PERSONALIZED ADVERTISEMENT ASSIGNMENT SYSTEM

by

ONUR BOZKURT

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
December 2009

PERSONALIZED ADVERTISEMENT ASSIGNMENT SYSTEM

APPROVED BY:

Assistant Professor Kemal KILIÇ ....
(Thesis Supervisor)

Associate Professor Bülent ÇATAY ....

Assistant Professor Gürdal ERTEK ....

Assistant Professor Nihat KASAP ....

Assistant Professor Murat KAYA ....

DATE OF APPROVAL: ....

PERSONALIZED ADVERTISEMENT ASSIGNMENT SYSTEM

Onur BOZKURT

IE, Ms. Sc. Thesis, 2009

Thesis Supervisor: Kemal Kılıç

Keywords: Personalized Advertising, Advertisement Assignment, Optimization, Virtual Platforms

Abstract

This thesis presents a comprehensive framework which will be used to maximize the advertising revenues of a company that develops a 3-D virtual reality social platform. The comprehensive framework includes the development of a personalized advertising business model for the company, representation of the business model with a mathematical program and proposing a set of heuristic solutions for the personalized advertising problem. The proposed heuristics are developed and their performances are compared with an experimental analysis under various conditions.

KİŞİSEL REKLAM ATAMA SİSTEMİ

Onur BOZKURT

IE, Yüksek Lisans Tezi, 2009

Tez Danışmanı: Kemal Kılıç

Keywords: Kişiye özel reklamcılık, Reklam atama, Optimizasyon, Sanal platformlar

Özet

Bu tez, 3 boyutlu sanal ortam geliştirmiş bir şirketin reklam gelirlerinin maksimize edilmesi için bütünsel bir yaklaşım sunmaktadır. Bu bütünsel yaklaşım çerçevesinde kişiye özel reklamcılık için bir iş modeli geliştirilmiş, iş modeli matematiksel programlama ile sunulmuş ve kişisel reklam problemi için sezgisel metodlar önerilmiştir. Önerilen sezgisel metodlar geliştirilmiş ve bu metodların performansları farklı test koşulları altında karşılaştırılmıştır.

Acknowledgements

After the most painful period of my life, I want to thank to my thesis supervisor Kemal KILIÇ for his sincerity, guidance, support and way of relieving me. His patience and confidence about me inspired and motivated me for this thesis to be a qualified one.

Special thanks for TÜBİTAK for the financial support which enables me to maintain my life quality and concentrate on my studies.

I want to thank to all of my friends in Sabanci University for their sharing of joy with me. I will remember them always by smiling.

My most sincere thanks are for my family. Their love persists me to fail better everyday and I succeeded because of their valuable support.

Lastly, I want to thank to “my dear” who makes me feel her love from kilometres away. I hope one day I will be sure about her reality and make her read these acknowledgements by smiling.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter One

INTRODUCTION

Advertising business is among the most sustainable industries considering the growth performance of the last 20 years. The industry is also negatively affected from the recent global economical crisis. However, reports assert that it will set off its growth starting from year 2011 [1]. As the world economy enlarges, it is expected that the advertising industry will grow even beyond the further state. On the other hand, despite the global crisis, one branch of advertising continued its growth. According to the report published by ZenithOptimedia [2], the Internet advertising continues to grow and it is forecasted to hit %15.1 of total advertising by year 2011 with an overall value of $72 billion [1].

The growth of internet advertising depends on an important reason besides the increasing number of internet users. Comparing other conventional advertising methods, internet advertising offers the advertisers a different form of relationship with its customers. The internet advertising media has the advantage of a closed loop control system, that is to say receives the reaction from the customer and adopts accordingly. This subtle difference creates a significance result in advertising. Advertisers attain the chance of directly touching the customers that they are targeting. Internet advertising opened a new era in advertising by offering a method other than mass advertising in various ways. Nowadays personalized advertisements are becoming more practical and cheaper particularly through internet. There are two different modes of internet advertising, namely, intrusive and non-intrusive methods. Intrusive method uses the information of user without any permission. For example, some methods use browsing history of the users. Such methods are perceived as a threat to user privacy by their critics. There are also non-intrusive methods that only utilize the keywords searched by the user.

This thesis motivates from a real life problem that a software company faces. The company develops a virtual reality based socialization platform similar to the well known Second Life [3] and recently launched Google Lively [4]. The current trend of increase in the number of users in social web sites and platforms actually provides further opportunities in internet advertising. Social web sites and platforms remove the concern of privacy and misuse of user information as users are willing to share information to a certain degree. In that sense, the virtual reality socialization platforms are similar to "Facebook" that users can share their personal stuff like interests, hobbies, relationship status, photos etc. with others as well as the system providers. Therefore, personalized advertising systems that can utilize the information provided by the users are becoming to evolve.

Generally speaking, the virtual reality worlds are real life simulations where users wander around in places which resemble real world places that are modelled by 3D technology. The platform simulates various environments such as squares, cafés, recreational facilities, etc. in which the users can interact with their real or virtual friends. Furthermore, you can travel around the world in your office, have a coffee with your friends while enjoying the Eiffel Tower view at a distance, or chit-chat with your colleagues while wandering around in Red Square or get familiar with Chinese customs before your company actually sends you to Shanghai as a sales representative.

Advertisements that are placed at certain locations within the virtual world are among the major source of income for the company. The users view certain advertisements while they are online at different virtual locations, e.g. when walking down the street occasionally a public bus with an advertisement on the sides might pass, the bus stop, the building across the road or the mirror in a café, etc. might display an advertisement.

Advertising revenue is composed of two different items; display revenue and click revenue. Display revenue is realized when a user is decided as "exposed" to an advertisement, thus the event is called exposure. Click revenue is realized when the user clicks to an advertisement, this event is called click-through. In web browsing a user is decided as "exposed" when a user enters to a web page. In our 3D world; consider there is a set of criteria that is used for deciding whether an exposure is occurred. When the user is wandering in the 3D world; advertisements are placed in virtual ad locations. There is no certainty for any of these placements will be transformed to an exposure. Exposures are realized whenever the user is in the vicinity of a place of an advertisement location. Note that the definition of vicinity is based on the virtual distance (i.e., magnitude of the advertisement on the screen) and the looking angle of the user (i.e., viewer).

In our study, we will focus to the problem that a local software company that develops 3D virtual socialization platform is facing. The company requires a Personalized Advertisement System (PAS) that will allow the advertisers to display their advertisements to the users that satisfy most the requirements that they specify. The *PAS* basically deals with the Personalized Advertising Problem which has two phases. The first one is basically a *matching problem* in which the set of candidate users that meet the minimum threshold levels in various features which are specified by the advertisers are determined. The second phase is the *assignment phase*, where a specific advertisement is assigned to an advertisement location whenever the user is approaching to the vicinity of a place of an advertisement location.

The *personalized advertising problem* is introduced relatively recently and the existing literature on the topic is scarce. Therefore, there is no generally accepted business model in the literature. One of the major contributions in this paper is developing a business model for such a socialization platform which is created based on our interviews with the company. Next, a mathematical model of the advertisement assignment problem is introduced. The constraints that are based on the contracts between the publisher (in our case the developer of the virtual socialization platform) and the advertiser make the problem intractable. Furthermore, the login time of the users, where they will visit and how long they will stay during each time login are unknown which further increases the computational complexity of the problem. Considering the increasing number of members of such sites and the fact that the assignment decision must be made in reasonable time (often instantaneously) the proposed solutions are heuristics in nature. Note that, in this paper, we particularly focused on the second phase of the problem, namely, the *personalized advertisement assignment* problem.

The organization of the paper is as follows: In Chapter 2 literature review is provided. Chapter 3 describes business model and defines the problem explicitly. Furthermore solution methods are explained with all details in this section. Chapter 4 discusses the performance of solution methods and thesis will be finalized with our concluding remarks and further research suggestions in Chapter 5.

Chapter Two

LITERATURE REVIEW

Internet has changed the paradigms of advertising and introduced some concepts which can be considered as a revolution. Before the internet revolution, the advertisers did not have a practical and economical alternative other than the mass advertising. Advertising made by conventional methods (TV's, radios, magazines) were far from targeting a customer range efficiently. Nowadays, by the help of internet, the advertisers are able to reach their target customer more efficiently and with less cost. The most important change of paradigm is *personalized advertisement* opportunities that internet (World Wide Web) provides. TV, radio, newspapers were not able to publish different advertisements for different users due to physical constraints. However, the virtual environment enables the website owners (i.e. publishers) to display different advertisements to the visitors (i.e. users) from their website. Not only the web site owners, but with the ongoing introduction of newer technologies and fast penetration to the daily life of the population mobile phone companies are also looking forward for opportunities of personalized advertisement applications.

Internet recognizes the users through collection of data by two means; either provided directly by the users via questionnaires and registration forms or obtained indirectly from the web log of users. Indirect information taken from online activities of users is considered to be more trustworthy and fruitful. Frequency of visiting, time spent in each page, the advertisements user clicks through, pages which user is interested most, location of users obtained from their IP address are all invaluable information. However, this type of information acquisition, referred to as the *intrusive methods,* is not generally welcomed by public due to privacy concerns.

The systems that are designed with the purpose of advertising the right products to the right user are called *Recommendation Systems* (RS). In the literature, these systems are classified in three groups, namely *the content based systems, the collaborative filtering systems and hybrid approaches*. Content based approaches are designed for advertising a product which is similar to the products that user have shown interest before. Collaborative filtering methods utilize preferences, demographics of the user and advertise a product which similar users are interested before. Hybrid approaches stand in the middle of these. For the interested readers we advise to read [5].

Targeted or Personalized Advertising Systems' popularity is also increasing and becoming a hot research area. There are some commonalities with its logic and RS's as they both strive to advertise the best advertisements to the users. However, they are different in the sense that, PAS try to satisfy the requirements of the publishers and the advertisers simultaneously. Contracts between the advertisers and the publishers limit the PAS's freedom to always publish the most suitable advertisement for the viewers in order to maximize the overall objective. For example, an RS would prefer an advertisement if the user's possibility to purchase the product is high, hence might yield certain commission to the publisher of the advertisement (the web site owner). However, the PAS should display an advertisement if the corresponding advertiser pays more than the rest. That is to say, the objective function and the constraints imposed due to the contract between the publisher and the advertisers change the structure of the problem drastically and lead to a new mathematical structure.

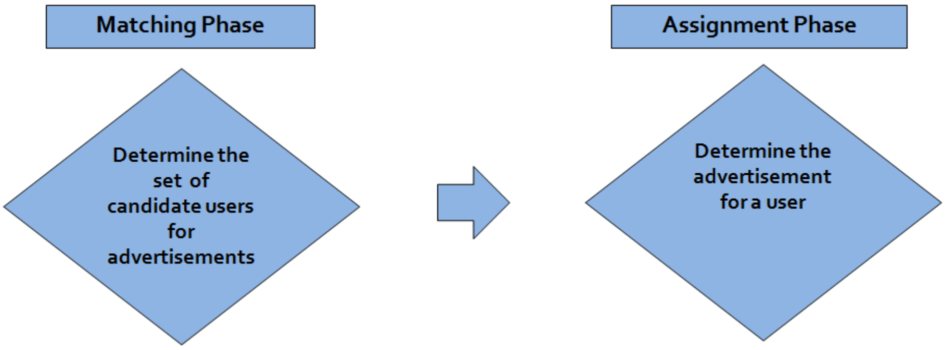


Figure 2.1 A simple representation of personalized advertising problem

As stated earlier, PAS has two phases as represented in Figure 1. The first phase handles the matching of the advertisements and the users (identification of the candidate viewers for the advertisements) and the second one is the assignment (scheduling of the advertisement, i.e. selection of the advertisement whenever the viewer is in the vicinity of the advertisement display location) phase.

Most of the current literature regarding to PAS deals with the matching phase of the problem. Bae et al. [6] uses fuzzy logic in their article for the selection process utilizing the users' preferences and content of the advertisements. No constraints were imposed to the system, only content of the advertisements were taken into account. Kazienko [7] and Zhou et al. [8] provided a framework for personalized advertising problems. However, none of them present a scheduling or assignment method in details. Yager [9] introduces a targeted e-commerce methodology using fuzzy intelligent agents. Algorithm proposed by the author discusses matching problem in details. Bids are done by intelligent agents of the advertisers in this problem and the related scheduling problem is briefly mentioned but no solution was proposed.

There are also various assignment and scheduling methods in internet advertising that does not include matching. Among this, Langheinrich et al.'s [10] article was a fundamental one in the field of advertisement scheduling. ADWIZ system of Langheinrich et al. classifies the users according to the keywords they write and assigns an advertisements using linear programming (LP) approach. LP satisfies the constraints and maximizes the total click rate. Constraints impose a minimum number of exposures for all advertisements and rent of a keyword by an advertisement (e.g. for users entered query of "luxury car"; the advertiser "BMW" can desire that %50 of these users will be shown their advertisements). LP provides information which advertisement to be displayed to the user who writes a particular keyword. ADWIZ system is a semi-dynamic one that it takes click information from assignments and modifies solution accordingly.

There are some successors of the Langheinrich et al. paper in the literature. Tomlin [11] demonstrates that the success of ADWIZ system relies highly on the accuracy of the model parameters *(e.g. click rates)* which is not the case in practice. He concludes that an advertisement selection algorithm must be robust in order to overcome the accuracy problems that would be encountered in real life. Furthermore, Tomlin modifies the objective that it includes both exposure and click revenues. Tomlin finalized his article with a solvable NLP model. Nakamura and Abe [12], the co-authors of Langheinrich et al., tried to improve the ADWIZ model by extensions and render it as a more practical model. All these three articles have different features; though approach for handling problem is similar.

Adler and his colleagues [13] considered a different problem in which advertisements are specified by three attributes: (i) Ad Geometry, (ii) Display Frequency and (iii) Time Interval. Adler et al. defined *ad placement problem* and developed a heuristic for this problem. Amiri & Menon[14] formulates the problem of Adler et al.'s advertising scheduling problem by mathematical programming. The problem is decomposed to small problems and a heuristic is developed which is using the results of these small problems. Besides these; Kumar et al.[15] proposes a new model for the click behaviours. Kumar et al. claims that the click probability is composed of two different factors, namely, the exposure effect and the re-click effect. The authors design a heuristic that publish advertisements for period t, then according to the click results; new advertisements are published in period $t+1$.

All articles are useful as they suggest distinctive solution approaches for different problems. On the other hand, [13],[14] and [15] are irrelevant as their focus is to select the most efficient set of advertisements to place in an advertising location at a single exposure.

However, in the 3D platform that we focus, only one advertisement is allowed to be displayed from a single advertising location at each exposure, i.e. no mixed advertising. In [10],[11] and [12], the predictions are made using broad terms, e.g. how many times the word "football" is typed or how many times a web page is entered. These broad terms are union of responses of several users. On the contrary, our system has to consider reactions of each individual user separately.

Even though, assignment and scheduling articles are not directly related to our study, a summary of [10], [11], [12], [13], [14], [15] are provided for interested readers below. Non-interested readers can skip to next section.

Langheinrich et al. [9] is a seminal paper in the field of advertisement scheduling and assignment problems. In their paper, Langheinrich and his colleagues proposes a system called ADWIZ for the assignment of the advertisements to the users. ADWIZ system of Langheinrich et al. classifies the users according to the keywords they write and assigns an advertisement in order to maximize click rate. ADWIZ system is illustrated in figure 2.1.

