that do not consider all individuals for selection, but only those with a fitness value that is higher than a given (arbitrary) constant. Other algorithms select individuals from a restricted pool where only a certain percentage of the individuals are allowed, based on fitness value.

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods evaluate the fitness of each solution and preferentially select the best solutions. Other methods evaluate only a random sample of the population, as this process may be very time-consuming.

Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions. That is why roulette wheel selection and tournament selection algorithm is popular and well studied.

2.4.4. Reproduction

The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation.

2.4.4.1. Crossover

In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based.

Many crossover techniques exist for organisms which use different data structures to store themselves.

One-point crossover: A single crossover point on both parents' organism strings is selected. All data beyond that point in either organism string is swapped between the two parent organisms. The resulting organisms are the children. The one point crossover has been shown in Figure 2-1.

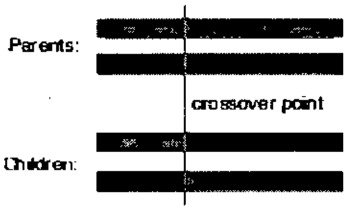


Figure 2-1 One point crossover.

Two-point crossover: Two-point crossover calls for two points to be selected on the parent organism strings. Everything between the two points is swapped between the parent organisms, rendering two child organisms.

Figure 2-2 Two point crossover.

Cut and splice: Another crossover variant, the "cut and splice" approach, results in a change in length of the children strings. The reason for this difference is that each parent string has a separate choice of crossover point.

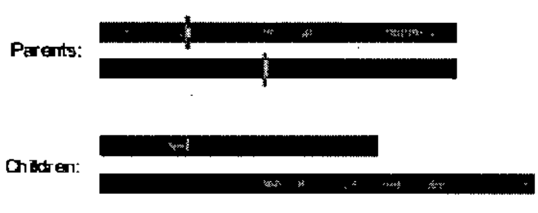


Figure 2-3 Cut and Splice crossover.

Uniform Crossover and Half Uniform Crossover: In both these schemes the two parents are combined to produce two new offspring.

In the uniform crossover scheme (UX) individual bits in the string are compared between two parents. The bits are swapped with a fixed probability, typically 0.5.

In the half uniform crossover scheme (HUX), exactly half of the non matching bits are swapped. Thus the Hamming distance (the number of differing bits) is calculated at first. Half of the hamming distance will provide the number of bits that need to be swapped between the two parents.

Crossover for Ordered Chromosomes: Depending on how the chromosome represents the solution, a direct swap may not be possible. One such case is when the chromosome is an ordered list, such as an ordered list the cities to be travelled for the traveling salesman problem. A crossover point is selected on the parents. Since the chromosome is an ordered list, a direct swap would introduce duplicates and remove necessary candidates from the list. Instead, the chromosome up to the crossover point is retained for each parent. The information after the crossover point is ordered as it is ordered in the other parent. For example, if our two parents are ABCDEFGHI and IGAHFDBEC and our crossover point is after the fourth character, then the resulting children would be ABCDIGHFE and IGAHBCDEF.

2.4.4.2. Mutation

In genetic algorithms, mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. It is analogous to biological mutation. The classic example of a mutation operator involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified.

The purpose of mutation in GAs is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. This reasoning also explains the fact that most GA systems avoid only taking the fittest of the population in generating the next but rather a random (or semi-random) selection with a weighting toward those that are fitter.

2.4.4.3. Reproduction Procedure

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above mentioned methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each child, and the process continues until a new population of solutions of appropriate size is generated. Although reproduction methods that are based on the use of two parents are more "biology inspired", recent researches (Islam Abou El Ata 2006) [citation needed] suggested more than two "parents" are better to be used to reproduce a good quality chromosome.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will be increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

2.4.5. Termination

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies minimum criteria
- A fixed number of generations is reached
- Allocated budget (computation time/money) is reached
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- Combinations of the above

2.4.6. Simple generational genetic algorithm pseudocode

The pseudo code of a simple generation genetic algorithm is given below -

- Choose the initial population of individuals
- Evaluate the fitness of each individual in that population
- Repeat on this generation until termination: (time limit, sufficient fitness achieved, etc.)
    - Select the best-fit individuals for reproduction
    - Breed new individuals through crossover and mutation operations to give birth to offspring
    - Evaluate the individual fitness of new individuals
    - Replace least-fit population with new individuals

2.5. Adaptive Content Delivery

Adaptive content delivery is a system that transforms Web content and delivery schemes according to viewers' heterogeneous and changing conditions to enable universal access. The goal of adaptive content delivery is to take into account these heterogeneous and changing conditions and provide the best information accessibility and perceived quality of service over the Internet. Ultimately, adaptive content delivery aims at universal access to multimedia information in a heterogeneous network environment, by accommodating the special needs of users and the constraints of client devices and network characteristics. In other word, the adaptive content delivery effort is to provide the necessary Internet infrastructure to allow users to access any information over any network from anywhere through any type of client device.

Adaptive content delivery has beneficial business implications beyond just reaching a wider audience for Web content. One of the main benefits is to decrease the Web access time for users. In a user survey conducted by Georgia Technical Institute's Graphics, Visualization, and Usability Group, 53% of respondents reported that they had left a web site while searching for product information simply because the site was too slow.

Adapting content to have more aesthetic appearance on the user device or allowing the user to have wider access may encourage the user to appreciate the site more. This can also result in higher hit rates and return rates, implying higher sales for e-commerce sites and higher advertising revenues. In order to provide adaptive content delivery over heterogeneous network environments, many technologies from different aspects of the delivery environment need to be developed and integrated. These technologies include

- Media processing and analysis algorithms to support content adaptation.
- A mechanism set for detecting the software and hardware capabilities of a client device.
- A way to effectively measure the characteristics of the current network connection between a client and a server.
- A standard approach for defining user preferences and a mechanism for tracking them from session to session.
- Decision rules on when and how to perform a particular content adaptation process based on various conditions.

2.5.1. Current Works on Adaptive Content Delivery

Numerous organizations, institutes and commercial companies have identified the issue of Web Content adaptation under heterogeneous client environment and lots of research works have been done extensively on Web Content Adaptation. Most of the adaptation technique focused on a single content adaptation such as Image [2, 4, 6, 26, 32], video [5] or text [4]. Some has focused on the layout formatting [4, 6, 7, 19, 28, 30] others provide adaptation based on special application on client side [8]. The adaptation engine can be deployed in server or in the client or in both the machines. All these deployment alternatives have some merits & demerits. Negotiation techniques like The HTTP/1.1 [9] and CC/PP [10] are proposed for requesting preferred version of content along with its user agent information. Some standard protocol are proposed by the W3C, the IETF and others which includes web techniques of Markup Language like SMIL [11], XML [18] along with XSL [22], WML or CHTML [21] and Web Composition Markup Language [12].

Current research of adaptation identified various way of adaptation like Information abstraction, Modality transform, Data transcoding, Data prioritization, Purpose classification and Pre-fetching & caching [3].

Commercial products and research in Web Content Adaptation include IBM Transcoding proxy, Intel QuickWeb [16], ProxiNet [17], Spyglass Prism [15], Smart Client, OnLineAnywhere, Odyssey, Digestor, Mobiware, TranSend, Bickmore and Schilit [4], AvantGo and so on. All these adaptation frameworks mostly focus on image based or limited adaptation.

Image adaptation mostly depends on some predefined static transformation like resizing, color depth changing, converting gray scale, alter format, changing dimension, cropping etc. Windows CE device has the capability to change color depth [32]. Content delivery system such as TranSend project, Spyglass Prism (TM) [15], Bickmore and Schilit [4], and QuickWeb (TM) [16] use some combination of compression, resizing, scaling by predefined scaling factors and changing color depth of image to meet client capability. Video adaptation through variation of bandwidth usage, screen size, frame-rates, encodings or compression scheme has been addressed. This type of adaptation does not consider multiple components such as video with image, audio, key frame and caption. Text adaptation [4] includes format conversion like postscript to HTML. Bickmore and Schilit's work mainly focused on textual, specifically HTML content. The Spyglass Prism (TM) performs some HTML filtering and modification, such as removal of Java and JavaScript, and conversion of tables to lists.

In a server-based architecture, the server is responsible for discovering the client capabilities and decides the best adaptation strategy. It supports both dynamic (on-the-fly) and static (off-line) content adaptation and provides more author control in heterogeneous environments. The drawbacks include geographical bound copyright infringement, additional computational load and resource consumption on the server.

Proxy based adaptation is the most common; in this adaptation proxy receive request from client and then makes the request to the server on behalf of the client. The proxy then intercepts the reply from the server, decides and performs the adaptation. Lastly it sends the transformed content back to the client. Proxy based content adaptation is often termed "**transcoding**". In the TranSend project proxy transcodes Web content on the fly using adaptation. "**Refinement**" mechanism [4] can be used to request the original version of the content. Proxinet, AvantGo provides a proxy which customizes content for a special browser on the PalmPilot. Bickmore and Schilit propose a proxy which use a number of heuristics and a planner to perform outlining and identification of the content to fit client capability. The Spyglass Prism (TM), Intel QuickWeb uses transcoding proxy which adapts image & HTML. Proxy based adaptation system performs best in case of speed where the link between client and proxy is slow but link between server and proxy is good. It also caches the contents, supports geographical distribution for faster access and both client and server need not to be modified. There is no customization for different client devices. The sole purpose of the service is to improve response times for PCs connected over slow links such as modems. This improvement in response time is even more significant when the adaptation is performed at the Web servers, where the transcoded content can be pre-cached.