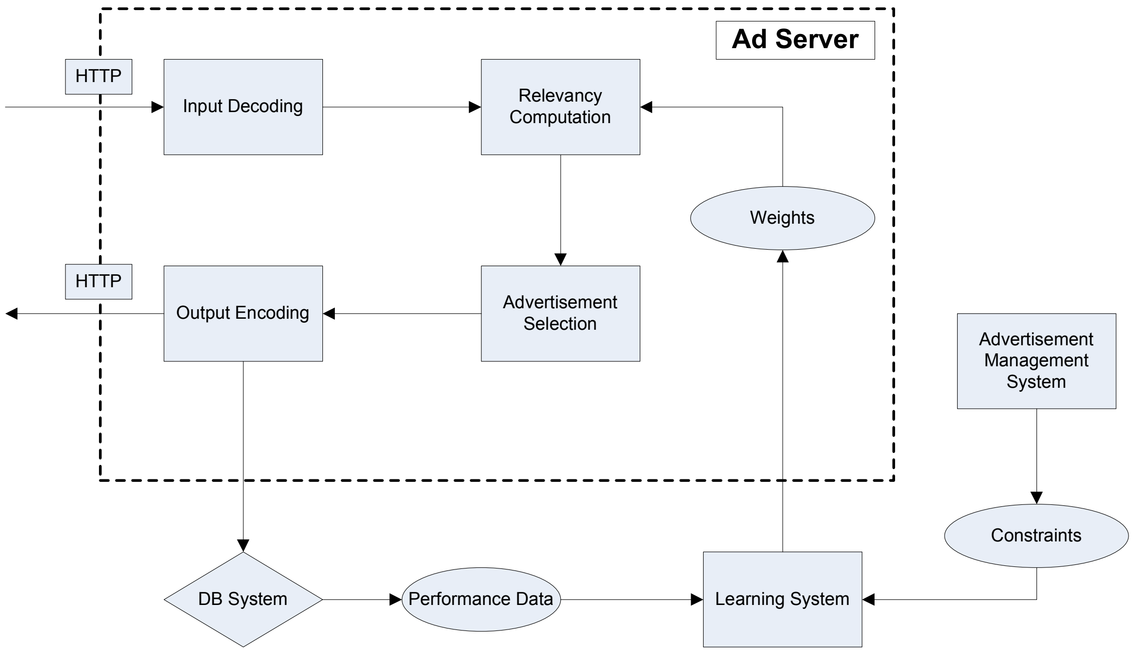


Figure 2.2 A scheme of ADWIZ system adopted from Langheinrich et al. [9]

In the model, first of all there is a database *(DB system*) that stores the information about the keywords, the ads that will be published and the click-information about each keyword-ad pair. *Performance data* is basically refers to this information. Note that, there are also some constraints which are stored in the *advertisement management system*. These *constraints* are determined during the negotiations between the advertiser company and the customer company (i.e., the publisher). Utilizing both constraints and performance data, *learning engine* determines the weights for each keyword-advertisement pair. After a user logins to the system, the keywords that are typed by the user are captured as the input. By the aid of the *relevancy computation* probabilities regarding to each one of the keywords are calculated using the input and the weight information. These probabilities are utilized by the *advertisement selection* mechanism and a particular one is selected for the user. Advertisement selection process repetitively works until the user logouts from the system. Selected advertisement is the output of the ADWIZ system. Note that every time an advertisement is presented to the user, new information is gathered (clicks or not clicks to the advertisement, how long it takes to click if the user clicks, etc.). This information is stored in the database. The weights are updated periodically where the period is decided by the publisher.

Langheinrich et al. proposes a negotiation model between the publisher and advertisers. According to this model there are two options for the customers that both of them can be used simultaneously. First one is determining a certain amount of exposures to users; let's say N. No specialization (keywords) is required in this option; N exposures are displayed for random keywords. Second one is to rent some portion of a keyword entry e.g for users entered query of "luxury car"; the advertiser "BMW" can desire that %50 of these users will be shown their advertisements.

Let's denote the set of keywords A with n keywords and the set of advertisements B with m advertisements (i.e., $A = \{1, 2, \dots i \dots . n\}, B = \{1,2, \dots , j, \dots m\}$). Langheinrich's objective is to obtain display probabilities *(i.e.* d_{ij}*)* for all keyword-advertisements pairs (*the decision variable)*. Langheinrich et al. developed a Linear Programming (LP) model for this purpose. In their proposed methodology, before running the LP model, some data modification is required.

- For each advertisement, numbers of earlier displays are subtracted from the desired number of displays and all of them are normalized so that $\sum_j h_j = 1$ where h_j refers to as the desired display rate of advertisement j. This step basically converts the desired number of display to display rates.
- By utilizing the historical data the input probabilities of keywords (i.e. k_i) are calculated. Normalization also takes place for this parameter so that $\sum_i k_i = 1$.
- Click rates for keyword-ad pair *(i.e.* c_{ij}*)* are calculated using historical data. It is obtained by dividing the number of clicks to number of exposures for each keyword-advertisement pair up till current time.

Formulation of the ADWIZ model is provided below.

$$maximize \ \sum_i \sum_j c_{ij} d_{ij}$$

$$subject\ to$$

$$\sum_i k_i d_{ij} = h_i ;$$

$$\sum_j d_{ij} = 1;$$

$$d_{ij} \geq 0 .$$

Note that, this model does not include keyword rental constraints. However, they can be easily incorporated to the model. According to the article, the weakness of their model is that using LP, tends to force many d_{ij} to 0. Therefore, some advertisements lose their chance to be displayed for some keywords hence the required data that would be used to calculate the click rates cannot be collected for such pairs ever. A lower bound is introduced in order to avoid such misleading cases: $d_{ij} \geq \frac{1}{2m\sqrt[2]{D_{ij}+1}}$, where D_{ij} is the number of exposures for (i,j) pair so far. Lower bound is specifically the standard deviation of click-through rate c_{ij}.

The proposed approach is evaluated empirically in an artificially simulated environment. The proposed LP model is compared with two other methods, namely, the random selection and the Max Click Rate method which is a greedy algorithm that assigns the available advertisement that has the highest click rate for the keyword that is used by the user. Simulation starts with the selection of a keyword randomly. An advertisement is selected using related method. After that, a new random number is generated in order to check whether this advertisement is clicked. This process is repeated for 1.000.000 times and d_{ij} are re-calculated in every 3125 trials.

In the experimental analysis, the Max Click Rate algorithm outperforms the other three in terms of the total click rate. However, the max click rate algorithm is unrealistic since it neglects the constraints imposed by the advertisers during the negotiations. Linear programming performs better than random selection and satisfies the constraints. One more thing questioned in empirical analysis whether lower bounding would result in an increase of total click rate. It can be concluded that for short intervals (less than 200.000 exposures) pure LP may result better, however otherwise the model with lower bounds is suggested.

There are several successor papers of Langheinrich et al. Among these, Tomlin [10] adds some kind of randomization to the ADWIZ model. Tomlin claims that randomization improves the performance particularly due to the fact that the estimation of click rates is usually inaccurate. He offers a Nonlinear Programming (NLP) model to solve the problem formulation that is presented by Langheinrich et al. Furthermore, Tomlin's model considers the advertising problem as a multi-period problem and allows to carry the ads that are not broadcasted as inventory from one period to another or to show some advertisement more than ordered. Tomlin also modifies the objective function so that it includes both exposure and click revenues unlike the Langheinrich et al. which focus only to the click revenues.

Tomlin demonstrates that the Langheinrich et al. model relies highly on the accuracy of the model parameters which is not the case in practice. He presents a small LP model and demonstrates that the results are misleading if the parameters change slightly in the example. He concludes that an advertisement selection algorithm must be robust in order to overcome the accuracy problems that would be encountered in real life. He also criticizes the lower bounds for ads that they have only a limited benefit.

Tomlin firstly tried to associate ADWIZ model to a one which is well-known in literature. By some modifications; Langheinrich's model is converted to the well-known transportation problem. Procedure is described below step by step.

- A new parameter is defined called ρ_{ij} and it is located in place of $k_i d_{ij}$. The following model obtained is the classical transportation problem.

$$\text{maximize} \sum_i \sum_j c_{ij}\rho_{ij}$$

$$subject\ to$$

$$\sum_j \rho_{ij} = k_i \ \ \forall i \qquad (1)$$

$$\sum_i \rho_{ij} = h_j \ \ \forall j \qquad (2)$$

$$\rho_{ij} \geq 0 \qquad \forall(i,j) \qquad (3)$$

- Tomlin introduces a model used in traffic theory. The model is used for the distribution of vehicles from destination (keywords) to sinks (ads). Destinations are denoted by i and sink points are denoted by j. A_i is the number of trips started from node i, B_j is the number of trips ended in node j. The importance of the formulation below stems from its similarity to the model above. Note that, constraints 1, 2 and 3 are same with constraints 4, 5 and 6. Only difference is that A_i and B_j have to be integers whereas k_i and h_j are rational numbers between 0 and 1. Tomlin claims that vehicles can be considered as advertisements and the integer model below can be used for our advertising problem.

$$\max \sum_i \sum_j c_{ij}x_{ij}$$

$$\sum_j x_{ij} = \ A_i \ \ \forall i \qquad (4)$$

$$\sum_i x_{ij} = \ B_j \ \ \forall j \qquad (5)$$

$$x_{ij} \geq 0 \qquad \forall(i,j) \qquad (6)$$

$\sum_i A_i = \sum_j B_j = X$ for feasibility where X is the total number of trips.

- Tomlin modified objective function since his aim not to maximize click-rate at all. For the time being, Tomlin assumed a target of click-rate is satisfied.

$$\sum_i \sum_j c_{ij} x_{ij} = C$$

- Tomlin's aim is distributing something more evenly. More distributed decision variables are better. Therefore, he formulates a function that measures how evenly these decision variables are distributed and denotes it by $w(x_{ij})$.

$$w(x_{ij}) = X! \Big/ \prod_{ij} x_{ij}$$

- An assumption here is made that the distribution of parameters is proportional to this function; therefore, this function has to be maximized. It is equivalent to find maximum of $\log(w)$. By the aid of Stirling's formula and neglecting constant terms; the function, we need to deal with has to be:

$$\max - \sum_i \sum_j x_{ij} \ln x_{ij}$$

- There are two objectives of Tomlin: To distribute more evenly and still obtain a good level of click rate. Below function is good for distributing more evenly however might lack of obtaining a good click rate. Tomlin defines a cost function which evaluates how much an advertisement gets far from best clickrate for any keyword *(i.e.* $\bar{c}_{ij} = \left[max_{pq} c_{pq}\right] - c_{ij}$ *).* Cost increases when $\bar{c}_{ij}$ value increases. Tomlin's model has to minimize this value to obtain good click rates. Minimizing this function is equivalent to maximizing c_{ij}. Therefore; Tomlin constructs an objective function which includes all considerations.

$$\max \sum_i \sum_j x_{ij} \left(\mathrm{c_{ij}} - \gamma \ln x_{ij}\right)$$

γ is determined by the user. When $\gamma = 0$; that means the user trusts the estimations on hand; no need to distribute evenly. Increase of γ demonstrates more emphasis is given distributing the x_{ij} evenly.

Using this objective function Tomlin constructed an NLP with some extensions to the Langheinrich's model. Besides the revenue obtained from click rates, revenue by impressions is added to model. As number of impressions has to be equal to the number of orders taken; carrying inventory is enabled. For excess and shortfall of shown ads; penalty costs are defined. Upper limits for number of impressions are incorporated to the system. Furthermore, marketing costs of agencies added to the objective function. Details are provided clearly in the mathematical model; interested can view [10] .

Tomlin claims that the problem can be solved by Generalized Benders Decomposition; he did not show any example or present some results by an empirical evaluation. He says in conclusion that this is the next step, however he does not implement such a job up-to-date.

Nakamura and Abe [11], the co-authors of Langheinrich et al., tried to develop the ADWIZ model by extensions and render it as a more practical model. In the article, clustering is suggested for better estimates of click rates and Gittins Index, Semi on-line learning using an interior point method and Lower Bounding are suggested considering exploration-exploitation trade off. Exploration is for obtaining new information about click rates that it enables to update the click rates of less popular ads. Exploitation is dealing with obtaining revenue. The article also covers the problem of multi-impressions and inventory management. All of these will be covered in details in the following paragraphs.

In Langheinrich et al. assignments are between keywords and advertisements. Nakamura and Abe make use of the same model for web-pages and advertisements. Web-pages are classified using attributes like sports, economy or technology. Some attributes depends on the time of the arrival. e.g. if a user visits a sports page in the morning; then the attribute combination of web-page is simply sports and morning. Therefore, authors first propose to utilize attribute combinations rather than keywords. Therefore, web-pages are assigned advertisements using attribute combinations. They state that the importance of users may change according to advertiser. They offer a new objective function for this case.

- $\sum_i \sum_j g_j c_{ij} d_{ij} k_i$ where g_j is the importance of customer j determined by the advertiser.

Authors state that at the very first stages of an advertising period data may be sparse. Therefore, estimations can be deceiving. Authors suggest a process called clustering for to avoid negative effects of data sparseness. Let's consider there are n attribute combinations (i) and m advertisements (j). The idea is to merge attribute combinations and obtain more healthy data for estimation of click rates. The decrease in attribute combinations also enables a decrease in computational time of LP. Firstly a measure is defined which measures the performance of clusters. Authors claim that by minimizing $I(\theta\rho) = LL(\theta\rho) + PT(\theta\rho)$, best clusters can be achieved where $LL(\theta\rho) = \sum_{Z \epsilon \rho} \sum_j -(C_{Z,j} \log \frac{C_{Z,j}}{D_{Z,j}} + (D_{Z,j} - C_{Z,j}) \log \frac{D_{Z,j} - C_{Z,j}}{D_{Z,j}}$ and $PT(\rho) = m|\rho|$.

Authors state that since it is infeasible to minimize $I(\theta p)$; a greedy algorithm is devised for this algorithm.

Nakamura and Abe's second issue is to resolve exploitation and exploration trade-off. Exploitation is simply utilizing the click-rates on hand and exploration is for obtaining new information about click-rate information. Three methods are suggested for exploitation and exploration trade-off. 1) Gittins Index 2) Semi on-line learning using an interior point 3) Lower bounding. Gittins Index is imported from Bayesian theory of Bandit problems. This method is implemented by using an objective function includes Gittins Index.

$\sum_i \sum_j G(C_{i,j}, D_{i,j} - C_{i,j}) k_i d_{i,j}$ where $C_{i,j}$ is the number of clicks for (i,j) pair in historical data and $D_{i,j}$ is the total number of exposures for pair (i,j) in historical data. Assuming $\alpha = C_{i,j}$ and $\beta = D_{i,j} - C_{i,j}$; the value of Gittins Index which is equal to a particular p value can be found from the following equation.

$$\frac{p}{1-\gamma} = \frac{\alpha}{\alpha+\beta}\left(1 + \gamma\, R(\alpha+1, \beta, p)\right) + \frac{\alpha}{\alpha+\beta} \gamma\, R(\alpha, \beta+1, p)$$

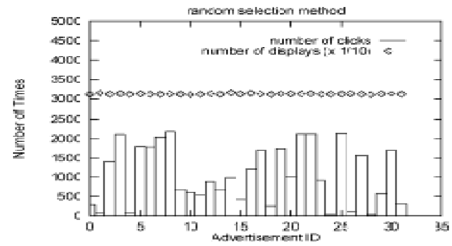
$$where\ R(\alpha, \beta, p) = \max\left(\frac{p}{1-\gamma}, \frac{\alpha}{\alpha+\beta}\big(1 + \gamma\ R(\alpha+1, \beta, p)\big) + \frac{\beta}{\alpha+\beta}\gamma\ R(\alpha, \beta+1, p)\right)$$

and γ is a constant determined by publisher.

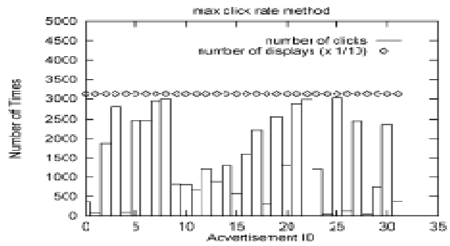
Semi-online learning using an interior point method, improves the previous solution step by step rather than optimizing in each learning interval. The weights are determined by a convex combination of previous solution and current optimum solution. Lower bounding is exactly what is proposed by Langheinrich et al., by using lower bounds for every (i, j) pair enough exploration is aimed.

Another issue that the authors deal is multi-impression case. In some web pages; there may be more than one advertising area might exist. Multi-impression of an advertisement may not be desired for both advertiser and publisher. Selection algorithm is unable to assign different advertisements to advertising areas. Therefore, authors develop an algorithm which includes a queue that selected but unable to be shown advertisements are stored in. Furthermore, they state that an upper bound has to be defined for display probabilities (d_{ij}) to prevent ad jam in queue. Upper bound is useful in the sense that it increases number of alternatives for any *(i,j)* pair. Inventory management is another important issue because number of impressions and number of contracted impressions have to be in balance. Assignment of advertisements needs some attention; idle impressions have to be assigned efficiently to maximize the utilization of advertising slots. A simple LP is defined for this approach that has structure resembling transportation problem. Solutions are used as an input to the main LP that determines display probabilities.

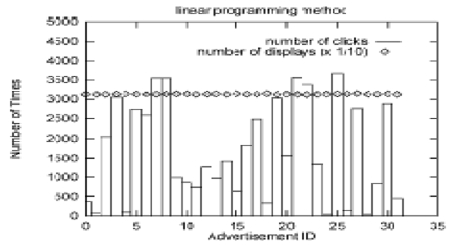
Click rates and related parameters are generated randomly and incorporated into 5 different scenarios with different learning interval lengths. Desired numbers of impressions are assumed equal for all advertisements. Simulation is performed with the same methodology like in Langheinrich's. Three approaches compared at first; random selection, maximum selection and LP selection. Random selection assigns the advertisements randomly from the set of advertisements that the display number has not been reached yet. When advertisement hits desired display number; algorithm discards this ad from available ads. Maximum selection differed from random selection in the way advertisement is selected. The advertisement with maximum click rate is assigned. Comparing three alternatives; LP performs better.



(a) The random sel. method



(b) The maximum sel. method



(c) Linear programming method

Figure 2.3 Performance charts for methods proposed from Nakamura et. al. [12]

Secondly, the proposed methods compared. 5 different approaches are simulated. First one is vanilla approach which is a pure LP approach; that none of extensions is used. Second one is LP with clustering. Third, fourth and fifth are LP with three proposed methods for exploitation and exploration trade off; LP with Gittins Index, LP with Interior Point and LP with Lower Bounding respectively. For a short time, Vanilla seems to be best option; however all of the other approaches are better in the long term.

Lower bounding is the best for cumulative storing of click-rate values; however clustering performs better in instantaneous case. In instantaneous case; last 250000 impressions are stored. According to the results above instantaneous storing outperforms cumulative storing.

Adler et al. [13] considered a problem in which advertisements are specified by three attributes: (i) Ad Geometry, (ii) Display Frequency and (iii) Time Interval. Ad Geometry determines the dimensions of the ads which is located in the advertising area.

Note that, the advertising area is 2D, therefore the advertisement geometry has to be specified in two dimensions. Display Frequency specifies how much of the users (or hits) a particular ad is exposed to. Time Interval determines the length of one exposure of an advertisement.

The authors refer to the problem that includes all of the three attributes as the "Full Assignment Problem". According to the problem statement, a set of ads A is assigned in a way that maximizes utilization of spaces available. Let T be the number of available advertisement spaces. The value of T is obtained from a function of display frequency values which we will discuss later. The objective of "Full Assignment Problem" is to select a subset of ads A' ($A' \subseteq A$) that gives the best utilization of the advertisement spaces. Adler et al. introduced a new bin packing problem which is referred to as the "ad placement problem". In their formulation, given T slots, a slot size S, and a set of ads A, where each ad $i \epsilon A$ consists of a size $s_i \leq S$ and a weight $w_i \leq T$, assign the ads to the slots, where ad i is assigned exactly once to each of w_i slots. Slot is similar to what is called space above in Full Assignment Problem; however, slot is 1-dimensional in ad placement problem. In short, we can state that ad placement problem is a relaxed version of Full Assignment problem in terms of dimensions. Let $P(j)$ be the set of ads that are assigned to slot j, and let the fullness of slot j be $|P(j)| = \sum_{j\ i \in P(j)} s_i$. An assignment is valid if $max_j |P(j)| \leq S$. Difference of the ad placement problem from the traditional bin packing problem is that at most one copy of ad i can be placed in one slot. Three attributes for advertisements i are denoted as follows;

- $U(i)$ - Access Fraction : Probability of a user is exposed to the ad i during her access. (i.e. the fraction of hits to a web page) e.g if $U(i)$ is 0,5 and total number of hits is 120; the number of exposures ad i has to be 60.

- $F(i)$ – Time Fraction : The time required for advertisement i in terms of a fraction of the time for a predetermined time limit. e.g. if an advertisement requires 5 seconds and time limit is 20; then time fraction will be 5/20=0.25.

- $L(i)$ - Geometry : Size of the ad in terms of its physical dimensions.

To make use of the "ad placement problem" with its most basic properties where S is one dimensional; one of the attributes has to be ignored. The authors refer to this approach as the 2-approximation problem. Finding a solution for ad placement problem enables us to solve the 2-approximation problem. The assumptions of the problem are presented below.

- Time fractions are assumed to be equal to 1 for all of the advertisements.
- Advertising space is fixed.
- Widths of ad geometries are assumed to be equal to the width of the advertising area Therefore; the geometry of the problem becomes one dimensional. Only the height of the advertising area has to be distributed among advertisements which is equal to S. This results with defining $L(i)$ in height only. Consider an advertising area with height 320 pixels and width 100 pixels. As width is assumed as equal for both advertisements; the concern is only about height. 320 pixels have to be distributed in a way consistent with objectives.
- T has to be decided. T is the least common multiple of denominators of access fraction. For example, Let A, i.e. the set of advertisements, be $A = \{1,2,3\}$ and $U(1) = 1/2$, $U(2) = 1/3$ and $U(3) = 1/4$; then T is least common multiple of 2, 3 and 4. That results 12.

Adler et al. proves that "ad placement problem" is NP-Hard. As this problem is NP-Hard; solution cannot be obtained in polynomial time. That results in massive computation time for increasing number of dimensions (e.g. number of advertisements) Therefore, they seek for a heuristic which works efficiently for this problem. In the "ad placement problem" the ad sizes are assumed to be divisible (when ad sizes form a sequence $S = p_1 > p_2 > p_3$ ... such that for all k, p_k is an integer multiple ofp_{k+1}.). The proposed algorithm is called LSLF (Largest Size – Least Full) that always gives the optimal solution whenever there is a solution that all ads in set A is assigned to the schedule. LSLF firstly sorts all advertisements by size from largest to smallest, then locates them to least full w_i slots starting from largest to smallest by size.

Adler et al. introduces two versions of the solution methodology, namely the offline and the online versions. In the offline version (i.e., the static version) all of the ads are assumed to be known before the scheduling takes place, in other words; the set A is known completely before the decision making process. In this version of the methodology, a subset of ads A' is selected and scheduled using the algorithm SUBSET-LSLF. Adler et al. provides an algorithm called SUBSET-LSLF for the static problem.

- Let $B_s = \sum_{i,s_i=S} Sw_i$. Let $B_{\bar{s}} = \sum_{i,s_i<S} s_i w_i$

If $B_s \geq B_{\bar{s}}$, then

- Place the ads of size S in order by weight, discarding any ad that, when placed, would violate the size limit for some slot.

- On the ads with size $< S$, run algorithm LSLF, discarding any ad that, when placed, would violate the size limit for some slot.

If $B_s \leq B_{\bar{s}}$, then

- Place the ads of size S in order by weight, discarding any ad that, when placed, would violate the size limit for some slot.

- The set A' is the set of ads that are assigned to slots.

Adler et al. proves that its worst case bound is 2 times the optimal solution when divisibility assumption is made for ad sizes. This approach can be criticized to be restrictive since divisibility assumption is imposed to the ad sizes. However, the authors claim that their algorithms increase the flexibility of internet advertising as their algorithm provides the opportunity to split advertising area. Note that in the paper experimental analysis is not provided.

On the other hand, for the online version of the problem, there are incoming orders during scheduling. Every time an order is received, a decision is to be made regarding to whether to accept or reject the order. The author provides some examples and demonstrates that the deterministic algorithms work poorly for online version of the problem. Therefore, authors impose an upper bound of ad size Z and an upper bound for ad weight V, so that $2ZV < ST$. They call their algorithm OL-LSLF. According to this algorithm a new customer is accepted if there is a valid schedule with the advertisement given. For any sequence of ads C , authors prove that their algorithm's worst case bound is $ST/\,ST - Z(2V - 1)$ times of optimal solution. This worst case bound is proved to be the best among the other algorithms presented in the paper mathematically, so OL-LSLF is claimed to be optimal.

Amiri & Menon [14] formulates the problem of Adler et al.'s advertising scheduling problem by mathematical programming. They assume that the time fractions for different ads are equal and revenue is based on only number of impressions (CPM) similar to the problem introduced in Adler et al. Amiri & Menon first construct the IP model for the advertising problem. They call it advertising placing problem (APP). In APP, geometry is assumed as one dimensional and space available for advertising is assumed to be fixed. The time fractions are assumed to be equal for both ads. In the *advertisement placing problem*, there is again a set of time slots T and a slot size S similar to [13]. Each ad i has a size $s_i \leq S$ and a weight $w_i \leq |T|$. Authors assume that the revenue is proportional to the weight of the ad, therefore they want to maximize $\sum_i s_i w_i$.Two decision variables are defined:

- x_{ij} is 1 if ad i assigned to slot j, otherwise it is 0.
- y_i is 1 if ad i assigned to any slot, otherwise it is 0.

The formulation is as follows;

$$maximize \sum_{i \in A} \sum_{j \in T} s_i x_{ij}$$

$$subject\ to$$

$$\textstyle\sum_{i \in A} s_i x_{ij} \leq S \quad \forall j \in T \qquad (1)$$

$$\textstyle\sum_{j \in T} x_{ij} = w_i y_i \quad \forall i \in A \qquad (2)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in A\,; \ \forall j \in T \qquad (3)$$

$$y_i \in \{0,1\} \quad \forall i \in A \qquad (4)$$

The structure of the formulation is changed by Lagrangean Relaxation. For this purpose, first the constraints 1 and 2 are de-linked by defining a new set of variables denoted as v_{ij}. The following model is obtained;

$$maximize \sum_{i \in A} \sum_{j \in T} s_i x_{ij}$$

$$subject\ to$$

$$x_{ij} - v_{ij} = 0 \quad \forall i \in A\,; \ \forall j \in T \quad (5)$$

$$\textstyle\sum_{i \in A} s_i x_{ij} \leq S \quad \forall j \in T \qquad (6)$$

$$\textstyle\sum_{j \in T} v_{ij} = w_i y_i \quad \forall i \in A \qquad (7)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in A\,; \ \forall j \in T \qquad (8)$$

$$v_{ij} \in \{0,1\} \quad \forall i \in A\,; \ \forall j \in T \qquad (9)$$

$$y_i \in \{0,1\} \quad \forall i \in A \qquad (10)$$

Model above is referred to as the *APP1*. Note that, in the paper, a second model with a different objective function ($\sum_{i \in A} s_i y_i w_i$) is also proposed as an alternative and referred to as the *APP2*. The Lagrangean relaxation is implemented to *APP1* after this step and the obtained model is referred to as the *APP1L*.

$$Z_L(\lambda) = max \textstyle\sum_{i \in A} \sum_{j \in T} s_i x_{ij} + \sum_{i \in A} \sum_{j \in T} \lambda_{ij} x_{ij} - \sum_{i \in A} \sum_{j \in T} \lambda_{ij} v_{ij}$$

$$subject\ to$$

$$\textstyle\sum_{i \in A} s_i x_{ij} \leq S \quad \forall j \in T \qquad (11)$$

$$\sum_{j \in T} v_{ij} = w_i y_i \quad \forall i \in A \qquad (12)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in A\,; \ \forall j \in T \qquad (13)$$

$$v_{ij} \in \{0,1\} \quad \forall i \in A\,; \ \forall j \in T \qquad (14)$$

$$y_i \in \{0,1\} \quad \forall i \in A \qquad (15)$$

APP1L can be decomposed to two independent problems *SUB1* and *SUB2*.

(SUB1)

$$max \sum_{i \in A} \sum_{j \in T} s_i x_{ij} + \sum_{i \in A} \sum_{j \in T} \lambda_{ij} x_{ij}$$

$$subject\ to$$

$$\sum_{i \in A} s_i x_{ij} \leq S \quad \forall j \in T \qquad (16)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in A\,; \ \forall j \in T \qquad (17)$$

(SUB2)

$$max - \sum_{i \in A} \sum_{j \in T} \lambda_{ij} v_{ij}$$

$$subject\ to$$

$$\sum_{j \in T} v_{ij} = w_i y_i \quad \forall i \in A \qquad (18)$$

$$v_{ij} \in \{0,1\} \quad \forall i \in A\,; \ \forall j \in T \qquad (19)$$

$$y_i \in \{0,1\} \quad \forall i \in A \qquad (20)$$

SUB1 and *SUB2* are decomposed further. *SUB1* is decomposed *SUB1j's* and *SUB2* is decomposed to *SUB2i's*.

(SUB1j)

$$max \sum_{i \in A}(s_i + \lambda_{ij})x_{ij} + \sum_{i \in A}\sum_{j \in T} x_{ij}$$

$$subject\ to$$

$$\sum_{i \in A} s_i x_{ij} \leq S \quad (21)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in A \quad (22)$$

(SUB2i)

$$max - \sum_{i \in A}\sum_{j \in T} \lambda_{ij} v_{ij}$$

$$subject\ to$$

$$\sum_{j \in T} v_{ij} = w_i y_i \quad (23)$$

$$v_{ij} \in \{0,1\} \quad \forall j \in T \quad (24)$$

$$y_i \in \{0,1\} \quad (25)$$

SUB1j is a binary knapsack problem. There are various approaches in the literature to solve this problem. On the other hand, *SUB2i* can be solved by a heuristic.

First, the v_{ij} have to be sorted in descending order of ($-\lambda_{ij}$) and compute $f_i = \sum_{j \in A_i} -\lambda_{ij}$, where T_i is the set consisting of the first w_i variables. If $f_i > 0$, the optimal solution is $v_{ij} = 1\ \forall j \in T_i, v_{ij} = 0\ \forall j \notin T_i\ and\ y_i = 1.$ If $f_i \leq 0$, then optimal solution is $v_{ij} = 0\ \forall j \in T, and\ y_i = 0.$ Calculations of λ_{ij} are done through subgradient optimization. Authors propose a five-step algorithm for finding results.