The main problem with most of the proxy based system is that, they usually deploy static, predefine, ad-hoc process for adaptation with only a single or limited type of content like scaling image with some predefine size, which support only a few devices and failed to dynamically adapt variety of device capability for different content type. The support of modality change (Audio to Text) [1] is also absent in these system. As everyday new complex multimedia contents are coming in picture static adaptation will not be sufficient to support Web Content adaptation using proxy based system.

Client Side Adaptation [25] can be found at Windows-CE devices which change color-depth of images. This may not be much beneficial due to low network bandwidth which results in slow access to pages with rich multimedia content and less computational power makes content adaptation at the device slow, or even impossible.

New protocols for markup language come with support for content adaptation. The Synchronized Multimedia Integration Language (SMIL) [11] is a successful content adaptation markup language that enables the synchronized delivery of multiple video streams, audio streams, and images. The Extensible Markup Language (XML) can be used to describe the logical representation of data which can be used to serve contents based on heterogeneous client capabilities. The Extensible Style Sheet Language (XSL) [22] can be utilized to convert the XML data into an appropriate representation. The Web Composition Markup Language (WCML) [12] describes an XML vocabulary for Web Composition that allows the definition of web contents, properties, and relationships between these contents.

Negotiation between the client & content provider (Server/Proxy) has been addressed in some negotiation protocols. Both the HTTP/1.1 [9] and the CC/PP [10] uses some mechanisms for the client to convey its preferred version of content and user agent information along with request. HTTP [23] header convey the agent and preference information in HTTP/1.1 content negotiation whereas in CC/PP client capabilities and user preferences sent by the client along with a HTTP request as a collection of URIs and Resource Description Framework (RDF) text. URIs point to an RDF document containing client's capabilities details. RDF provides a way to express "metadata" for a Web document.

The main problem with most of the server or proxy based system is that, they usually deploy static, predefined, ad-hoc process for adaptation with only a single or limited type of contents. The support of modality change [1] (Audio to Text) is also absent. As everyday new complex multimedia contents are coming in picture static adaptation will not be sufficient to support web content adaptation. Client side adaptation [8] may not be very much beneficial always for the following reasons:

- Browsing high quality content requires high bandwidth and it results slow access to the pages with rich multimedia content. But the users enjoy low quality adapted content.
- Less computational power makes content adaptation at the device slow, or even impossible.

Generalized Dynamic adaptation for any supported media is not a widely researched topic. The adaptation technique presented in [1] is the only related research found through extensive search. Recently research has been done on dynamic selection of video content strategy [31]. In the research Genetic Algorithm has been used together with Pareto Optimality in the process of selection of a suitable video content adaptation strategy. The article refines the process of selection of an optimal strategy by taking into account the distribution alongside user preferences, video content characteristics and usage history. In order to make the refined process dynamic, it pursues its implementation using Self-Organizing Neural Networks.

2.6. Chapter Summary

This chapter briefly focused on the content, content capability, content adaptation related definition and solutions. It also shows the current research trend towards the content adaptation along with the limitations. A details discussion on the Genetic Algorithm along with its working procedure has also been discussed in this chapter. In the next chapter we will describe details of the implementation and working procedure of the proposed framework.

Chapter 3. Proposed New Framework

This chapter describes the proposed new dynamic adaptive content delivery framework. A mathematical model of the core phase of the proposed system has also been presented here. The learning technique employed to solve the content delivery problem is a well known artificial intelligence algorithm called Genetic Algorithm [13] that effectively identify the majority support from a given population. The use of Genetic Algorithm in solving the dynamic adaptive content delivery is not straight forward as the problem should be formulated in such a way so that Genetic Algorithm can be applied on it. The working procedures along with used algorithms for GA have been presented in this chapter. The chapter concludes by presenting a complexity analysis of the proposed framework.

3.1. The Dynamic Adaptive Content Delivery System

The proposed framework is for an adaptive content delivery system where different contents are stored in a content server and contents can have multiple versions by the changed value of their several attributes according to the clients need. Upon the receipt of any client request for a particular content the content server has to identify the client capability for the content and then prepare the content according to the capability and lastly respond the client with the adapted version of the content. There are several ways to deliver the adaptive content. For example sever can preserve a master copy of the content and up on request it can compile new adaptive content on the fly but server with heavy load may be unable to handle this due to the huge computational power required for the conversion task. Other alternative is to pre compute all the possible version but is will require huge storage which will become impractical in some extent and to identify all possible versions is also very complex. The environment of the clients and the demand of content versions change very often. That is why dynamic adaptation technique becomes necessary instead of existing static content adaptation techniques.

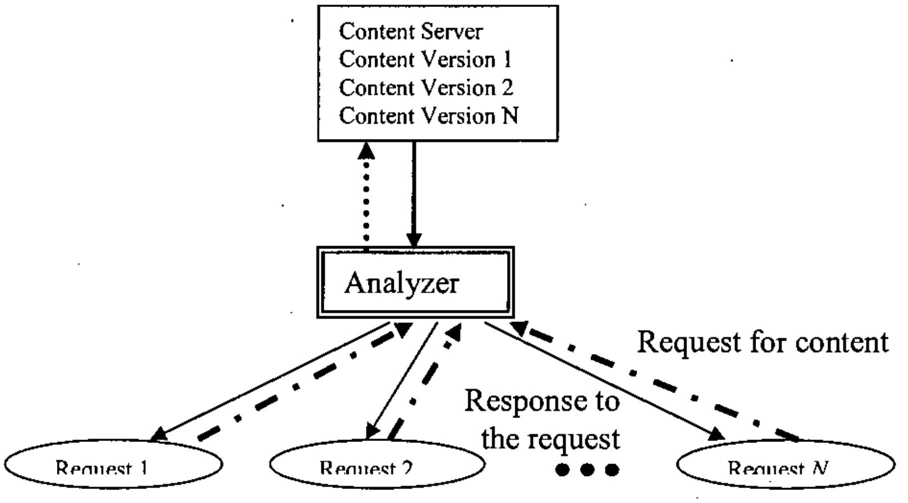


Figure 3-1 A typical architecture of the adaptive content delivery system.

3.1.1. Working Principle of the Proposed Dynamic Adaptive Content Delivery System

There are two phases, learning and delivery. The following points demonstrate the working principle of the proposed dynamic adaptive content delivery system by phases:

Working principle of learning:

- The content request made by the client is usually some HTTP request over the internet. In the standard HTTP request some basic information is passed between server and client besides the basic content request. The information is passed by the request header for client and response header by the server.
- Client sends its user agent information in the request header which can be used to identify the client capability for a particular type of content. This capability identification using user agent information can be accomplished by WURFL [14] repository which maintains almost every client's information.
- The framework first collects the client list which is available in the current environment by logging the requests from different clients for contents during a particular period of time.

- From the current client list the current capability of the clients for the contents are extracted.
- The capabilities are then passed to the learning engine to find out the Majority Supported Capability Set, i.e. the capability set which has gained the maximum support from the current universe.
- The server then compiles and preserves the contents according to the learned capability set for the future content delivery.

Working principle of delivery:

- When the content server gets a new request then it identifies the client's capability and then matches with the contents preserved in the server by the learning and extract the best matching content for it. This task is done by the Analyzer engine. The content is then delivered to the client accordingly.
- The performance of the system is monitored every time and if the performance falls below a certain level or a particular time span expires then the learning is applied again.

3.1.2. Assumptions of the Framework

The following points are assumed to simplify the framework in the context of real or practical situation.

- The framework will support the majority of the clients not the total clients maintaining the limitation of the server.
- The environment change is less frequent and not abrupt by the inclusion of new client or removal of old clients.
- The learning will remain usable for a long period compared to the time taken for the learning phase.
- Client can support some content that does not give the full match with its capability but gives a partial match.

3.2. Overall Architecture of the Framework

The overall architecture and the connection between the components are depicted in the Figure 3-2 below. The brief descriptions of each component are followed by the overall architecture figure.

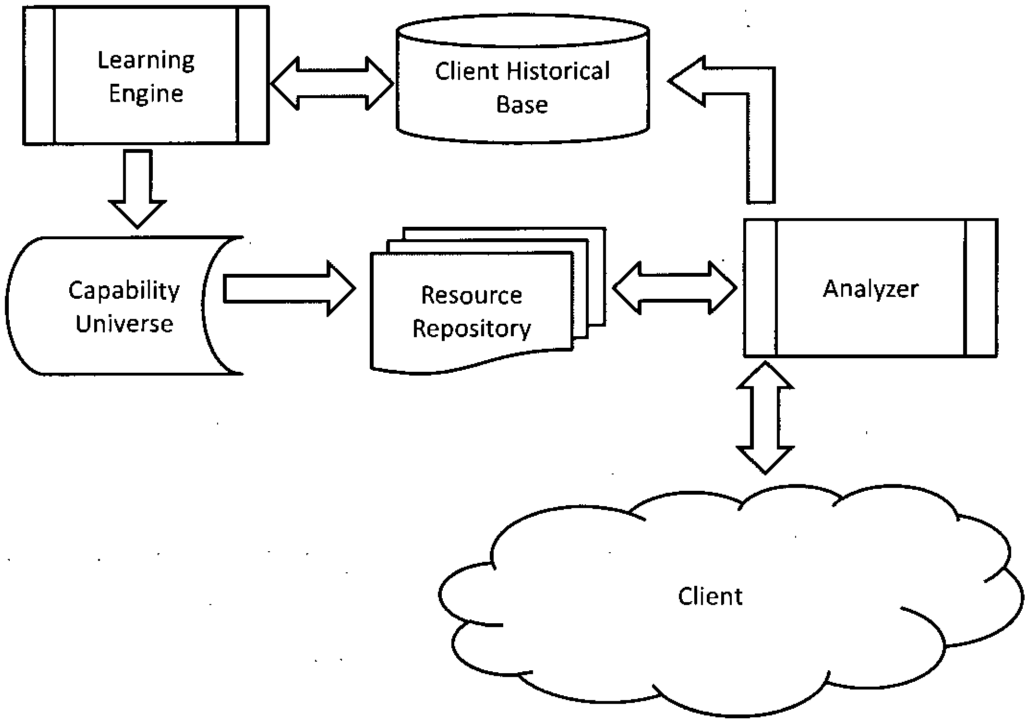


Figure 3-2 Architecture of dynamic adaptive content delivery framework.

- *Learning Engine:* The learning engine is the core mechanism to learn about the Majority Supported Capability Set (MSCS) of the current universe using Genetic Algorithm [13].
- *Client:* Clients are the external devices who send request to the content server for some content or service and get response accordingly.
- *Client Historical Base:* Repository of the entire client list that has accessed the system. It works as the information base for the learning engine.
- *Capability Universe:* Preserves the MSCS of latest learning to serve the decision engine.
- *Resource Repository:* Pre compiled content storage as per the MSCS to help faster content delivery and optimum resource utilization.

- *Analyzer:* Decision engine to find out the best suitable content among MSCS which matches best for the requesting client.

3.2.1. The Learning Engine

The learning engine is the key component of the proposed framework. The outcome from the learning engine is used for the content adaptation. The learning engine uses Genetic Algorithm [13]. The overview of the working procedure of the learning engine is given below -

- At the beginning two random sets of clients are selected from Client Historical Base, the first set is considered as the Test Set and the second is considered as working set.
- Test Set is the set of client which remains unchanged in the entire learning phase. It is used to evaluate the current working set i.e. each member of current working set is evaluated against all the member of Test Set for the capability matching.
- Working Set is the set which goes through several generations. In each generation breeding among Working Set members produce new generation. The members of Working Set that gets the higher evaluation from the Test Set members will be more probable to participate in breeding than others.
- Working set goes through initial breeding and generates the first generation. Then consecutive breeding goes on till some stopping criterion matches.
- Every new generation is evaluated against the Test Set in every phase.
- The candidate of next breeding is extracted from the evaluation by considering the following criteria -
    - Majority supported new generation members are more probable candidate of next breed.
    - Candidate selection with majority support helps the learning to converge towards the MSCS.
- The learning can be stopped if any of the flowing condition or some combination of these becomes true

- Some predefined goal is reached for example the Working Set achieved some predefined evaluation from the Test Set.
- Specific time or number of generation or iteration has been crossed.

▪ The Majority Supported Capability Set from the current universe is returned when the learning is finished.

3.2.2. The Analyzer

The analyzer is the second key component of the proposed framework. After the learning, Learning Engine Concludes with some MSCS. The framework then compiles all the existing content available at the content server according to the MSCS and preserves it in the Resource Repository. Then the task of Analyzer is to deliver the optimal content to the requesting client. The working procedure of the Analyzer is given below -

- The Analyzer gets the request from the client and identifies the client capabilities.
- It then evaluates the capabilities of the client for each member of the MSCS and gets the evaluation result.
- The Analyzer then decides the best match of requesting content by the evaluation result.
- It then extracts the content from the resource repository and delivers to the client as the response to the client request.

3.3. Mapping of the Learning Engine to Genetic Algorithm.

A heuristic based on the genetic algorithm [13] has been presented to implement the learning of the proposed framework of dynamic adaptive content delivery problem in a content delivery system. If there were no limitation of resources of the servers then all the content requests could have been served with the highest possible capability match, which is not a possible case in reality. Thus the delivery of content must not overuse the amount of available resources at the servers. That is, the consumption of resources must not exceed the server capacity, which forms the basic idea of using a genetic algorithm [13] in the proposed scenario. The key challenge here is to identify the majority supported content by using the genetic algorithm [13] and transform the client capability using WURFL [14] to make it appropriate to the genetic algorithm [13]. The client capabilities here are modified into bit string so those can be used with the genetic algorithm and the outcome from the algorithm can again converted to the client capabilities.

3.3.1. Mapping of the Client Capabilities to Genetic Algorithm

Every web content consists with some attributes. The client capability is restricted to support some of the attributes or some specific values of the attributes. The attributes of the contents can have discrete or continuous values. So a particular attribute can be represented as A_j. Where $A_j = (X, Y)$ or $A_j = \{X_1, X_2, X_3 ... Y\}$. In first case A_j contains some continuous values ranging from X to Y. For the second case A_j contains some discrete values represented by the set.

On the other hand in Genetic Algorithm [13] each candidate can be identified by some set of genes associate with it. The gene is a set of binary bits. A particular gene can be represented as $G_i = b_{i1}\ b_{i2}\ b_{i3}\ ...\ b_{in}$, where $b_{ix} = \{0, 1\}$ and G_i is a gene with n bit binary string.

The initial task is to convert these attributes into binary string of the gene. And after learning the binary gene string will be decoded back to the attribute again. In our proposed framework both the continuous value attribute and discrete value attributes are converted to binary gene.

In case of continuous value attribute like $A_j = (X, Y)$ the range has been divided into some discrete ranges like $R = \{(X, X_1), (X_1, X_2), (X_2, X_3) \ldots (X_N, Y)\}$ with a range length of N. It requires $log_2(N)$ bits to represent these N ranges. Thus the length of the gene will be $log_2(N)$.

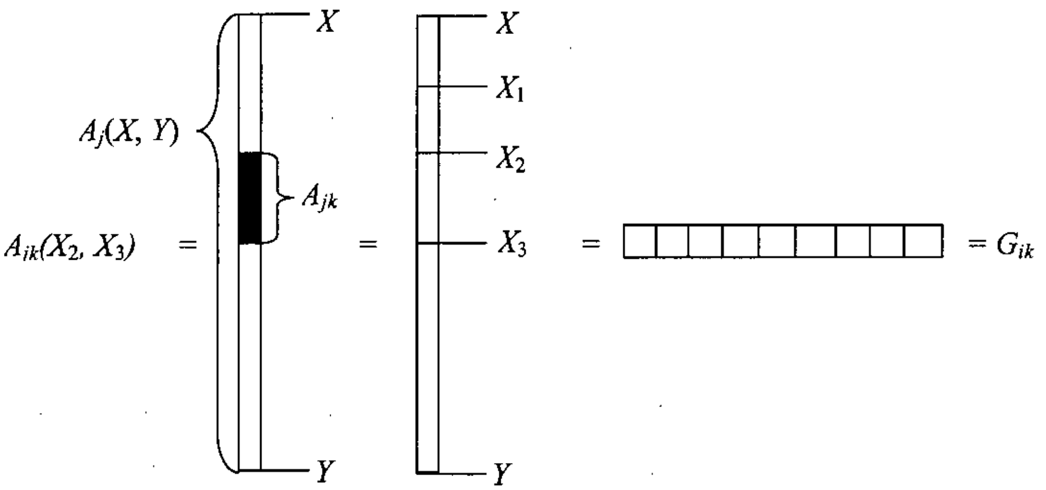


Figure 3-3 Continuous value attribute gene conversion.

For the discrete value attributes the gene has been generated by considering the size of the discrete value set. If the size of $A_j = \{X_1, X_2, X_3 \ldots Y\}$ is N then a gene of length will be $log_2(N)$. The corresponding genes of the discrete values will be represented by the binary representation of 0, 1, 2 ... N-1.

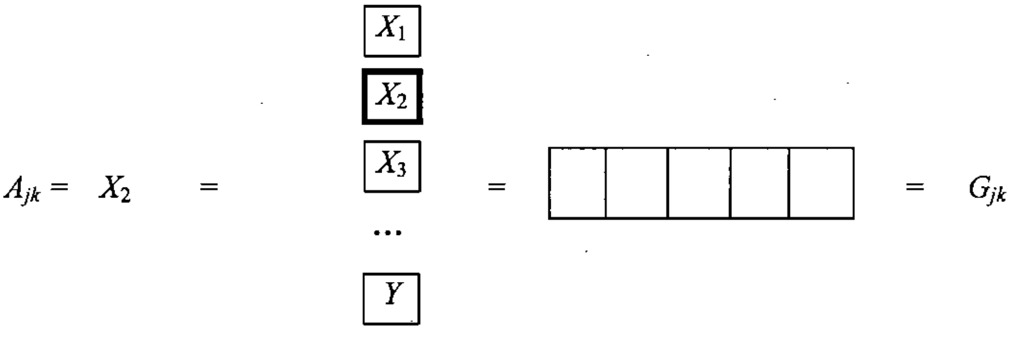


Figure 3-4 Discrete value attribute gene conversion.