- Sort the ads in decreasing order based on f_i , the solutions to *(SUB2i)* for each ad $i \in A$.
- Start with the first ad in the sorted list.
- Check whether assigning this advertisement to the w_i time slots recommended by *(SUB2i)* is feasible (i.e., check whether assigning this ad will keep the space used in

each of the w_i slots within the available space S). If yes, assign this ad to the appropriate time slots.

- Remove this ad from the sorted list.
- Repeat Steps 2 and 3 until the sorted list is empty.

Authors compare the performance of the two objective functions in *APP1* and *APP2* in their paper. Based on the analysis, *APP1* yields better upper bounds for the *APP*. On the other hand, the *APP2's* objective function provides better feasible solutions. The experimental analysis is conducted on data-sets presented in Kumar et al. [16]. Upper bounds of *APP1*, the best feasible solutions of the algorithm proposed by Adler et al., Kumar et al. and the Lagrangean Decomposition *(APP2)* are compared in the analysis. The analysis demonstrates that the Lagrangean Decomposition *(APP2)* performs better in all cases in terms of the upper bounds. However, the tradeoff is the cost of the computational time.

Besides these two different approaches, namely the Adler et al. and Amiri and Menon, Kumar et al. [15] proposes a new model for the click behaviours. Kumar et al. claims that the click probability is composed of two different factors, namely, the exposure effect and the re-click effect. The exposure effect has a second degree formulation that has a U shape curve; $e_{ik} = -a_{i1}k + b_{i1}k^2 + a_{i0}$. This formulation is proved by a study before by [17]. Principle behind re-click effect is that an individual clicks an ad is more likely to click again in subsequent exposures. No presumption is made for advertisements or users; it only uses click behavior of an individual. However, it is personalized that scheduling is implemented independently for every individual user.

Three types of constraint are incorporated in solution model. *Size constraints* are dealing with the placement of advertisements in advertising area. *Exposure constraints* state that at most one copy can be replaced in one slot. *Pairwise ad constraints* determine the ads to be displayed simultaneously in the same advertising slot (inclusion) or distinctively in different advertising slots (exclusion). A hybrid pricing scheme is used in model. Mathematical model is given and concluded that the processing times can be substantially long. In figure 2.6 mathematical model is provided.

As, they state that computational time required is long; they offered heuristics for static and dynamic cases. In static case, click data is not stored; the click probabilities are calculated considering the probability of click-rate of previous exposures. For example click probability of advertisement i's second exposure (C_{i2}) is calculated from the formula $C_{i2} = C_{i1}(e_{i2} + p) + (1 - C_{i1})e_{i2} = pC_{i1} + e_{i2}$ where p is the probability added because of re-click effect. Then a general formula is constructed below.

$$C_{ik} = \left\{1 - \prod_{l=1}^{k-1}(1 - C_{il})\right\}p + e_{ik}$$

For the heuristic of static case; firstly an LP is defined. This problem is called *MKP(j)* and it is a modified knapsack problem.

$$MKP(j) = Maximize \sum_i m_{ij}x_{ij}$$

$$subject\ to$$

$$x_{uj} + x_{vj} \leq 1; (A_u, A_v) \epsilon\ D_{ex}$$

$$x_{uj} - x_{vj} \leq 0; (A_u, A_v) \epsilon\ D_{in}$$

$$\sum_i s_i x_{ij} \leq S$$

$$x_{ij}\ \epsilon\ \{0,1\}, i = 1,2, \dots, n$$

where m_{ij} is the expected revenue if advertisement i is published in slot j.

Assuming total number of slots as N; Successive Slot Knapsack*(SSK)* heuristic is defined for the problem.

(Algorithm SSK)

Step 1: Define $m_{i1} = r_{i1}\ ; i = 1,2, \dots, n. Set\ j = 1.$

Step 2: Solve problem *MKP(j)*.Let x_{ij}, $i = 1,2, \ldots, n$, be an optimal solution for *MKP(j)*. Set $j = j + 1$.

Step 3: If $j \leq N$, update the values of m_{ij}, $i = 1,2, \ldots, n$, as follows $m_{ij} = r_{ik}$, where $k = \sum_{q=1}^{j-1} x_{iq} + 1$ and go to Step 2; otherwise, terminate.

SSK heuristic is tested for different values of n, N and S. For n; 10, 50 and 100 are considered as realistic. For N; 10, 30 and 50; for S; 20 and 40 are selected. Therefore there exist 18 (3x3x2) combinations of problems in the test bed. Beside of these; there are 4 click parameters (a_{i0}, a_{i1}, b_{i1}, p) and 2 revenue (a_{i2}, b_{i2}) parameters to be determined. a_{i0}, a_{i1} and b_{i1} are determined using the data collected from a real, high traffic web site. For p; there different values (0.001, 0.02 and 0,4) are determined. For $a_{i2}$and b_{i2} two different values assigned for each so that 12 (3x2x2) instances are constructed for each problem. Total number of two types' pairwise ad constraints are selected to be 0.1 of total number of ads. Optimality gap is calculated for 18 problem types and average is calculated using both constraints. Results are provided [15].

In the dynamic case; heuristic is a bit different. Assuming the number of slots as N; heuristic starts with solving the static problem (LP). First slot of the solution is used as a solution for slot 1 and published. According to the click information; click probabilities are updated. Then static problem is solved again with the updated parameters. Again the first slot used as a solution for slot 2 and published. This process continues until N slots are solved. Authors name this heuristic as a look-ahead dynamic algorithm. Moreover, authors propose a myopic dynamic algorithm that is similar to SSK Heuristic. Only difference is the calculation of m_{ij} values. Dynamic algorithm is advantageous that it can utilize the click-behaviour of users dynamically. Empirical results also show that dynamic approach is better than static one. Look-Ahead Dynamic Algorithm clearly outperforms static SSK algorithm. Look-Ahead Dynamic Algorithm is also better than myopic dynamic algorithm with the cost of computational time.

N	number of slots.
n	number of ads.
S	height of a slot.
D_{in}	set of pairs of ads on which an inclusion constraint is imposed: $(A_u, A_v) \in D_{in}$ implies that ad A_u is exposed in a slot only if ad A_v is also exposed in that slot.
D_{ex}	set of pairs of ads on which an exclusion constraint is imposed: $(A_u, A_v) \in D_{ex}$ implies that at most one of A_u and A_v can be exposed in a slot.
R_{ik}	expected revenue, per unit size, from the k^{th} exposure of ad A_i.
r_{ik}	expected revenue from the k^{th} exposure of ad A_i; $r_{ik} = s_i R_{ik}$.
T_{ik}	expected revenue from k exposures of ad A_i; $T_{ik} = \sum_{l=1}^{k} r_{il}$.
z_{ik}	1 if ad A_i is exposed a total of k times; 0 otherwise.
x_{ij}	1 if ad A_i is scheduled in slot j; 0 otherwise.

$$\text{Maximize} \quad \sum_{i=1}^{n} \sum_{k=1}^{N} T_{ik} z_{ik}$$

subject to

$$\begin{aligned}
\sum_{j=1}^{N} x_{ij} &= \sum_{k=0}^{N} k.z_{ik}; \quad i = 1, 2, \ldots, n \\
\sum_{k=0}^{N} z_{ik} &= 1; \quad i = 1, 2, \ldots, n \\
x_{uj} + x_{vj} &\leq 1; \quad j = 1, 2, \ldots, N; \ (A_u, A_v) \in D_{ex} \\
x_{uj} - x_{vj} &\leq 0; \quad j = 1, 2, \ldots, N; \ (A_u, A_v) \in D_{in} \\
\sum_{i=1}^{n} s_i x_{ij} &\leq S; \quad j = 1, 2, \ldots, N \\
x_{ij} &\in \{0, 1\}, \quad i = 1, 2, \ldots, n; \quad j = 1, 2, \ldots, N \\
z_{ik} &\in \{0, 1\}, \quad i = 1, 2, \ldots, n; \quad k = 1, 2, \ldots, N
\end{aligned}$$

Figure 2.4 - Mathematical Model presented in Kumar et. al. [15]

Chapter Three

PROBLEM DEFINITION AND SOLUTION METHODS

In this chapter, first a business model for the software developer company that was mentioned earlier will be introduced. Next, we will present the mathematical model of the problem. Finally the proposed solution alternatives will be discussed.

3.1 Business Model

The company (i.e., the *publisher*) provides the medium between the users (i.e., *viewers*) and the advertisers. Advertisements are displayed to the users from various advertisement locations in the virtual platform developed by the company. Advertisers are seeking for effective advertising opportunities that enable them to reach their target customers with a smaller budget. On the other hand, the publisher is trying to maximize their total revenue and maintain its attractiveness for the advertisers and the users simultaneously. Users are seeking for a nice environment that they can spend quality time. One can safely assume that the web site has a stable traffic and no difficulty to attract users. Our first objective was to develop a business model that is going to attract the advertisers with flexible advertising opportunities. Various possible alternatives were generated and presented to the company and the final decision was settled together with them.

The developed business model offers a quite satisfactory *personalized advertisement* opportunity for the advertisers. The advertisers make payments according to the users' matching degree *(i.e. compatibility score)* for their advertisement. Figure 2, depicts an example of the payment scheme. In this example, a constant price scenario is depicted, in which there are 4 threshold levels and the advertiser promises to pay 5¢ for *compatibility scores* in the interval [0.3, 0.7); 8¢ for the interval [0.7, 0.9) and 13¢ for [0.9, 1]. Note that the pricing for different compatibility scores are settled through negotiations between the advertiser and the publisher. On the other hand, each click through has a constant price which will again be determined through negotiations.

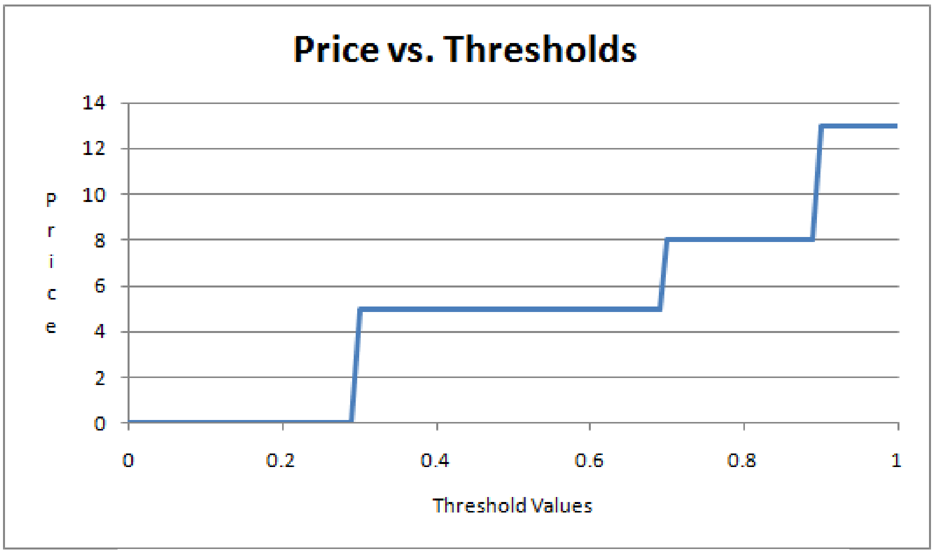


Figure 3.1 - An example pricing scheme

According to the developed business model, the publisher (i.e., the company) presents a list of possible features and clustering information associated with each one of the features to the advertiser and the advertiser sets the specifications based on their target viewers. The features provided by the publisher include a set of certain demographics such as age, income level, education level, etc., certain interests such as likes traveling, books, fashion stuff, etc. and certain preferences such as logins mostly at nights, stays short, likes meeting new users, etc. Note that some of the data is nominal, e.g. gender = {Female, Male}, some of them are ordinal, e.g. likes travelling, hates travelling, etc. , and some of them are interval, e.g. age, income level, etc. The advertisers are allowed to specify constraints that are either based on crisp sets, e.g. "Advertise this one only to the male users" or fuzzy sets, "Advertise to the users that are young, have middle income and likes travelling a lot". Note that the advertisers should also have the right to specify "Don't care" to some of the attributes and unified fuzzy sets such as "young OR middle-aged". The compatibility scores are calculated by blending the users' attributes with advertisers' specifications.

One of the main concerns of the advertisers for exposure of the advertisements is the *maximum display number* and *minimum display number* per individual users. The very same advertisement that keep popping up every corner in a virtual web site could be quite annoying for the user and not desirable both for the publisher and the advertiser. The advertiser wants to make sure that the message is transmitted to the user. However extensive repetition of the advertisement has no effect after a certain point (i.e. advertisements start to *wear out*) and there is no need to make payment to the publisher if the message is already received by the user. Therefore setting a *maximum display number* per person is common in practice. On the other hand, a single exposure might not guarantee that the goal of the advertisement is attained. It is quite possible that the user won't pay attention at the first couple of times and totally miss the advertisement, whereas some repetition would enhance the viewer's ability to remember the advertisement in future. Therefore a *minimum display number (i.e.* $MinDisplay_j$ *for ad j)* is also a desirable constraint for the advertisers. Note that, whenever the advertiser sets a minimum display number, the publisher would be entitled for a payment after an exposure only if the advertisement is displayed at least $MinDisplay_j$ times to the specific viewer. Besides, exposure is not the only source of income and *click through* activities are separately priced by the publishers. Click through substantiates that the viewer actually paid attention to the advertisement and depending on the situation might result with a real sales at the point. Therefore, whenever a click-through is realized, a separate payment for the click is incurred by the publisher.

Naturally, there is a *maximum payment* for every advertiser. The contracts would specify the amount of maximum payment which will specify the level that the advertiser would not pay even if the total revenue of the publisher from the exposures and the clicks would exceed that amount. On the other hand the advertisers also desire that the publisher guarantees a certain number of displays so that they ensure to target a critical mass which is required to initiate a word of mouth effect. This constraint is referred to as the *minimum payment* and the advertiser only makes payment only if their advertisement is published more than a certain number of viewers (i.e., more than a threshold level). Note that it is always possible to set the *trivial* quantities for all of the above mentioned constraints, that is to say 0 for the *minimum display number* and *minimum payment*, and a large number for *maximum display number* and the *maximum payment* depending on the contract between the publisher and the advertiser.

The diagram presented in Figure 3 represents the business model developed for the publisher. The rectangles in the figure represent the processes that the publisher has total control. Clouds are uncontrollable inputs for the publisher. Lastly oval shapes are combination of controllable and uncontrollable inputs, i.e., minimum payment would be settled after the negotiations with the advertiser. Moreover, the boundaries of matching and assignment problems are also demonstrated in this figure. Region covered with flat lines is representing matching problem and oval rectangle region covered with dashed lines is representing the matching problem. Note that in this paper we will focus to the matching problem and assume that the assignment problem is solved and the compatibility scores of the viewers for each advertisement is attained by utilizing one of the techniques presented in the literature such as the one proposed by Yager [9].

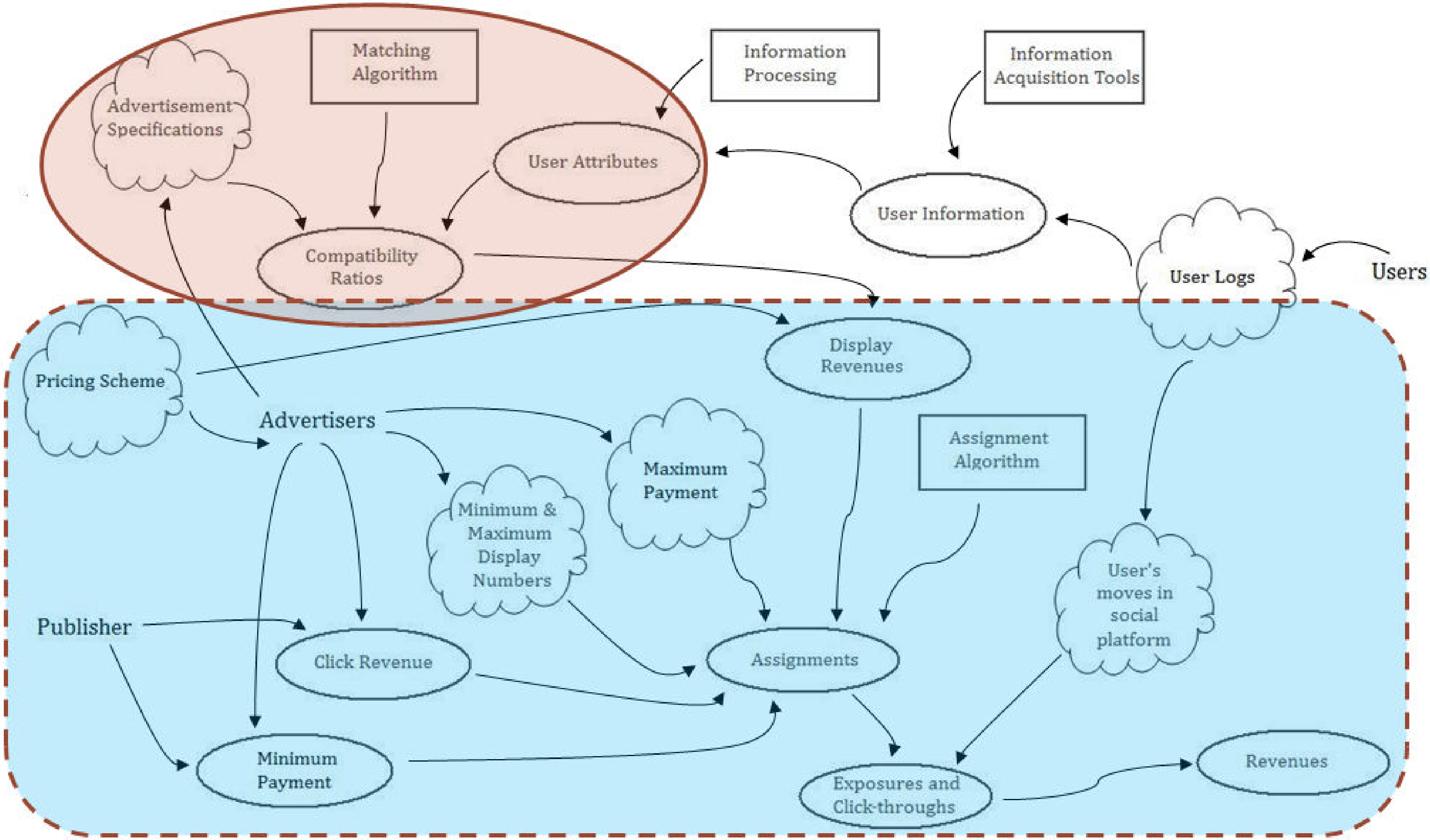


Figure 3.2 - Influence Diagram for Business Model

3.2 Mathematical Model

In this section we will present a mathematical formulation of the personalized advertisement scheduling problem. The mathematical model of the problem will clarify the details of the business model discussed earlier and enhance the understanding of the readers. The notation that will be used in the paper is as follows;

Indices:

$i = 1, \ldots, C$ *Users;*

$j = 1, \ldots, A$ *Advertisements;*

$l = 1, \ldots, L$ *Threshold Levels for price(display revenue) levels.*

Parameters:

N_i: Number of exposures for user i

C_j: Revenue obtained from one click of user $for\ advertisement\ j$.

π_{ij}: Compatibility of user i to the specifications set for the advertisement j.

T_{jl}: Threshold value for level l of advertisement j.

P_{jl}: Price for level l for one exposure of advertisement j.

$Click_i$: Click rate for user i.

$MaxPay_j$: Maximum payment for advertisement j.

$MinPay_j$: Minimum payment for advertisement j.

$MaxDisplay_j$: Maximum display number for advertisement j.

$MinDisplay_j$: The minimum display number for advertisement j.

Decision Variables:

X_{ij}: Number of payments for exposure of advertisement j to user i.

Q_{ij}: Number of exposures of advertisement j to user i.

R_{ij}: Revenue obtained from user i showing advertisement j.

B_{jl}: Binary variable that is equal to 1 if $\pi_{ij} > T_{jl}$ and 0 otherwise.

$BinX_{ij}$: Binary variable that is equal to 1 if $Q_{ij} > MinDisplay_j$ and 0 otherwise.

$BinMin_j$: Binary variable that is equal to 1 if total revenue for advertisement j is a positive amount and 0 otherwise.

$BinMax_j$: Binary variable that is equal to 1 if total revenue for advertisement j is greater than $MinPay_j$ and 0 otherwise.

$Fine_j$: Portion has to be discarded as a fine from total revenue of advertisement j.

Objective Function:

Maximize $\sum_i \sum_j (C_j Click_i Q_{ij} + R_{ij} X_{ij}) - \sum_j MinPay_j (1 - BinMin_j) + \sum_j MinPay_j (1 - BinMax_j) - \sum_j Fine_j$

subject to;

$$\sum_j Q_{ij} = N_i \qquad \forall\, i \qquad (1)$$

$$X_{ij} \leq Q_{ij} \qquad \forall (i,j) \qquad (2)$$

$$Q_{ij} \geq MinDisplay_j BinX_{ij} \qquad \forall (i,j) \qquad (3)$$

$$X_{ij} \leq MaxDisplay_j BinX_{ij} \qquad \forall (i,j) \qquad (4)$$

$$\pi_{ij} \geq T_{jl} B_{jl} \qquad \forall (i,j,l) \qquad (5)$$

$$R_{ij} = \sum_1^L B_{jl} P_{jl} - \sum_2^L B_{jl} P_{j(l+1)} \qquad \forall (i,j) \qquad (6)$$

$$\sum_i (C_j Click_i Q_{ij} + R_{ij} X_{ij}) \geq MinPay_j BinPay_j \qquad \forall j \qquad (7)$$

$$\sum_i (C_j Click_i Q_{ij} + R_{ij} X_{ij}) \leq MaxPay_j BinMax_j + Fine_j \qquad \forall j \qquad (8)$$

Objective function of the model maximizes the revenue obtained from exposures and clicks. Actually, click revenue includes uncertainty as the user behaviour regarding clicks is not obvious. Click revenue part of the objective function is the expected value of the click revenue. (1) ensures all the exposures are assigned to an advertisements. Constraints (2), (3), and (4) determines number of payments (X_{ij}) using number of exposures (Q_{ijd}). (5) and (6) calculate revenues using compatibility scores. (7) and (8) are payment constraints for advertisement-day pairs. This model is non-linear because of constraints (5) and (6); however, it can be decomposed to two models easily and resulting R_{ij}'s are imposed to the model above discarding constraints (5) and (6). Optimum result is not affected from the decomposition since R_{ij} values can be determined optimally.

3.3 Proposed Solutions

In the business model and the corresponding mathematical formulation the boundary of the problem is presented. If all of the parameters of the mathematical formulation are known in advance (deterministic case) then the integer programming (IP) model is able to find the optimum solutions. However, this problem is non-polynomial; therefore, solutions cannot be obtained in polynomial time even when all parameters are deterministic. On the contrary, the user behaviours are stochastic so that it is impossible to know exact exposure number of users (*i.e.* N_i *in mathematical model)*. Their behaviour is highly stochastic individually that it is hard to model them one by one with any well known distribution. As emphasis is on the total revenue which is the sum of each advertisement's revenue, possible approach would be to make predictions about the revenue of the advertisements using the past login data of the users.

Publisher's objective is to maximize total revenue. Therefore, developing a greedy algorithm which assigns the most profitable advertisements to users might be useful. A greedy algorithm is unable to consider *minimum display number*, *minimum* and *maximum payment* constraints since handling these constraints need intelligent methods (e.g. prediction of future). On the other hand, it is able to consider *maximum display number*. Greedy algorithm, that is developed, discards the display revenue (R_{ij}) from expected revenue variable when exposures assigned to a user for an advertisement j (i.e. Q_{ij}) hit $MinDisplay_j$. Steps of the greedy algorithm (GA) are provided below.

(Greedy Algorithm)

1) Whenever an assignment is required for the user i; calculate expected revenues for all advertisements (i.e. ER_{ij}). Expected revenue is the sum of display revenue and expected click revenue.

- If $Q_{ij} < MaxDisplay_j$ and $revenueobtained(j) < Maxpay_j$.
    - $ER_{ij} = R_{ij} + C_j\ Click_i$.
- Else if $Q_{ij} > MaxDisplay_j$ and $revenueobtained(j) < Maxpay_j$.
    - $ER_{ij} = C_j\ Click_i$
- Else
    - $ER_{ij} = 0$

2) Assign advertisement j with maximum ER_{ij} to user i.

3) Update Q_{ij}.

Greedy algorithm (GA) sounds logical. However, it does not always result with optimum solution because of the imposed constraints. Therefore, before designing a solution methodology one should understand where greedy approach fails. In the following, it will be explained how to handle all constraints.

Minimum display number encourages assigning the advertisements with smaller minimum display number for disloyal users. Disloyal users are the users that do not spend enough time to complete minimum display number of any advertisement. Consider a case where there is only one user and two advertisements. Assume number of total exposures(N_i) for the user is equal to 4. Display revenue (R_{ij}), expected click revenue (C_jClick_i) and minimum display numbers ($MinDisplay_j$) values are provided below in table 1.

Ads	R_{ij}	C_jClick_i	$MinDisplay_j$
Ad1	15	3	5
Ad2	2	4	2

Table 3.1 - Advertisement parameters for the example that GA fails for minimum display number constraint

GA assigns all exposures to advertisement 1 as ER_{i1} is greater than ER_{i2}. However, assigning all exposures to advertisement 2 is a better solution (BS) as it can be observed from table 2 below.

Algorithm	Q_{i1}	Q_{i2}	Total Revenue
GA	4	0	(0+3)*4 = 12
BS	0	4	(2+4)*4= 24

Table 3.2 - Results for the example that GA fails for minimum display number constraint

For to avoid such failures, we have developed two groups of algorithms. "*Classification*" block classifies users into two classes (i.e. loyal and disloyal) using log data of users. Log data includes number of exposures for users in past periods. "*Valuation*" block calculates ER_{ij} values for user-advertisement pairs. These two blocks promote the advertisements having smaller $MinDisplay_j$ for disloyal users. Steps of blocks are provided below.

(Classification)

1) Find maximum and minimum of minimum display number *(i.e.* $classmax$ *and* $classmin$*)*.

2) Classify users in two groups.

   - If a user i in each day she logins have exposure quantities are above $classmin$ and below $classmin$, assign $Userclass_i = 0$; otherwise assign $Userclass_i = 1$.

(Valuation)

1) Calculate ER_{ij} for every (i, j).

- If $Userclass_i = 0$, then place ad j for all past exposures and calculate total revenue for D days. Assign average of them to ER_{ij}. Otherwise; ER_{ij}= $(R_{ij} + C_j Click_i)$.

Maximum display number forces to partition the assignments of the advertisements to multiple viewers. Consider a user having *N* exposures and a case where there is no maximum display number constraint. In this case, the system will assign all exposures to the most profitable advertisement only. That is to say, all *N* exposures will be assigned to a unique advertisement. On the contrary, assume that this advertisement have a maximum display number constraint equal to *N/2;* then it will be divided to two portions between the most profitable two advertisements *(i.e. N/2: N/2)*. Maximum display number changes the way that expected revenues for advertisements are calculated.

Minimum Payment is the minimum acceptable limit of displays for the advertisers. Recall that, the advertisers prefer to set such a limit to ensure that the advertisement at least reaches to an adequate number of users. If the revenue is less than this value the publisher will earn no money. Therefore, if the probability of exceeding this level is low, the publisher should consider not publishing this advertisement any more since these exposures will not yield any revenue for the company. Consider a case with two advertisements and two users. Assume $MinPay_1 = 20$ and $MinPay_2 = 60$. ER_{ij} and N_i values are provided below in table 3.

Users	ER_{i1}	ER_{i2}	N_i
User1	15	2	3
User2	8	10	5

Table 3.3- Problem parameters for the example that GA fails for minimum payment constraint

In this example; it is obvious that advertisement 2 will not exceed *minimum payment constraint*. However, GA is not able to predict it, therefore GA assign exposures of user 1 to advertisement 1 and exposures of user 2 to advertisement 2. GA obtained total revenue of 45. However, it is better to assign all exposures to advertisement 1. Solutions of GA and BS are provided below in tables 4 and 5.

Exposures(GA)	Ad1	Ad2
User1	3	0
User2	0	5
Revenue	45 (3*15=45 > 20)	0 (5*10=50 < 55)

Table 3.4 - Results of GA for the example that GA fails for minimum payment constraint

Exposures(GA)	Ad1	Ad2
User1	3	0
User2	0	5
Revenue	45 (3*15=45 > 20)	0 (5*10=50 < 55)

Table 3.5 - Results of BS for the example that GA fails for minimum payment constraint

We have developed two groups of algorithms in order to handle minimum payment constraint. Consider, we have a set A that includes all advertisements. "β-Discarding" block discards advertisements from set A that are not likely to exceed minimum payment constraint. "β" block both discards and adds advertisements to the set A. Inside of the algorithm; there is a parameter called β. If $\beta = 0$, it causes all advertisements to remain in set A. On the other hand; if $\beta = 1$, then all advertisement will be discarded from set A. Steps of blocks are provided below.

(β-Discarding)

1) Rank advertisements according to $ER_{ij}coef_{ij}$. *(i.e.* $rankad_{ij}$*)*

2) Place advertisements to the past exposures using their ranks. Record assignments *(i.e.* ε_{ijd}*)*

3) Set $j = 0$.

4) Set $j = j + 1$. If j is larger than the number of advertisements, set $j = 0$; END.

5) Check if $revenueobtained(j) \geq MinPay_j$. If equation is satisfied go to step 4, continue otherwise.

   a. Calculate the amount needed for exceeding minimum payment *(i.e.* $slackmin(j)$*)*
      - $slackmin(j) = minbudget(j) - revenueobtained(j)$

   b. For every day d; calculate expected total revenue for advertisement j *(i.e.* $expectationsum_{jd}$*),* then calculate averages and standard deviation of $expectationsum_{id}$ with the following equations.
      - $expectationsum_{jd} = \sum_i \varepsilon_{ijd} ER_{ij}$
      - $Avg(j) = \sum_d expectationsum_{jd}$
      - $Std(j) = StandardDeviation\left(expectationsum_{j1}, \dots, expectationsum_{jD}\right)$

   c. Check if $\Phi\left(\frac{Avg(j) - slackmin(j)}{Std(j)/\sqrt{D-1}}\right) < \beta$. If the equation above is satisfied, then do the following.
      - $safe(j) = 0$
      - $coef_{ij} = 0 \ \forall\ (i,j)$

6) Go to the step 4.

(β)

1) Carry out β-Discarding.

2) Set $action = 0$ and $j = 0$.

3) Set $j = j + 1$. If j is larger than the number of advertisements, set $j = 0$; go to the step 6.