3.3.2. Attribute Extraction from the Client Capability using WURFL

The WURFL [14] - Wireless Universal Resource File repository contains the updated capability information of every current mobile device available in the network. All the capabilities are preserved in XML [18] - Extendable Markup Language format. All the clients are identified using the USER AGENT information sent by the client's request header along with every content request. The USER AGENT is the unique information for a particular type of client. So the API of the WURFL [14] can be called with the parameter of USER AGENT to get the capability of the particular client.

The following XML [18] response can be found from WURFL [14] for the HTC S620 device -


```
<device user_agent="HTCS620-Mozilla/4.0 (compatible; MSIE 4.01; Windows CE; Smartphone; 320x240)" fall_back="ms_mobile_browser_ver1" id="htc_s620_ver1">
    <group id="product_info">
        <capability name="brand_name" value="HTC"/>
        <capability name="model_name" value="S620"/>
        <capability name="has_qwerty_keyboard" value="true"/>
    </group>
    <group id="wml_ui">
        <capability name="softkey_support" value="true"/>
    </group>
    <group id="markup">
        <capability name="html_wi_oma_xhtmlmp_1_0" value="true"/>
        <capability name="html_wi_w3_xhtmlbasic" value="true"/>
    </group>
    <group id="display">
        <capability name="resolution_width" value="320"/>
        <capability name="resolution_height" value="240"/>
        <capability name="max_image_width" value="320"/>
        <capability name="max_image_height" value="220"/>
        <capability name="rows" value="25"/>
        <capability name="columns" value="10"/>
    </group>
    <group id="image_format">
        <capability name="bmp" value="true"/>
        <capability name="colors" value="65536"/>
        <capability name="png" value="true"/>
    </group>
</device>
```

In the response XML [18] the capabilities are arranged in some groups. So by extracting all the capabilities of all the devices available in the Client Historical Base the Framework can identify the following key information -

- *The capability groups available among the current client universe:* All the groups are based on particular type of contents like image, sound etc. For example one particular capability group for image can be Image Format as shown in previous example of HTC S620. So if the adaptation is required on a particular type of content the framework can focus only on that group capability discarding other groups.
- *The specific capability or attribute:* There are several number of attributes or capabilities associated with a particular type of content. The WURFL [14] will return all the available attributes. Color, bmp, height, width are example of capabilities.
- *The values associated with each attribute:* A particular client will have a single or small set of particular attribute values and that is the capability of that client. But when the framework runs for every client it extracts all the possible values of a particular attribute. If some client dose not support a attribute the value of this attribute for that client should be considered as null i.e. not supported.

3.3.3. Example of Attribute to Gene Conversion

In the previous two sections we have shown the extraction of the client capabilities of the current universe and conversion of the attributes or capabilities into gene of genetic algorithm [13]. In this chapter we will focus on some example to accomplish this task.

For simplicity let us consider the image to be converted. Extracting the capabilities from the image group of the WURFL [14] capabilities we may get the following capabilities:

- Maximum Width
- Maximum Height

- Wallpaper Maximum Width
- Wallpaper Maximum Height
- Preferred Width
- Preferred Height
- Colors
- Resizing Ability
- Wallpaper Colors

The next thing is to figure out the possible values of the attributes. If we consider the wallpaper colors we may get the following values along with how many client support the attribute value in the table below -

Table 3-1 Possible values for the wallpaper color attribute along with support nos.

SL	Wallpaper Color (Bit)	# of Clients
1	0	3
2	1	1
3	2	3319
4	4	43
5	8	301
6	9	1
7	10	9
8	12	573
9	14	2
10	16	1993
11	18	816
12	24	34
13	32	7

We are getting 13 different attribute values for the wallpaper color ranging from 0 bit to 32 bit and this is a discrete value attribute. To convert this attribute to gene we have to take $log_2(13) \approx 4$ bit string. So the gene for wallpaper color can be considered as shown in the following table -

Table 3-2 The gene representation of the wallpaper color attribute.

SL	Wallpaper Color (Bit)	# of Clients	Gene Representation
1	0	3	0000
2	1	1	0001
3	2	3319	0010
4	4	43	0011
5	8	301	0100
6	9	1	0101
7	10	9	0110
8	12	573	0111
9	14	2	1000
10	16	1993	1001
11	18	816	1010
12	24	34	1011
13	32	7	1100

The genes corresponding to the other attributes can be identified in the similar way. The total capability gene can be compiled by concatenation of all the genes representing different attributes.

3.4. Learning of Majority Supported Capability Set

For adaptation of the content within the resource limitation of the server the framework must learn about the Majority Supported Capability Set. This is the set which can maximize the content support for the clients currently available in the network. In some cases the supported content may or may not fully compatible with client's capability.

To identify the MSCS the learning is initiated. The capability genes are used for the learning. There are several phases associated with the learning. Also there are some logic defined for the fitness of a member, selection criteria for breeding and stopping of the learning.

3.4.1. Learning procedure

To start the learning phase two distinct set needs to be identified as Working Set and Test Set. The sets should not need to be mutually exclusive. Both the sets are built by taking some random members from the initial capability set. The size of the sets should be small enough compared to the initial capability set, but it should be able to cover all the different capabilities available. In this framework the size of the Test and Working set has been limited to 5% ~ 10% of total population size.

After building the Working and Test sets, the learning can be started. The Test set will be remained unchanged all over the learning phase; it will be only used to evaluate the working set members. The Working Set will go through several continuous breeding and generate new generations. Each generation is evaluated with the Test Set using the fitness function. The members for breeding from the Working Set are selected by the evaluation ranking. The learning will continue till one of the stopping criteria is reached. After the learning is finished the top members from the Working Set are selected as the Majority Supported Capability Set.

3.4.2. Fitness function for the learning phase

For the application of the genetic algorithm the population at any stage should be evaluated in such a way that maximizes the goal of the problem. In our dynamic adaptive content delivery problem our goal is to find out some capability set that are supported by the majority. To achieve this goal the fitness is defined as the matching ratio between two client capabilities. The capabilities can have multiple attributes. When ever some member of the Working Set is evaluated against the Test Set members; it is matched with the every attribute of the capability of every Test Set member. For example let us consider the Capability Set as C and it contains N attributes i.e. $C = \{A_1, A_2, A_3 \ldots A_N\}$. For a Working set member C_{WSi} and a Test Set member C_{TSj} suppose M ($M \leq N$) attributes matches. So the evaluation outcome will be M/N. The overall fitness of the Working Set member C_{WSi} will be the weighted average of all Test Set member evaluation as given in section 3.5.

3.4.3. Selection of the breeding member

The target of the learning is to maximize the capability support with the most capable member. The member whose fitness is greater can contribute more to achieve the goal. Breeding among such members increases the possibility of convergence towards the solution. So the selection of the breeding member is made proportional to the fitness of that member.

3.4.4. The Breeding and the New Generation

Two members are selected from the Working Set for each breeding. The genes of the members are some binary string as shown in figure below.

Type	Width	Height	Size Limit	Gray Scale	Color Depth	Resizable
101	1000	10110	0011	0000	10110	011

Figure 3-5 Representation of a capability Gene.

We have considered two point cross over for the breeding. The points of cross over is selected randomly and restricted at the beginning or end of some gene of particular attribute. By the cross over the exchange of the gene is happened between two members and generates two new members of the new generation. The example of a two point crossover has been shown in the following figure -

101	1000	10110	0011	0000	10110	011

+

111	1010	10110	0010	0100	11100	001

Crossover Point 1 Crossover Point 2

111	1000	10110	0011	0100	11100	001

+

101	1010	10110	0010	0000	10110	011

Figure 3-6 Crossover of the Genes.

There is some possibility that the learning may converge towards local maxima. To avoid this possibility of local maxima mutation has been applied with very low possibility, 0.02% in case of proposed framework. For the mutation the "Bit Flip" mutation [13] has been considered where a random bit has been reversed. The mutation procedure and the result has been show in the following figure -

111	1010	10110	0010	0100	11100	001

Mutation Point

111	1010	10110	0010	0110	11100	001

Figure 3-7 Mutation of a Capability Gene.

3.4.5. Stopping Criteria of The Learning

The perfect learning may not be achieved by the genetic algorithm. Some criteria should be set to stop the learning. In case of the dynamic content adaptation the following criteria can be considered for stopping the learning -

- The solution is found that satisfies minimum criteria for majority support.
- Fixed number of generations reached.
- Allocated budget (computation time/resource) reached
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- Combinations of the above.

3.5. Mathematical Formulation of The Learning

First let us consider about the mathematical model of the fitness function used in the learning. For the fitness evaluation the following parameters are defined:

M = member gene for which fitness is to be evaluated.

TS= Test set for evaluation.

TS_i = i^{th} member of TS (Test Set)

A= Attribute set of the capability.

A_j= j^{th} attribute of the attribute set.

$i = 1, 2, 3 \dots n$, the members of the Test Set

$j = 1, 2, 3 \dots m$, the attributes of the Attribute Set

m = Total numbers of attributes in the Attribute Set

n = Total size of Test Set

comply is a step function which returns 1 when the first argument complied with the second otherwise it returns 0.

The fitness function can be formulated as

$$fitness(M) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} comply\left(A_j(M), A_j(TS_i)\right)}{m \times n}.$$

Top t members in the working set as per fitness function can be expressed by

$$\bigcup_{x=1}^{t} (\forall_{c_x \in WS, fitness(c_x) \geq fitness(c_{x+1})} c_x)$$

Here, WS is the Working set and c_x is the x^{th} member of WS (Working Set)

The selection possibility of a member c_x of working set to be the candidate of next breed is defined by

$$Pr_{selection}(c_x) = \frac{fitness(c_x)}{\sum_{i=1}^{n} \forall_{c_i \in WS} fitness(c_i)}$$

To formulate the stopping criteria of the learning let us consider the following parameters.

M_m = m^{th} member of MSCS.

f_{min} = Minimum fitness value that can be considered for learning.

S_{min} = Minimum acceptable MSCS size.

I_{max} = Maximum number of iteration allowed before terminating.

The stopping criteria can be represented by one of the following inequalities:

$$\max_{\forall c_m \in MSCS} (fitness(M_m)) \geq f_{min}$$

$$size(MSCS) \leq S_{min}$$

$$count(iteration) \geq I_{max}.$$

3.6. The Delivery of Content by Analyzer

The role of the analyzer is to deliver the appropriate content to the client upon getting the client request. Analyzer uses the learned *MSCS* for this purpose. Suppose the learning phase returns *MSCS* with size m i.e. it contains m number of capability set. Thus *MSCS* can be expressed by $\{C_1, C_2, C_3 \dots C_m\}$. In response to any client's request the analyzer first identify the the requested client C_{client}. Then it uses the same fitness function like the learning algorithm. But here the *MSCS* is considered as the *WS* and the client capability C_{client} is treated as the Test Set. The analyzer stores the evaluation set $E = \{E_1, E_2, E_3 \dots E_m\}$ after the fitness evaluation, where E_i is the evaluation outcome of C_i and so on. Then the Analyzer finds out the maximum evaluation and the corresponding capability and delivers the content according to that capability.

3.7. The Algorithm

The proposed learning algorithm for dynamic adaptive content delivery is as follows:

PROCEDURE Extract_MSCS

/*Initialization of variables and constants*/

Client Universe (*CU*) ← All Current Universe Member

Working Set (*WS*) ← Select Random (*CU*)

Test Set (*TS*) ← Select Random (*CU*)

Attribute Base (*A*) ← Build Attribute (*WS*) UNION Build Attribute (*TS*).

```
    Evaluation Set (ES) ← NULL
    Current Set (CS) ← NULL
    F_MIN ← Minimum Acceptable Fitness
    I_MAX ← Maximum Acceptable Iteration
    S_MIN ← Minimum Acceptable Size
    S_MSCS ← Size of MSCS
    NEG_MAX ← Maximum Negative Number
    fitness ← NEG_MAX
    loop_count ← 0

    NP ← WS      //Assigning New Population (NP) to Working Set (WS)
    /* Creation of generations using Genetic Algorithm*/
    LOOP
        NP ← Crossover (CS)
        NP ← Mutate (CS)
        FOR EACH NP AS NP_i
            ES_i ← Evaluate (NP_i, TS)
        END LOOP
        CS ← Extract Best Fit Population (NP, ES)
        fitness ← sum_fitness (CS)
        IF fitness > F_MIN THEN
            MSCS ← CS
            RETURN
        ELSE IF loop_count > I_MAX OR Unique Size (CS) ≤ S_MIN THEN
            MSCS ← Most Fit Set (CS, S_MSCS)
        END IF
        loop_count ← loop_count +1
    END LOOP
RETURN MSCS
END PROCEDURE Extract_MSCS
```

3.8. Complexity Analysis

The complexity of the proposed framework for dynamic adaptive content delivery depends on the learning phase for finding MSCS and analyzing phase for delivering contents. The learning is executed once for a comparable long period existence. On the other hand the delivery is performed by the analyzer as a regular task. The complexity of each task has been briefly described in the following sections.

3.8.1. Complexity for the Learning

The learning phase can be divided into several sub phases including set preparation for Test Set and Working Set, Evaluation, Candidate Selection, Cross over and Mutation for Genetic Algorithm learning.

The set preparation operation is straight forward as it only requires some random selection form the initial population. Both Test Set and Working Set preparation can be of linear order complexity based on the number of member in the set. Let the size of test set and working set are M and N.

So, set preparation complexity $C_{set} = O(M) + O(N)$.

The evaluation will run for each member of working set and evaluate against each member of test set. It will compare each attribute. So if there are k attributes in the capability then each pair evaluation will have k comparisons. N member of the working set should be evaluated against M members of test set and each evaluation is done by k comparisons. So the number of comparisons required is $(k \times N \times M)$.

So the complexity of the evaluation will be, $C_{eval} = O(kMN)$

For the selection step all the member of Working Set should be ordered according to their evaluation value. The sorting can be done by insertion sort or bubble sort and the complexity of the sorting can be considered as the complexity of the selection step. The worst case complexity can be taken for the complexity measurement.

So the selection complexity, $C_{selection} = O(M^2)$

The crossover is done after the selection step and it takes two members at a time from the working set and does the crossover in a linear time.

And so crossover complexity, $C_{crossover} = O(M/2)$

The mutation is done by flipping just one bit and the probability of the mutation is very less compared to others. So mutation complexity can be ignored from the total complexity calculation.

The overall complexity of a single iteration of the learning is the summation of complexity of previous steps, i.e.

$$\text{Complexity of Iteration, } C_{iteration} = C_{eval} + C_{selection} + C_{crossover}$$

$$= O(kMN) + O(M^2) + O(M/2)$$

The learning will continue the previous steps (excluding set preparation) till the goal or stopping criteria reached. For the complexity analysis let us consider the worst case where no optimal solution has been found and the learning has reached the maximum allowed iteration. Let us assume the maximum iteration has been set to I_{max}. So all the steps will repeat I_{max} times.

So the complexity of the learning will be,

$$C_{learning} = C_{set} + I_{max} \times C_{iteration}$$

$$= O(M) + O(N) + I_{max} \times (O(kMN) + O(M^2) + O(M/2))$$

For simplicity let us consider the size of test set and working set is equal i.e. $M \approx N$. The number of attribute in the capability k is very smaller compared to set size, i.e. $k << M$. And finally the maximum iteration size can be considered some multiple of set size, i.e. $I_{max} = l \times M$, where l is the multiplication factor and $l << M$. So the learning complexity becomes,

$$C_{learning} \approx O(M) + O(M) + l \times M \times (O(kM^2) + O(M^2) + O(M/2))$$

$$\approx \quad O(M) + O(M^3) \qquad \text{[ignoring } k, l \text{ and } 1/2]$$

$$\approx \quad O(M^3)$$

The worst case complexity of the learning is polynomial order of the set size.

3.8.2. Complexity for the Analyzer

The complexity of the analyzer can be calculated in the same way as the complexity of the evaluation step of the learning phase. The only difference is the parameter. The evaluation step deals with Test Set and Working Set but for analyzer complexity we have to deal with client's capability and MSCS accordingly.

The analyzer will evaluate the fitness of each member of *MSCS* against the requested client capability. It will compare each attribute of the capability set. So if there are k attributes in the capability then each pair evaluation will have k comparisons. Suppose there are S members in the *MSCS*, so these S members of the *MSCS* should be evaluated against the single client capability and each evaluation is done by k checking. So number of comparison required is $(k \times S)$.

So the complexity of the Analyzer will be, $C_{analyzer} = O(kS)$

3.8.3. Overall Complexity of the Frameork

The overall complexity of the framework is based on the complexity of the learning and complexity of the analyzer. As we have seen in the previous sections that the learning has a polynomial complexity of the size of initial set. And the analyzer has linear complexity of the size of *MSCS*, S and number of attribute in capability, k.

The learning remains unchanged for a very large span of time. If we assume the time span on which the learning remains unchanged is T and the time taken for learning is t then it is obvious that $T >>> t$. So we found $t/T \approx 0$.This concludes that the learning complexity has very minimum effect in the overall framework. So the framework complexity becomes the complexity of the Analyzer which is a linear complexity.

3.9. Chapter Summary

In this chapter the design of the dynamic adaptive content delivery framework using the genetic algorithm has been described. The framework has been designed in such a way that it can satisfy the majority client subject to the server resource limitations. Although in many cases the perfect match may not be possible but the framework improves the content delivery performance very effectively. The chapter concludes by providing a worst case complexity analysis of the proposed framework. It has reduced the NP hard problem to a polynomial problem with a very little deviation from the perfect solution which is impractical due to the resource constraint of the server. The next chapter confirms the performance of the framework by testing the framework with some real network data sets from a renowned Telecom Operator (Axiata Bangladesh Limited). Experimental results showing the improvement of the proposed framework over currently available adaptation algorithms have also been presented in the next chapter.

Chapter 4. Result Analysis

The proposed dynamic adaptive content delivery framework will not always deliver the perfect content for all clients. To compare the output of the proposed framework it has been evaluated against Brute Force adaptation technique which produces the perfect content for all the clients. It has been also compared with some well established adaptation techniques. Lastly the learning phase of the proposed framework has also been tested with many variable parameters to check the performance effect. These algorithms along with the experimental results are presented in the following sections.