4) Check if $safe(j) = 0$.

a. Place ad j in past exposures with maintaining safety of safe advertisements.
b. Obtain expected total revenue for every past period d, average *(i.e.* $CompetitiveAvg(j)$*)* and standard deviation *(i.e.* $CompetitiveStd(j)$*)*
c. Add $revenueobtained(j)$ to find $competitivescore(j)$.
   - $competitivescore(j) = CompetitiveAvg(j) + revenueobtained(j)$

5) Go to step 3.

6) Find the best unsafe advertisement j with the maximum competitive score *(i.e.* $competitivescore(j)$*)*.

a. Check if $\Phi\left(\frac{CompetitiveAvg(j) - slackmin(j)}{CompetitiveStd(j)/\sqrt{D-1}}\right) > \beta$. If the equation above is satisfied, then do the following and go to step 16; otherwise go to the next step.
   - $safe(j) = 1$
   - $coef_{ij} = 1 \ \forall\ (i, j)$
   - $action = 1 \ \forall\ (i, j)$

b. Set $competitivescore(j) = 0$. If all ads are checked go to the step 7; otherwise, go to the step 6.

7) If $action = 1$; go to the step 6, otherwise END.

Maximum payment is the maximum revenue publisher can obtain from an advertisement. If potential is much more than maximum payment, it is required to fill this budget with the best set of users in terms of the *relative* revenue (i.e., by taking the other advertisers' payments and set of potential users into account as well). One should keep in mind that, even though a particular advertisement is the one that pays most for a particular user, if the potential set of users which have high compatibility with that advertisement is very large, then it might be more profitable to assign *another* advertisement which has a narrow set of potential users rather than it. Consider a case with two advertisements and two users. Assume $MaxPay_1 = 60$ and $MaxPay_2 = 60$. $ER_{ij}$and N_i values are provided below in table 6. User1 enters the system first. Table 6 illustrates relative revenue values of users for advertisement 1. As $relativerevenue_{21} > relativerevenue_{11}$, user 2 is more valuable for advertisement 1.

Users	ER_{i1}	ER_{i2}	N_i	$relativerevenue_{i1}$
User1	15	9	4	15 – 9 = 6
User2	12	2	5	12 – 2 =10

Table 3.6 - Problem parameters for the example that GA fails for maximum payment constraint

When user 1 enters into the system, GA will assign all exposures to advertisement 1 and will obtain a revenue of 60. As GA hits $MaxPay_1$, GA will assign all exposures of user 2 to advertisement 2. However, a better solution (BS) can be obtained. Table 7 and 8 presents the solutions of GA and BS respectively. As it is obvious here that advertisement 1 hits $MaxPay_1$, it is better to assign advertisements to users with higher relative revenue.

Exposures(GA)	Ad1	Ad2
User1	4	0
User2	0	5
Revenue	60	10

Table 3.7- Results of GA for the example that GA fails for maximum payment constraint

Exposures(BS)	Ad1	Ad2
User1	0	4
User2	5	0
Revenue	60	36

Table 3.8 - Results of BS for the example that GA fails for maximum payment constraint

In order to handle *maximum payment* constraint, we have developed a block called α. α block determines a set of users for each advertisement using the relative revenues. If $\alpha = 0$, all users will be discarded from the advertisements; else if $\alpha = 1$, all users will remain in the their set. Details are provided below.

(α)

1) Rank advertisements according to $ER_{ij}coef_{ij}$.

2) Place advertisements to the past exposures using their ranks. *(i.e.* $rankad_{ij}$*)* Records assignments *(i.e.* ε_{ijd}*)*.

3) Calculate *relative revenues* for advertisement-user pairs *(i.e.* $relativerevenue_{ij}$*)*. Relative revenues are calculated subtracting the ER_{ij} value of the succeeding advertisement in terms of rank from ER_{ij} value of its own.

   a. $relativerevenue_{ij} = ER_{ij} - ER_{ij_c}\ where\ rankad_{ij} = n\ \&\ rankad_{ij_c} = n + 1$

4) Rank users in terms of relative revenues for each advertisement. *(i.e.* $rankuser_{ij}$*)*

5) Set $j = 0$.

6) Set $j = j + 1$. If j is larger than the number of advertisements, set $h = h + 1$; then END.

7) Check if $revenueobtained(j) \geq maxpay(j)$. If yes, go to step 6, otherwise continue.

   a. Calculate the amount required to exceed $maxpay(j)$ *(i.e.* $slackmax(j)$*)*
      - $slackmax(j) = maxpay(j) - revenueobtained(j)$.
   b. Start with an empty set of users, U. Set $n = 1$;
   c. Select i that $rankuser_{ij} = n$. Add ad i to the set U.
   d. For every past period d; calculate expected total revenue with the set of U for advertisement j *(i.e.* $expectationsum_{jd}$*)*, then calculate averages and standard deviation of $expectationsum_{id}$.
      - $expectationsum_{jd} = \sum_{i \in U} \varepsilon_{ijd}\, ER_{ij}$
      - $Avg(j) = \sum_{d} expectationsum_{jd}$
      - $Std(j) = StandardDeviation(expectationsum_{j1}, \dots, expectationsum_{jD})$

   e. Check if $z\left(\frac{Avg(j) - slackmax(j)}{Std(j)/\sqrt{D-1}}\right) > \propto$. If the equation is satisfied go to the step g, otherwise continue.
   f. If n is equal to number of users, go to the step 6. Otherwise, set $n = n + 1$ and go to the step 7c.

   g. Change the coefficient of users to a small number S that are not inside of set U. Go to step 6.
      - $coef_{ij} = S\ for\ i \in \overline{U}$

Considering all details we have designed an algorithm called *"Personalized Advertisement Assignment System"* (PAAS). Blocks developed are used in PAAS. Details of the algorithm are provided below.

3.3.1 PAAS Algorithm

PAAS algorithm is an algorithm that evaluates "state" and manipulates assignment decisions accordingly. $State(t)$ is simply all the relevant information about the problem at time t and it is changing in each second. It is impossible predict changes in system since user's behavior are stochastic. Therefore, the changes in the system should be monitored periodically and action has to be taken accordingly. Below the control loop is showing dynamic nature of PAAS algorithm.

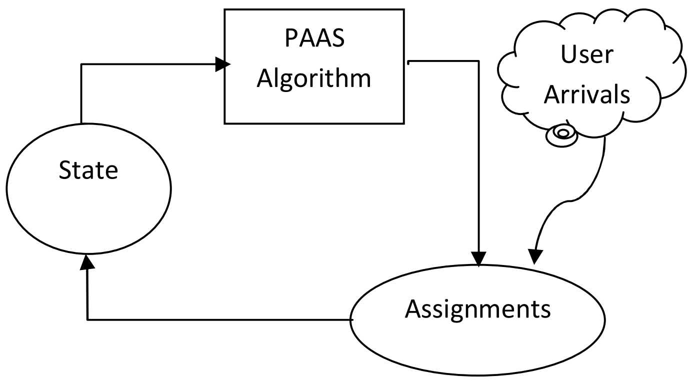


Figure 3.3 - Control Loop of PAAS algorithm

Consider our period is a day (i.e. maximum and minimum payments are determined daily). Then, the loop above is traversed T $(t = 0,1,2, \ldots, T - 1)$ times for a single period. Algorithm starts with $State(0)$ and advertisement-user pair coefficients $coef_{ij}$ $\forall$ (i, j) are calculated in PAAS algorithm. Assignments are determined using these coefficients whenever a user arrives into the system. According to the assignments $State(1)$ is obtained. New coefficients are calculated for $State(1)$. Increasing T makes our system more dynamic; however, more data are required to be stored. After the selection of T; the most representative past data have to be chosen for forecasting the future exposures of users. Number of past periods taken for to predict future exposures is denoted with D $(d = 1,2,\dots,D)$. Past data include how many exposures were realized for each individual user in each subperiod t of day d (*i.e.* $Exposures_{idt}$) and click information for every exposure. Moreover, information about click rate is calculated from past data. Click-rate is basically number of clicks is divided by total number of exposures (*i.e.* $Click_i$). PAAS starts with a block called "Preliminary". Preliminary block includes starting steps for PAAS. It is followed by a heuristic which is composed of blocks explained in section 3.3. Finally it ends with assignment block. Steps of preliminary and assignment blocks are provided below.

(Preliminary)

1) All parameters are stored in database.
2) Calculate display revenues (R_{ij}) from compatibility ratios.
3) Assign 1 for all $coef_{ij}$. PAAS algorithm manipulates assignment decisions by changing this variable. This coefficient is multiplied with a function in terms of display revenue and click revenue. As this is the starting point all coefficients have to be equal to 1.
   a. $coef_{ij} = 1 \ \forall \ (i,j)$
   b. $t = 0$

(Assignment)

1) When a user enters into the system; $index_{ij}$'s are calculated for all advertisements and highest of them are assigned to the user.

   - $index_{ij} = ER_{ij}coef_{ij}$
   - $bestad(i) = mdp(j) = argmax_{j \epsilon A}\left(index_{ij}\right)$
   - $published_{ij} = published_{ij} + 1$

2) Revenues are calculated and added to related advertisements. If obtained revenue for advertisement j is greater than maximum payment set all coefficients to 0 for advertisement j.
   - $coef_{ij} = 0$

3) If $published_{ij}$is equal to the maximum display number, update $indrevenue_{ij}$ and $coef_{ij}$.

- $ER_{ij} = C_j\ Click_i$
- $coef_{ij} = S$

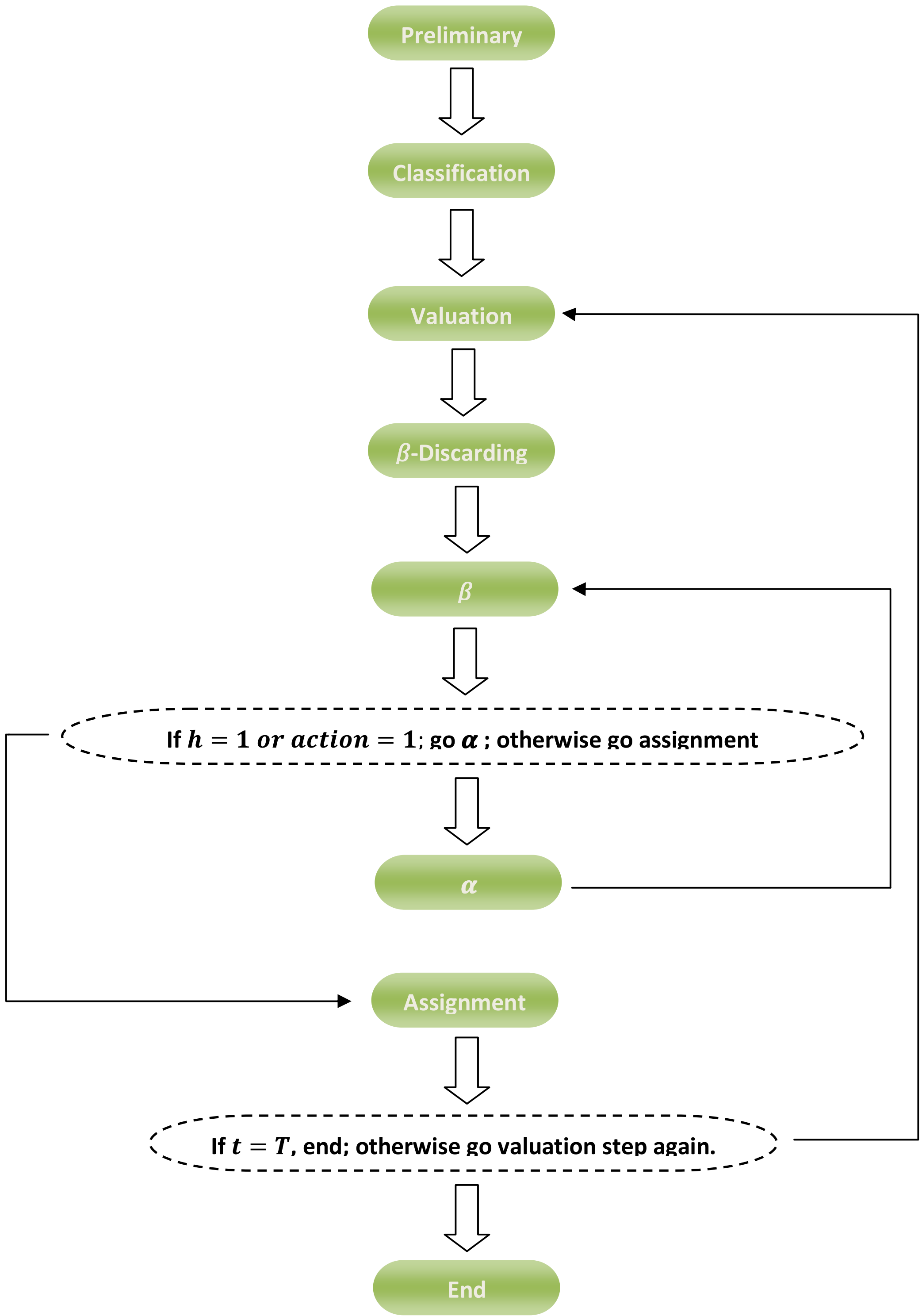


Figure 3.4 - Flow chart of PAAS algorithm

PAAS algorithm is constructed with a combination and relationship of blocks presented before; preliminary, classification, valuation, β-Discarding, β ,α and assignment. Flow chart in figure 3.4 illustrates how PAAS algorithm works.

Chapter Four

EXPERIMENTAL DESIGN AND ANALYSIS

In order to test the performance of the proposed algorithm experimental analysis is conducted. Alternative to the PAAS algorithm, three more algorithms are also proposed, namely the PAAS Without Classification (PAASWOC), the Greedy Algorithm (GA) and the Greedy Algorithm With Classification (GAWC). Four algorithms' performances are compared under various experimental conditions. Different testing conditions are constructed with combinations of different levels of maximum payment, minimum payment, maximum display number, minimum display, number of users. Constraints are defined in two groups as tight and loose e.g. if maximum payment of an advertisement is too high that there is a low probability to reach then it is a "loose" constraint; if there is a high probability to reach then it is a "tight" constraint. Below table 1 is illustrating alternatives for these parameters. As it is shown in the table the algorithms are tested under 36 distinct cases.

Parameters	*Options*	
A) Minimum Payment	1) All ads are tight 2) Half of the ads are tight 3) All ads are loose	3
B) Maximum Payment	1) All ads are tight 2) Half of the ads are tight 3) All ads are loose	3
C) Minimum Display Number	1) All ads are tight 2) All ads are loose	2
D) Maximum Display Number	1) All ads are tight 2) All ads are loose	2
Total number of combinations		36

Table 4.1 - Problem parameters and options

We have developed a representation system for cases instead of writing in a long way. Cases are coded like (A, B, C, D) where A refers to minimum payment constraints, B refers to maximum payment constraints, C refers to *minimum* display number constraints and D refers to *maximum* display number constraints. For example; $(1, 2, 2, 1)$ denotes the case which all *minimum* payments are tight, half of the *maximum* payments are loose, all *minimum* display number constraints are loose and all *maximum* display number constraints are tight.

Some of the parameters are generated using the data provided by the company, some are generated randomly using global standards and rest is generated randomly. The data provided by the company included only a portion of the users' activity, namely, the login and the logout times of the users. Unfortunately, further information (e.g. number of exposures and number of clicks) was not available so they were generated using these data assuming there is one exposure for every 120 seconds. We created 12 advertisements for the analysis. The compatibility scores of the users were randomly generated for each one of the advertisement. We assumed three levels for the price of the exposures for the advertisers. The threshold values and corresponding prices are selected randomly.