4.1. Brute Force Adaptation Technique

Since the Brute Force algorithm explores all possible content solutions the solution space of this algorithm grows exponentially with the problem size. Thus the optimal solution using Brute Force algorithm can not be found even for moderate size of problems. For example, if there is a single composite content with M attributes and the number of the client in the network is N, then the total possible number of content will become M^N, since all possible combination of all attribute should be available for all the clients. So for the attribute of 10 values and only for 10 different clients the possible content combination becomes 10^{10}. This is only feasible when both the attribute values and number of clients are very small. The problem is shown in the figure below -

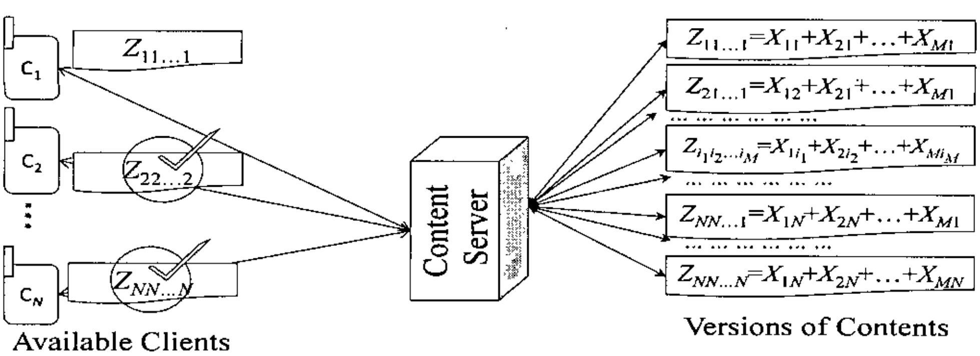


Figure 4-1 Brute force adaptation for M attribute composite content and N clients.

$Z_{i_1 i_2 \ldots i_M}$ is a version of the composite content where i_j is the capability of the j^{th} component. Thus the composite content version is expressed as $X_{1i_1}+X_{2i_2}+\ldots+X_{Mi_M}$. i_j has N possible values as there might be N capabilities for N clients.

4.2. Content Adaptation using Info-Pyramid

The Inforpyramid based adaptation is proposed by Mohan et al [1]. A progressive data representation scheme called the InfoPyramid is used for the adaptation. Content items on a Web page are transcoded into multiple resolution and modality versions so that they can be rendered on different devices as shown in Figure 4-2. For example, a video item is transcoded into a set of images so that it can be rendered on a device not capable of displaying video. For certain devices, the appropriate content modality may not be available. The required modality may be generated by transforming other modalities. For example, a video clip can be transformed into images showing key-frames (as shown in Figure 4-2), while text can be synthesized into speech. The InfoPyramid provides a multi-modal, multi-resolution representation for the content items and their transcoded versions. It also used a customizer that selects the best versions of content items from the InfoPyramids to meet the client resources while delivering the most "value." The customizer allocates resources on the client among the items in the content document. This resource allocation results in the selection of the appropriate resolution or modality of the content items. If the client has limited resources (such as a PDA or pager), some of the content items may not get any resources assigned and thus not be delivered to the client. The algorithm proposed a novel value-resource framework for the customizer. This value-resource framework allows the algorithm to design and analyze a number of content adaptation strategies.

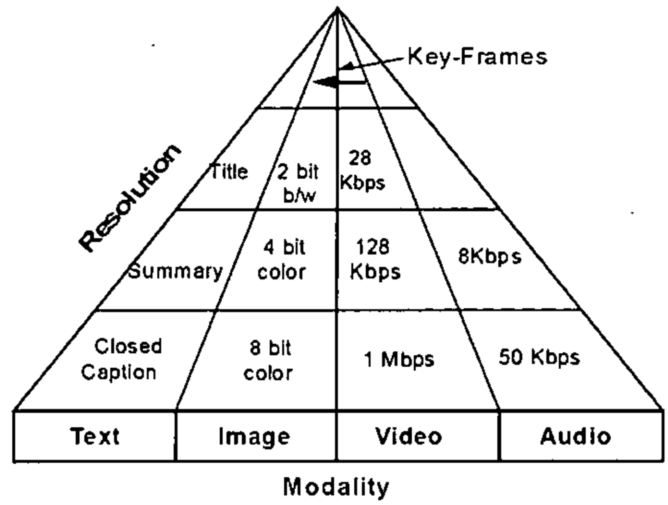


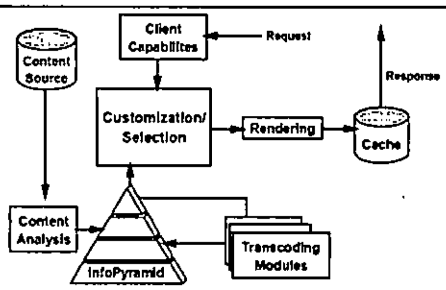
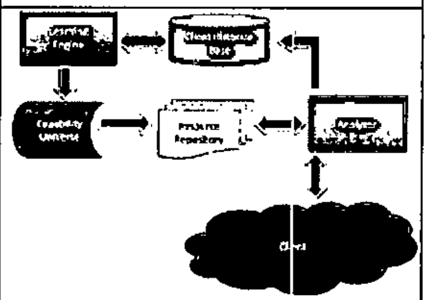
Figure 4-2 Infopyramid for video content.

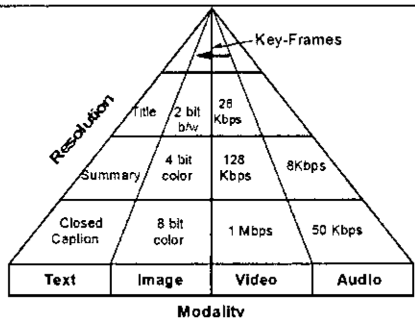
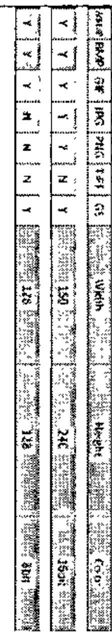
Our framework has been also compared with this inforpyramid based adaptation both in the way of working procedure and the outcome achieved.

4.2.1. Comparison between Info-Pyramid & Proposed Framework

There are similarities & differences between our proposed framework and the adaptation by Infopyramid technique. These are shown in the Table 4-1.

Table 4-1 Comparison between Infopyramid and proposed framework.

Comparison on Criteria	**Info-Pyramid**	**Proposed Framework**
Framework Architecture		
Representation	Info-Pyramid uses a Pyramid of multimodal and multi resolution	In the proposed adaptation all the modalities are

	representation of content. Where the resolution is placed in hierarchical fashion from bottom to top and the modalities are placed in the bottom of the pyramid hierarchy.	accommodated in MSCS (a set of gene) and every member of MSCS contain a particular resolution for each of the modality types.
Representation (Diagram)		
Transcoding	Info-Pyramid transcode the contents according to the modality and resolution match of the client, which are available in Info-Pyramid hierarchy.	Proposed framework transcode contents according to the MSCS modality and resolution.
Transcoding Complexity	As Info-Pyramid has to deal with all the modality and resolution combination it has a polynomial complexity, i.e. for n types of modality each having m resolution the complexity will become n^m.	For the same configuration as above proposed framework will have linear complexity of the number of genes in the MSCS achieved from learning.
Initiation	The Info-Pyramid should be designed or initiated manually or it should cover all the modality and	The MSCS for adaptation are automatically generated after each learning process.

	corresponding resolution of contents.	
Dynamic Adaptation	Info-Pyramid has not addressed the dynamic adaptation with the environment changes.	Proposed framework addresses the effect of environment change and adapts itself by recalculating the new MSCS.
Analyzer Task	Analyzer has to cross check each requesting client's capability with each and every modality and resolution available in Info-Pyramid.	Analyzer has to cross check against the members of MSCS and decide accordingly to the best match.
Learning	No learning is required on environment change and it is designed manually.	Learning is the key to this adaptation technique.

4.3. Experimental Data Set

The experimental data for the test is collected from a real telecom network repository. It has been extracted from the GPRS request log file of renowned Telecom Operator in Bangladesh, Axiata Bangladesh Limited. The log file has been collected for a long span of time, for more than 2 years. The reason behind taking long time data is to identify the network client's environment change effect. The log files are simple ASCII file which contain some basic information of a content request covering date and time of the requests, and user agent of the clients. Some sample data of the log file is shown in Table 4-2.

Table 4-2 Data of the log file from AXB GPRS.

Date & Time	User Agent
2005-06-26 11:34 am	Nokia7250/1.0 (3.12) Profile/MIDP-1.0 Configuration/CLDC-1.0
2005-06-26 11:34 am	SEC-SGHE310/1.0

2005-06-26 11:34 am	Nokia3120/1.0 (06.01) Profile/MIDP-1.0 Configuration/CLDC-1.0
2005-06-26 11:34 am	Nokia6100/1.0 (04.70) Profile/MIDP-1.0 Configuration/CLDC-1.0
2005-06-26 11:34 am	Nokia2650/1.0 (5.48) Profile/MIDP-1.0 Configuration/CLDC-1.0
2005-06-26 11:34 am	Nokia6230/2.0 (04.44) Profile/MIDP-2.0 Configuration/CLDC-1.0
2005-06-26 11:34 am	Nokia6820/2.0 (3.70) Profile/MIDP-1.0 Configuration/CLDC-1.0
2005-06-26 11:34 am	Nokia3120/1.0 (06.01) Profile/MIDP-1.0 Configuration/CLDC-1.0
2005-06-26 11:34 am	Nokia3200/1.0 (4.16) Profile/MIDP-1.0 Configuration/CLDC-1.0
2005-06-26 11:34 am	Nokia3200/1.0 (4.16) Profile/MIDP-1.0 Configuration/CLDC-1.0

This information is converted into gene string using the WURFL repository. The following steps have been followed for the gene string generation.