The virtual environment provider company provided *14 days* of data and exposures numbers are calculated for every subperiod accordingly. The first *7 days* of the historical data was utilized in order to estimate the users' activities parameters (i.e., the training set) and the second *7 days* was utilized to compare the performances of the algorithms (i.e., the test set). For each of the 36 cases we have carried out 10 replications. Furthermore, three different levels of users, namely, 100, 250 and 500 is considered in the analysis. Therefore altogether, 36*3*10 = 1080 experiments are conducted.

We assumed that the contracts between the publisher and the advertiser were daily, i.e., the *minimum* and *maximum* payment constraints as well as the *minimum* and *maximum* display number constraints were defined daily. *Minimum* and *maximum* payment constraints as well as *minimum* and *maximum* display number constraints are generated independently for every replication. According to Novak and Hoffman, [18] less than three exposures are ineffective for user to understand the message given. Therefore, we decided that tight minimum display constraint is distributed randomly between 3 and 5 and loose minimum display number constraint is distributed between 1 and 3. Moreover, Novak and Hoffman [16] state that exposures after 10 exposures have a small incremental effect. Therefore, we decided that tight maximum display constraint is distributed randomly between 6 and 10 and loose minimum display number constraint is distributed between 10 and 14. On the other hand, *minimum* and *maximum* payment constraints are generated via the procedure presented below.

- Determine advertisements with tight and loose minimum payments
- Determine advertisements with tight and loose maximum payments
- Find expected revenues for each advertisement-day pair.
- Calculate average and standard deviation from expected revenues.
- Generate a random number for both minimum and maximum payment *(i.e.* $randmin_j$, $randmax_j$*)*from the following formulas. If payments are tight it is represented by 1, otherwise represented by 0.

(Minimum Payment, Maximum Payment)	$randmin_j$	$randmax_j$
(0,0)	$0.01 + 0.01\ \phi(rand)$	$0.98 + 0.02\ \phi(rand)$
(0,1)	$0.01 + 0.01\ \phi(rand)$	$0.05 + 0.05\ \phi(rand)$
(1,0)	$0.2 + 0.1\ \phi(rand)$	$0.98 + 0.02\ \phi(rand)$
(1,1)	$0.2 + 0.1\ \phi(rand)$	$0.2 + 0.1\ \phi(rand)$

Table 4.2 - Random number generation formulas for payment constraints

- Calculate minimum and maximum payments from the formulas in the table.

(Minimum Payment, Maximum Payment)	$MinBud_j$	$MaxBud_j$
(0,0) , (0,1) , (1,0)	$average_j + std_j \phi(randmin_j)$	$average_j + std_j \phi(randmax_j)$
(1,1)	$average_j + std_j \phi(randmin_j)$	$MinBud_j(1 + randmax_j)$

Table 4.3 - Payment generation formulas

PAAS is the basic algorithm that we have proposed. Other three algorithms are obtained discarding one or a set of blocks; namely classification, α *and* β. If classification of users is carried out; the algorithm is called PAAS. Otherwise, the algorithm is called PAASWOC. PAAS and PAASWOC are run for combinations of 6 different levels of $\boldsymbol{\alpha}$ and 6 different levels of β. *(i.e.* $\alpha = \{0.7, 0.8, 0.85, 0.9, 1\}$ *and* $\beta = \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$. *)* Therefore, 72 different solution methods are tested for each replication. GAWC is equal to PAAS with $\alpha = 1$ and $\beta = 0$ and GA is equal to PAASWOC with $\alpha = 1$ and $\beta = 0$.

Firstly, we have prepared plots which illustrate the mean performances of algorithms under 36 cases. Plots are drawn for three levels of users. Interested readers can view appendix A. Below in figure 6, plot for 500 users, mean performances of algorithms are illustrated. GA and GAWC, and best results *(i.e best* (α, β) *pairs)* of PAAS and

PAASWOC for each case are placed in plot. Horizontal axis presents cases and vertical axis presents mean performances.

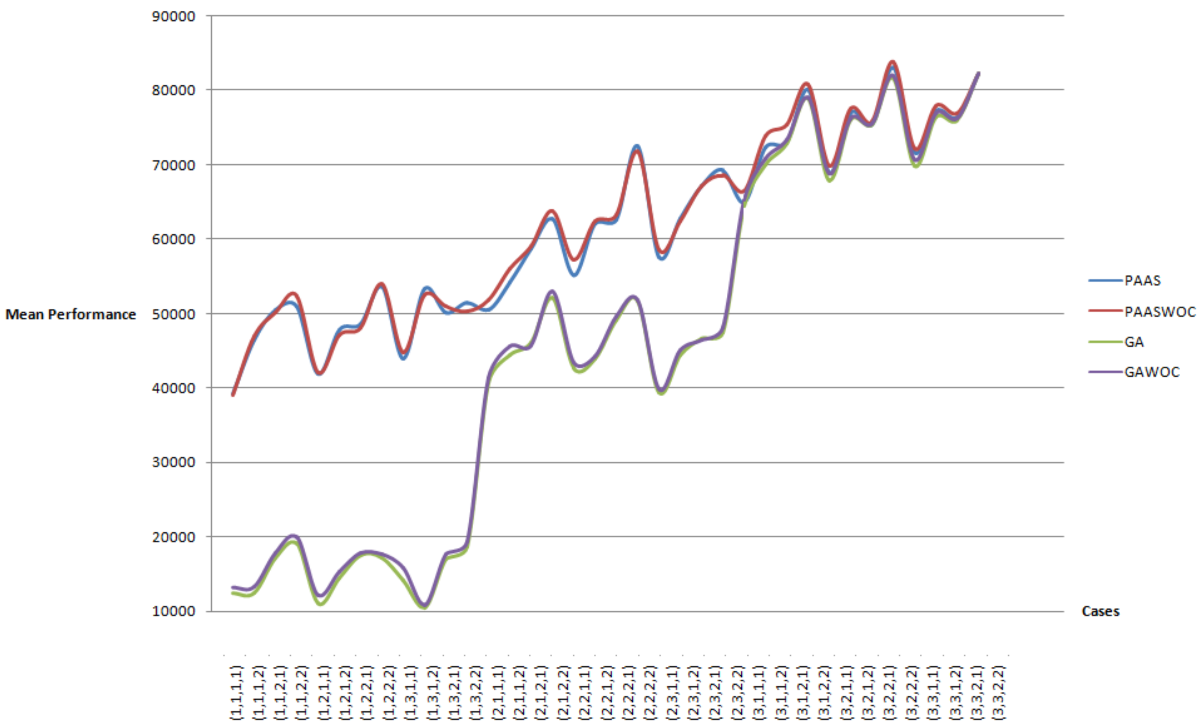


Figure 4.1 - Mean performance vs. cases for algorithms ($\mathbf{Number\ of\ Users = 500}$)

According to the graph, we have observed that PAAS - PAASWOC and GA – GAWC pairs are very close in terms of mean performance. Moreover, PAAS and PAASWOC outperform GA and GAWC clearly when minimum payments are tight. *(i.e when* $A = 1$ *or* $A = 2$*)*. However; differences are decreasing from $A = 1$ *to* $A = 3$ and A seems more dominant determining differences than other factors. Therefore, it is adequate to simplify analysis by decreasing case number from 36 to 3 by eliminating other factors, B, C *and* D.

For finding the best (α,β) *pairs* for PAAS and PAASWOC in three regions; means of each pair are calculated and best pairs are compared with second, third, fourth, fifth and sixth best pairs via pairwise t tests. For three levels of number of users, tables below summarizes which (α,β) *pairs* which are not significantly worse than the best pair for type

I error of 0.1. Table 4 and 5 are for PAAS and PAASWOC respectively. Pairs are written in order of the mean performances.

	$A = 1$	$A = 2$	$A = 3$
100	(1,0.9)	(0.95,0.9)	(1,0),(1,0.9),(1,0.7), (1,0.5)
250	(1,0.9)	(1,0.9), (0.95,0.9)	(1,0.9)
500	(0.95,0.9), (1,0.9)	(1,0.9), (0.95,0.9), (0.9,0.9), (0.85,0.9), (0.8,0.9)	(0.9,0.9), (0.85,0.9), (0.95,0.9), (0.8,0.9), (1,0.9)

Table 4.4 - Best pairs of PAAS under different number of users and minimum payment constraints

	A = 1	A = 2	A = 3
100	(1,0.9), (1,0.7)	(0.95,0.9), (1,0.9), (0.9,0.9)	(1,0), (1,0.5), (1,0.7), (1,0.1), (1,0.9)
250	(1,0.9)	(0.95,0.9), (0.9,0.9), (0.85,0.9), (1,0.9)	(1,0.7), (1,0.5), (1,0.9), (1,0.3), (1,0.1), (1,0)
500	(0.95,0.9), (0.85,0.9), (0.8,0.9)	(1,0.9), (0.95,0.9)	(0.95,0.9), (0.9,0.9), (0.8,0.9), (1,0.3), (1,0.1)

Table 4.5 - Best pairs of PAASWOC under different number of users and minimum payment constraints

*Bes*t pairs for different levels of minimum payment are determined as final parameters of PAAS and PAASWOC. For example, for 250 users and $A = 2$; best parameters of PAAS is (1,0.9) whereas PAASWOC performs best with (0.95,0.9). Using these parameters, we calculate the averages of 10 replications for 36 cases. Below table 6 illustrates average mean performances of algorithms under different number of user levels. These average performances are obtained taking the averages of 36 cases. According to the results; PAAS performs better than PAASWOC for 100 users. On the contrary, PAASWOC is better for 250 and 500 users. GAWC performs better than GA for 100 users; however, GA is better for 250 and 500 users. Performance of greedy algorithms are worse than PAAS and PAASWOC for all number of user levels.

	100 Users	250 Users	500 Users
PAASWOC	12549.42	31258.65	61734.6
PAAS	12642.39	30777.85	61235.48
GA	11208.23	26744.44	45767.75
GAWC	11527.54	26315.91	45194.95

Table 4.6 – Average mean performances of algorithms vs. number of users

Using averages of 36 cases, pairwise t tests are applied. According to the results; all the differences are statistically significant for all pairs except (PAAS-PAASWOC) for 100 users. Therefore, we cannot say anything about PAAS or PAASWOC performs better for 100 users. However, we can state that PAASWOC performs better than PAAS in general. For 250 and 500 users, GA performs better; however for 100 users GAWC performs better.

One more thing, we tried to explore is which α and β values are more common in best results. Below histograms are provided for 100 users separately for PAAS and PAASWOC. From figures 7 and 9, it is observed that most of the best results are obtained when $\beta = 0.9$. On the other hand, we have seen that distribution is more equal for α. However, again most of the solutions are obtained when $\alpha = 1$.

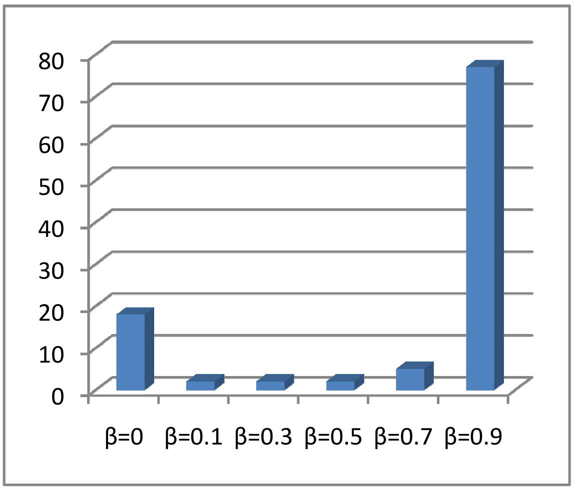


Figure 4.2 - Quantity of best results for different values of β (PAAS)

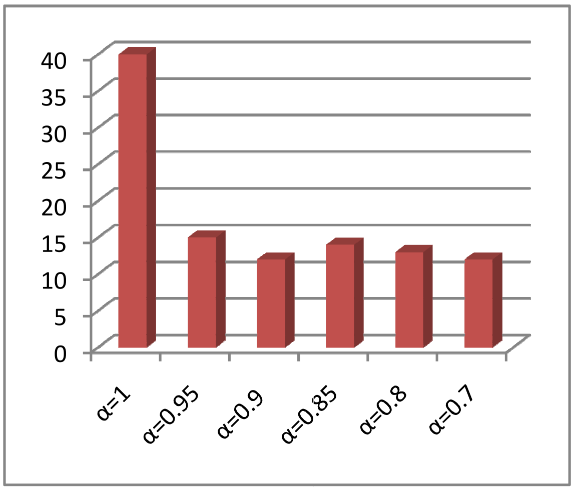


Figure 4.3 - Quantity of best results for different values of α (PAAS)

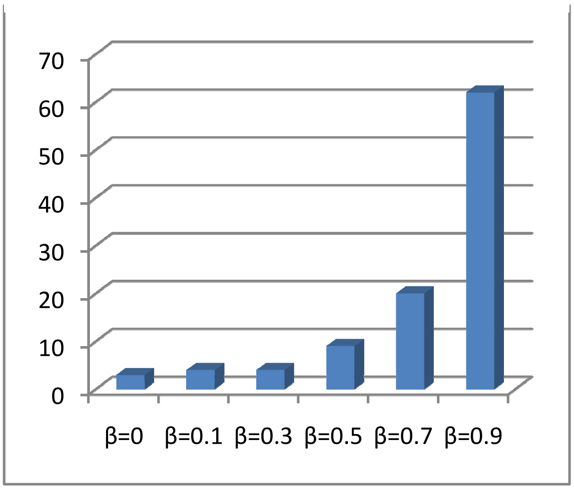


Figure 4.4 - Quantity of best results for different values of β (PAASWOC)

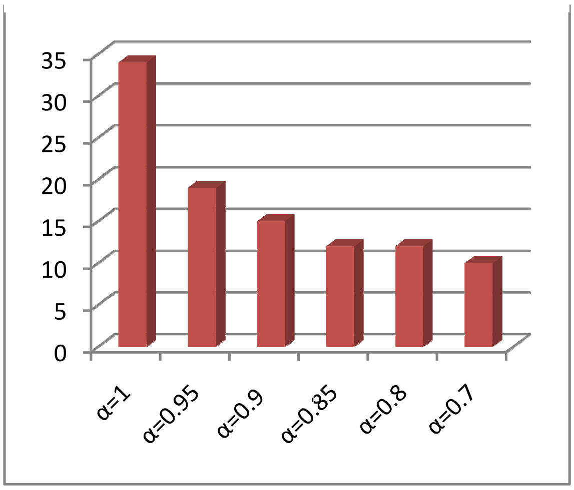


Figure 4.5 - Quantity of best results for different values of α (PAASWOC)

Moreover, we had prepared plots to understand the role of α and β under different combinations of minimum and maximum payment constraints. All these figures are provided in appendix B. Below one example figure for $A = 2$ and $B = 1$ is provided. In this figure, again we observe the dominance of β. β affects the result primarily; however in this figure we observe interaction effect of α and β since the lines for different values of β behaves differently for changing values of α. Another implication is that the effect of α is almost disappeared when $B = 3$. Power of α increases when A goes through 1 and power of β increases when B goes through 1.