- The entire log files have been imported into some MySQL table.
- In MySQL some query has been used to identify the entire possible attributes along with the values they can have.
- The attributes are then mapped into gene string.
- Each value of attributes has been converted into corresponding gene string.
- By concatenating the entire attribute gene string the gene for a particular user agent or client is identified.

After the conversion some sample output is given below -

Table 4-3 Gene String Converted Data of the log file from AXB GPRS.

User Agent: Gene String
SEC-SGHE310/1.0;0011000001110110111001
SAMSUNG-SGH-X480/1.0;0011000001110110111001
SEC-SGHX430;0011100001110110111001
SEC-SGHE310/1.0;0011000001110110111001
SAMSUNG-SGH-X640/1.0 UP.Browser/6.2.2.6 (GUI)

MMP/1.0;0011100001110100011001
SEC-SGHX430;0011100001110110111001
SAMSUNG-SGH-X640/1.0 UP.Browser/6.2.2.6 (GUI) MMP/1.0;0011100001110100011001

4.4. Result Analysis

In the experiment we do adaptation of Image content. More than 10 attributes are identified for Image and 5 attributes among them are found relevant for adaptation. For these 5 attributes 22 bit gene has been constructed considering,

- 6 different image types with 6 bit,
- Grayscale with 1 bit,
- 25 different widths with 5 bit,
- 52 different heights with 6 bit and
- 13 different colors with 4 bit.

The experiment runs with a working set of 400 members and a test set of 800 members. Several terminologies are defined in Table 4-4 to analyze the data set and the performance of the dynamic adaptation technique. The experimental result is given in Table 4-5. We have a brief discussion about the result following the experimental outcome.

Table 4-4 Terms definition for performance analysis.

SL	Term	Description
1	Date	The particular date when the data has been taken.
2	Total	The number of total clients that has requested to the network in the particular date.
3	WS Unique	It is the number of unique members in the working set.
4	WS/MSCS	The ratio between WS Unique and MSCS indicates the effectiveness of the adaptation to reduce the capability diversity.
5	Score	It indicates the measurement of fitness of MSCS capability with the original requests by function fitness.

6	Iteration	Number of iterations required to finish learning.
7	Best Gen	Indicates the number of iterations required to generate sub-optimal result.

Table 4-5 Experimental outcome from the framework.

Date	Total	WS Unique	MSCS	WS / MSCS	Score	Iteration	Best Gen.	Time required for learning (Sec)
26/05/2005	6,148	52	3	17.3	76.7%	34	25	400
01/07/2005	2,097	52	2	26.0	77.9%	19	19	217
31/08/2005	837	53	2	26.5	75.8%	18	18	243
31/03/2007	4,214	52	5	10.4	76.0%	27	4	327
29/04/2007	4,966	53	2	26.5	77.4%	18	18	209

Lets us focus on the result of first row of Table 4-5. In the first row the data has been collected from network request of 26th May 2005 where the total number of requester was 6,148. The Test Set & Working Set has been generated from this requester list. At the working set of 400 members 52 has been found with unique capabilities and presented by WS Unique. After learning we have got MSCS comprised with only 3 members. So the reduction from unique working set members to MSCS is 52 to 3 and represented by MSCS/WS Unique which is 17.3 times reduction. This reduction also indicates the minimum reduction in storage as well as computing power, ignoring considerable learning time. The fitness achieved is 76.7% i.e., the MSCS got 76.7% support from the Test Set. The learning was done in 34 iterations and the MSCS was found in 25th iteration, it indicates that at the 25th iteration the learning converge to the maximum and began diverge again.

The overall analysis on Table 4-5 shows that 52-53 capability sets are covered by MSCS of size 2-5 maintaining a score of 75-78%. The learning time along with MSCS identification also remains between 4 ~ 25 iterations that is considerably small. So the applicability of GA in this dynamic content adaptation seems to be realistic as the requirement and the number of iteration in GA seems to be reasonable.

In case of brute force technique the solution can be achieved by creating 22^{52} versions of the image contents to gain 100% support. But which has been achieved only by 3 versions of contents using the proposed framework. So the framework has huge gain over the brute force technique of content adaptation.

The performance comparison between the proposed framework and the Infopyramid based adaptation is explained in Table 4-6.

Table 4-6 Performance comparison between Infopyramid & proposed framework.

Item	Proposed	Infopyramid
Possible versions	2^{22}	2^{60}
Total candidate	6000+	Few (<100) for Testing
Learned versions	2~5	2^{60}
Learning complexity	Polynomial	Manual
Decision complexity	Linear	Polynomial
Support efficiency	75~78%	Transformed content based
Adaptation	Automatic by GA	Manual human decision
Dynamic adaptation	Yes	No
Content value identification	As given in Section 4 by fitness function.	Value of content represented by, (1/1+*D*). Where *D* is the distortion of the versions.
Learning duration	18~34 Iteration	Manual decision

4.5. Performance on Parameter Change

The performance of the learning of our proposed dynamic adaptive content delivery depends on various performance parameters of Genetic Algorithm. To observe the effect on solution performance test and working set size, and mutation rate have been adjusted and tested. The overall result on performance parameter change is shown in Table 4-7.

Table 4-7 Performance on parameter change.

TS Size	WS Size	Mutation	Iteration	Best Gen	Score	MSCS
200	200	0.20%	9	4	78.50%	5
1,600	200	0.20%	20	20	77.79%	2
2,000	200	0.20%	18	10	76.66%	5
3,000	200	0.20%	12	12	77.46%	2
800	50	20.00%	3	3	77.80%	2
800	80	20.00%	4	4	77.40%	1
800	100	20.00%	20	7	76.68%	5
800	200	0.20%	4	2	77.17%	2
800	800	0.20%	13	10	76.85%	5
800	1,600	0.20%	71	4	75.97%	5
800	3,000	0.20%	87	9	76.72%	5
800	400	0.20%	22	22	76.60%	2
800	400	0.02%	27	23	77.68%	4
800	400	2.00%	34	24	76.50%	5
800	400	20.00%	24	0	75.80%	5

From the result observed in Table 4-7 it can be identified that the performance degrades with the small & large size of working set. In case of small working set the candidate may not contain each capability type and hence some capabilities are ignored in all generations which results in poor performance. For larger working set although all the capabilities are covered it takes more iteration for the learning to finish. Increasing the test set size increases the time for iteration steps as more comparisons need to be done for the evaluation. Increasing the mutation rate changes the generation so much that it becomes incompatible or it is not supported by test set. As a result the learning stops in premature stage with a small MSCS.

4.6. Chapter Summary

In this chapter the proposed dynamic adaptive content delivery framework using Genetic Algorithm has been compared with a Brute Force Content Adaptation technique and an Infopyramid based adaptation technique. The results show that the proposed framework provides better solutions and also provides better adaptation technique than the Infopyramid based adaptation. It is also clear from the result analysis that the running time of the proposed framework does not depend on the number of clients since the learning is based on the randomly selected Test Set and Working Set which is constant. The following chapter concludes the thesis and also presents some suggestions for further improvement in this area of research.

Chapter 5. Conclusion

This chapter starts with describing some major contributions of this thesis and finally presents some options for future research in this area.

5.1. Major Contribution

The previous content adaptation techniques aim at the adaptation of a particular type of content or content set in a static manner. There was no adaptation that can adapt itself dynamically on the environment changes. In this thesis a new content adaptation framework is presented that aims to adapt any kind of content in a dynamic fashion and can adapt itself on the environment changes. In the dynamic adaptation the content adaptation for the majority has been satisfied. Beside this it also takes care of the resource limitation of the server. Thus this framework can be considered as the optimal content delivery framework in terms of client support, resource utilization and dynamic adjustment. The dynamic adaptation will happen automatically, based on the criteria defined for the new learning to initiate either by checking the performance of the system or introduction of certain new devices or removal of certain old devices.

Also, this thesis explores the idea of applying Artificial Intelligence technique like Genetic Algorithm to the Adaptive Content Delivery, which has not been done previously. The Genetic Algorithm based adaptation approach presented here not only finds the Majority Supported Content Set in optimal time, but also provides the way to the adaptation supporting the majority in an efficient manner. It requires high complexity in the learning phase. But the learning happens less frequently, so the complexity of learning can be ignored compared to the total time span learning result persists.

Since the framework proposed in this thesis solves the adaptive content delivery problem in a dynamic changing environment, it can be implemented in any system that needs some sort of dynamic adaptation. The model on which the proposed framework is built perfectly fits today's content delivery system. To ensure the best possible content service with limited resources, the content adaptation must be optimized for maximum gain by achieving majority client's support. This is exactly what the proposed framework is designed to solve.

5.2. Future Works

In our proposed framework we have used the Genetic Algorithm to identify the Majority Supported Capability Set. The Majority Support is identified by the attribute matching is not always optimal. There are several reasons behind this non optimality. The first thing should be considered is the priority of attributes, some attribute may be dominating over others. For example in case of image content as shown in Figure 5-1 the image type may be considered as the key attribute for image capability of a client. Without the proper match of image type the full match of other attribute will result in a non compatible image. The solution of this problem can be resolved by assigning attribute weight-age. The critical attribute can be assigned with higher weight then the others. On the other hand in some case matching of some particular attribute may be sufficient without matching of others, for example let us consider again the image type scenario. There can be multiple types of images that can be cross matched. But it is sufficient enough to match only a particular image type and ignore others. There are also some correlations between the attributes. For example whenever dealing with image if we consider a particular image type such as GIF the color will not be more than 8 bit but for JPEG the color may be 16 bit or more. Thus if we can assign weight to attribute and utilize correlation among attributes in the fitness function more perfect result for adaptation can be achieved.

Gene	WBMP	BMP	GIF	JPG	PNG	TIFF	GS	Width	Height	Color
Client A	Y	Y	Y	Y	Y	N	Y	160	240	16bit
Client B	Y	Y	Y	N	N	N	Y	128	128	8bit

Figure 5-1 Attribute weight-age and correlation.

Genetic Algorithm is a greedy kind of solution. There can be several groups of similar capability clients in the environment and some of them may be dominating and the other may have minor effect on the environment. The scenario is described in Figure 5-2 below. Whenever Genetic Algorithm runs on this environment it may converge towards the dominating group and as a result we may get the outcome with no support for minor groups. The solution to this problem can be resolved by first applying some clustering algorithm to identify the groups and then running genetic algorithm on those groups.

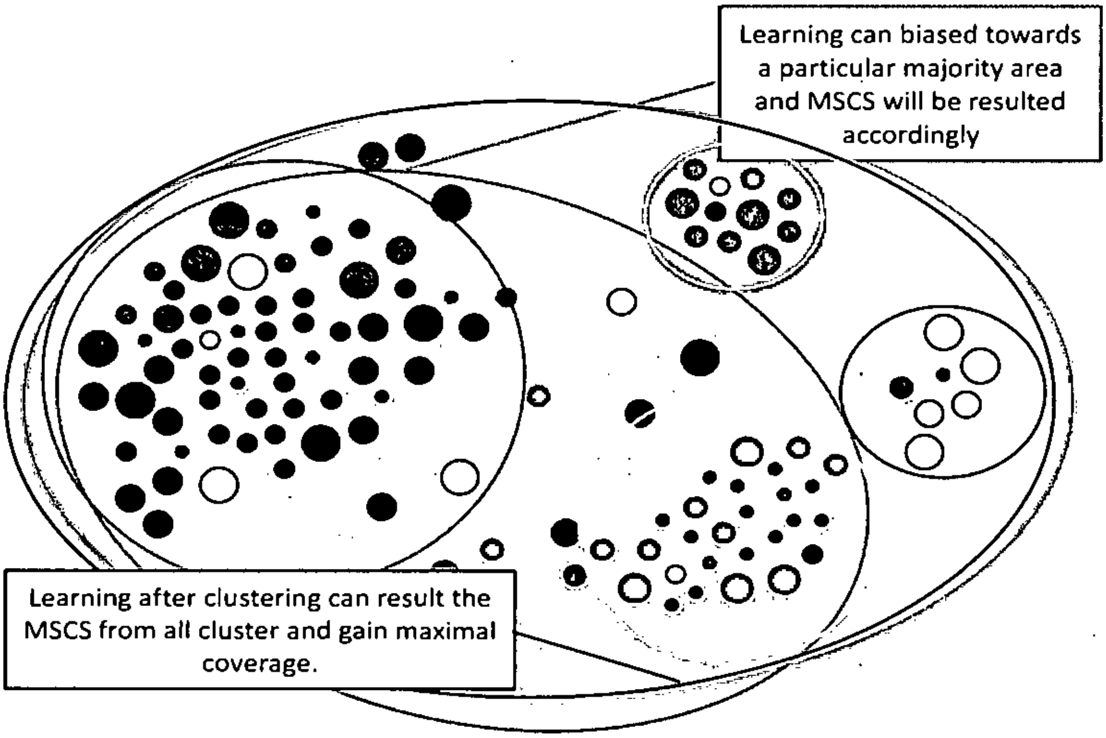


Figure 5-2 Clustering & grouping of the clients.

References

[1] Mohan, R., Smith, J. and Li, C., "Adapting multimedia internet content for universal access", IEEE Transactions on Multimedia, Vol-1, No. 1, March 1999, pp. 104 – 114.

[2] Fan, X., Xie, X., Ma, W., Zhang, H. and Zhou, H., "Visual attention based image browsing on mobile devices", Proceedings of the International Conference on Multimedia and Expo, Vol-2, 2003, pp. 53 – 56.

[3] Zhang, H., "Adaptive content delivery – A new application area for media computing research", Microsoft Research, China, available at http://research.microsoft.com.

[4] Bickmore, T.W. and Schilit, B.N., "Device-independent access to the World Wide Web," Proceedings of the 6th International WWW Conference, 1997.

[5] Chang, S.F., Eleftheriadis, A., Anastassiou, D., Jacobs, S., Kalva, H. and Zamora, J., "Coulmbia's VOD and multimedia research test bed with heterogeneous network support", International Journal of Multimedia Tools and Applications, Vol-5, No. 2, September 1997, pp. 171-184.

[6] Borning, A., Lin, R.K. and Marriott, K., "Constraint-based document layout for the web", ACM Multimedia System Journal Vol-8, No. 3, 2000, pp. 177–189.

[7] Chen, Y., Ma, W.Y. and Zhang, H.J., "Detecting web page structure for adaptive viewing on small form factor devices", Proceedings of WWW'03, 2003.

[8] Opera Mini - Free mobile Web browser for your phone. http://www.operamini.com/.

[9] Content Negotiation. http://www.ietf.org/html.charters/OLD/conneg-charter.html.

[10] CC/PP: A user side framework for enhanced content negotiation. http://www.w3.org/TR/NOTE-CCPP/.

[11] Synchronized Multimedia Integration Language. http://www.w3.org /TR/REC-smil/.

[12] Gaedke, M., Beigl, M., Gellersen, H.W. and Segor, C., "Web content delivery to heterogeneous mobile platforms", Proceedings of the Workshops on Data Warehousing and Data Mining: Advances in Database Technologies, Vol-1552, November 1998, pp. 205–217.

[13] Genetic Algorithm - Wikipedia, the free encyclopedia. http://en. wikipedia.org /wiki/Genetic_algorithm.

[14] WURFL. http://wurfl.sourceforge.net/.

[15] Spyglass-Prism. http://www.spyglass.com

[16] Intel QuickWeb. http://www.intel.com/quickweb.

[17] ProxiNet. http://www.proxinet.com.

[18] W3C Extensible Markup Language (XML). http://www.w3c.org/XML/.

[19] Xie, X., Wang, C., Chen, L., Ma, Y., "An adaptive web page layout structure for small devices", Springer-Verlag, October 2005.

[20] WAP Forum, "Wireless Application Protocol - Wireless Markup Language Specification", WAP Forum, 1998.

[21] World-Wide Web Consortium (W3C), "Compact HTML for Small Information Appliances", W3C NOTE, February 1998.

[22] W3C Extensible Stylesheet Language (XSL). http://www.w3.org/TR/WD-xsl/.

[23] Berners, T., Fielding, R., Frystyk, H., "Hypertext Transfer Protocol - HTTP/1.0", RFC 1945.

[24] Badros, G., Borning, A., Marriott, K., Stuckey, P., "Constraint cascading style sheets for the web", Proceedings of UIST, Asheville, USA, 1999, pp. 73–82.

[25] Björk, S., Holmquist, L.E., Redström, J., Bretan, I., Danielsson, R., Karlgren, J., Franzén, K., "WEST: A Web Browser for Small Terminals", Proceedings of UIST, Asheville, USA, 1999, pp. 187–196.

[26] Chen, L.Q., Xie, X., Fan, X., Ma. W.Y., Zhang, H.J., Zhou, H.Q., "A visual attention model for adapting images on small displays", ACM Multimedia System Journal Vol-9, No. 4, 2003, pp. 353–364.

[27] Fuchs, M., "An evolutionary approach to support web page design", Proceedings of 2000 Congress on Evolutionary Computation, Piscataway, USA, 2000, pp. 1312–1319.

[28] González, J., Rojas, I., Pomares, H., Salmerón, M., Merelo, J.J., "Web newspaper layout optimization using simulated annealing", IEEE Transaction System, Man, Cybernet. – Part B: Cybernet. Vol-32, No. 5, 2002, pp. 686–691.

[29] Gu, X.D., Chen, J.L., Ma, W.Y., Chen, G.L., "Visual based content understanding towards web adaptation", Proceedings of 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Malaga, Spain, 2002, pp. 164–173.

[30] Milic-Frayling, N., Sommerer, R., "SmartView: flexible viewing of web page contents", Proceedings of WWW'02, Honolulu, USA, 2002.

[31] Sofokleous, Anastasis, A., Angelides, Marios C., "Dynamic Selection of a Video Content Adaptation Strategy from a Pareto Front", Computer Journal, Oxford University Press, Vol-32, No. 5, 2002, pp. 686–691.

[32] Wobbrock, J.O., Forlizzi, J., Hudson, S.E., Myers, B.A., "WebThumb: interaction techniques for small-screen browsers", Proceedings of UIST, Paris, France, 2002, pp. 205–208.