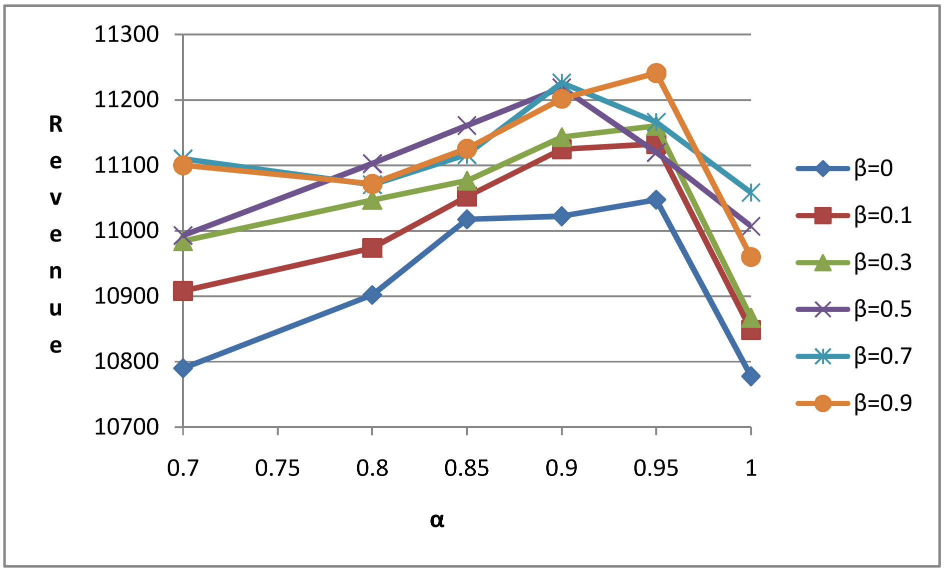


Figure 4.6 - Revenue vs. (α, β) when $A = 2$, $B = 1$ and Number of users $= 100$

Finally, we have questioned the effect of parameter T, *i.e. number of subperiods*. Same problems are solved for three different levels of T for 100 users; namely 4, 12, 24. Low T means less number of interventions to the system; and high T means opposite. If algorithms performs better under higher T; that means interventions improves the solution quality. Best averages of 36 cases for all 4 algorithms are calculated for three levels of T. Then $T = 4, T = 12$ and $T = 24$ are tested pairwise with $36 * 4 = 144$ cases. According to the results, differences between all pairs are significant and $T = 12$ is better than $T = 24$ and $T = 24$ is better than $T = 4$.

Pairs	Mean Difference	Significance
$(T = 4) - (T = 12)$	-102.8727	0.040
$(T = 4) - (T = 24)$	-92.9024	0.060
$(T = 12) - (T = 24)$	9.9703	0.017

Table 4.7 - Pairwise T test for different values of $T = \{4, 12, 24\}$

Chapter Five

CONCLUSIONS AND FURTHER RESEARCH

In this study a real problem is presented that a 3D social platform provider faced. Firstly, problem is defined and adequate problem solutions are researched from the literature. Considering the requirements of the advertisers and the publisher company; a new business model is developed. The business model and two distinct problems are defined within this business model; namely the advertisement matching problem and the personalized advertisement assignment problem. We have focused in assignment problem since many different approaches are provided for matching problem in literature.

An algorithm called "Personalized Advertisement Assignment System" (PAAS) is proposed as a solution method for personalized advertisement assignment problem. Four different variations of PAAS are tested under various conditions; namely PAAS, PAASWOC, GAWC and GC. According to the pairwise t-test results, PAAS and PAASWOC are significantly better than GAWC and GA where p values are less than 0.001. PAASWOC is significantly better than PAAS for numbers of users 250 and 500 ($p < 0.001$) whereas PAAS is better in terms of mean performance for 100 users. However, the difference is not statistically significant. Moreover, GA is significantly better than GAWC for number of users 250 and 500; however GAWC is significantly better for 100 users ($p < 0.001$). Furthermore, it is illustrated visually by a graph that the mean difference between intelligent algorithms (i.e PAAS and PAASWOC) and greedy algorithms (i.e. GAWC and GA) is quite different for different number of tight minimum payments.

PAAS and PAASWOC's performances are measured for different values of parameters; namely α, β and T. Best α, β values are discovered for both PAAS and PAASWOC under different number of tight minimum payments. Moreover, interaction between α and β is illustrated visually. Best number of subperiods, T is determined as 12 when number of users is equal to 100.

The most challenging part of our study is the stochastic user behaviour. A statistics based solution has been suggested for estimating user behaviours. Note that, the success of algorithm can be improved with better estimating methods. Click revenue estimation methods can be improved, Kumar et al.'s approach in estimating click rates can be integrated in our PAAS algorithm [15]. PAAS and PAASWOC algorithms are flexible that; they can be modified for another problem. Note that, algorithms provided are designed for a virtual environment; however, it is available for any social web site.

This study introduced a new business model and a new advertising problem to the literature. Furthermore, a pool of various problems is generated for further studies and solution algorithms are proposed. Performances of algorithms are measured using these problems.

APPENDIX A

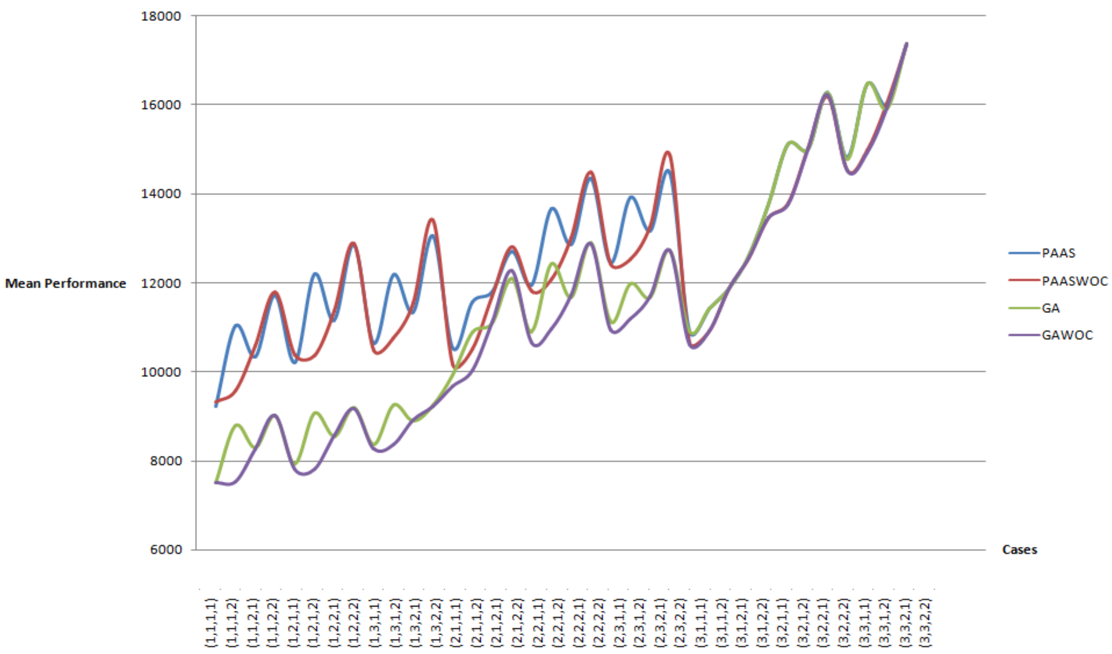


Figure A-1 Mean performance vs. cases for algorithms (Number of Users = 100)

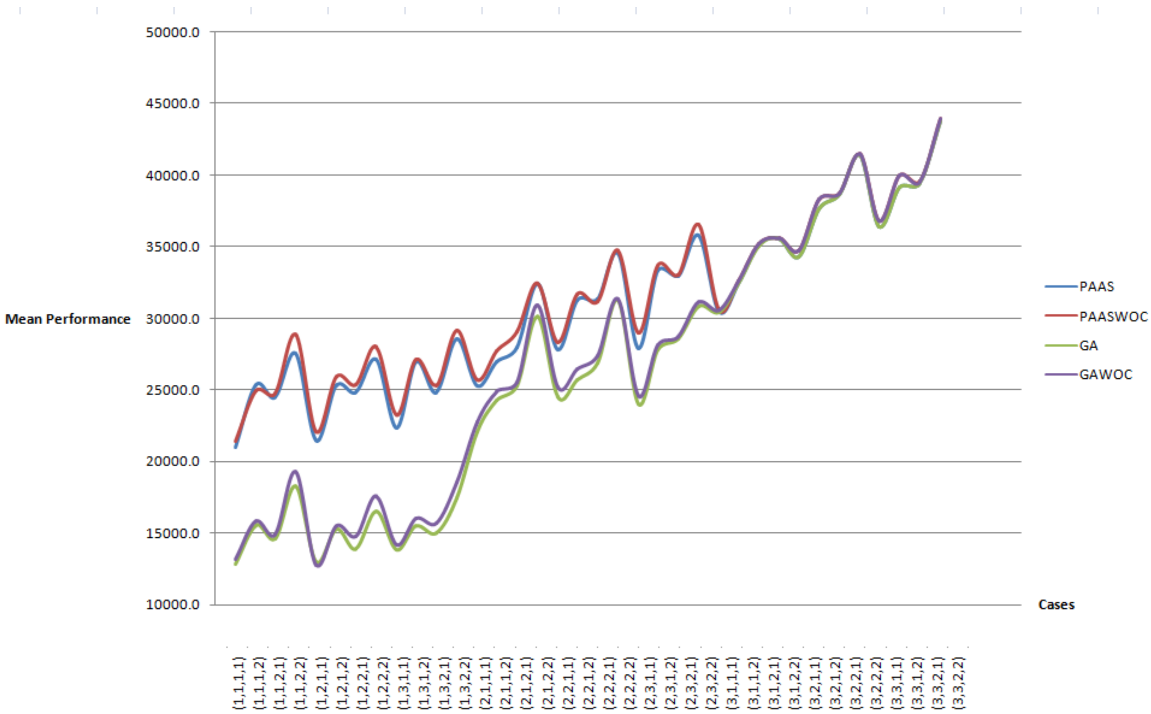


Figure A-2 Mean performance vs. cases for algorithms (Number of Users=250)

APPENDIX B

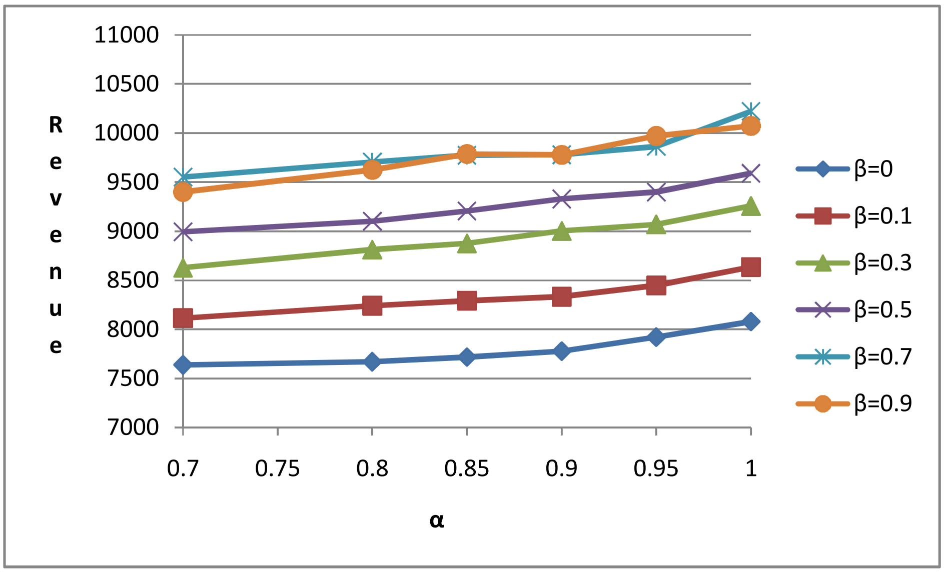


Figure B-1 Revenue vs. (α, β) when A = 1 , B = 1 and Number of users = 100

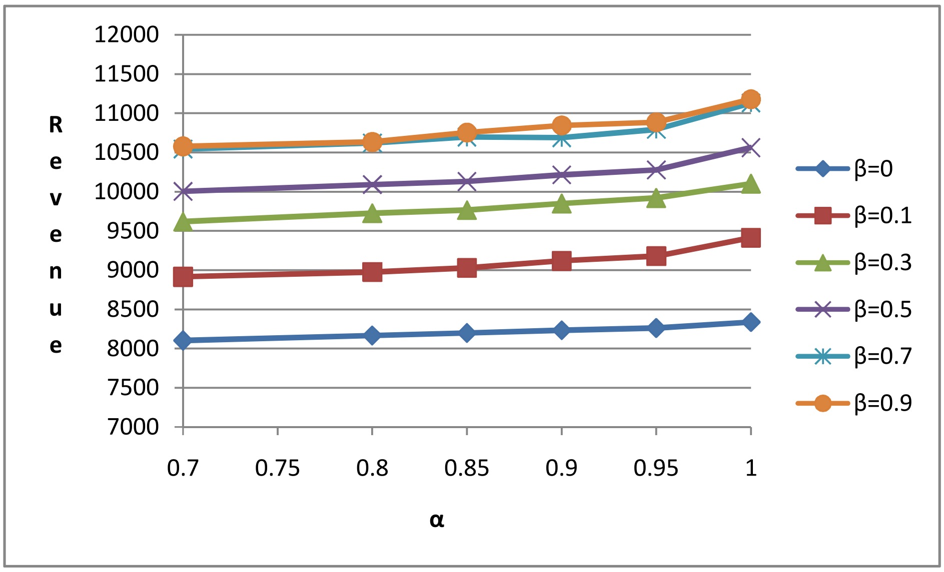


Figure B-2 Revenue vs. (α, β) when A = 1 , B = 2 and Number of users = 100

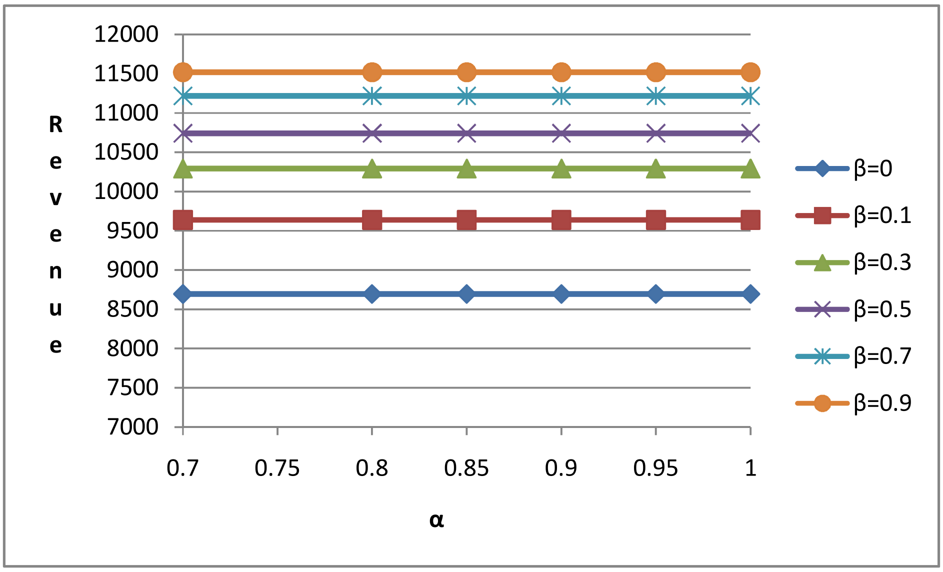


Figure B-4 Revenue vs. (α, β) when A = 1 , B = 3 and Number of users = 100

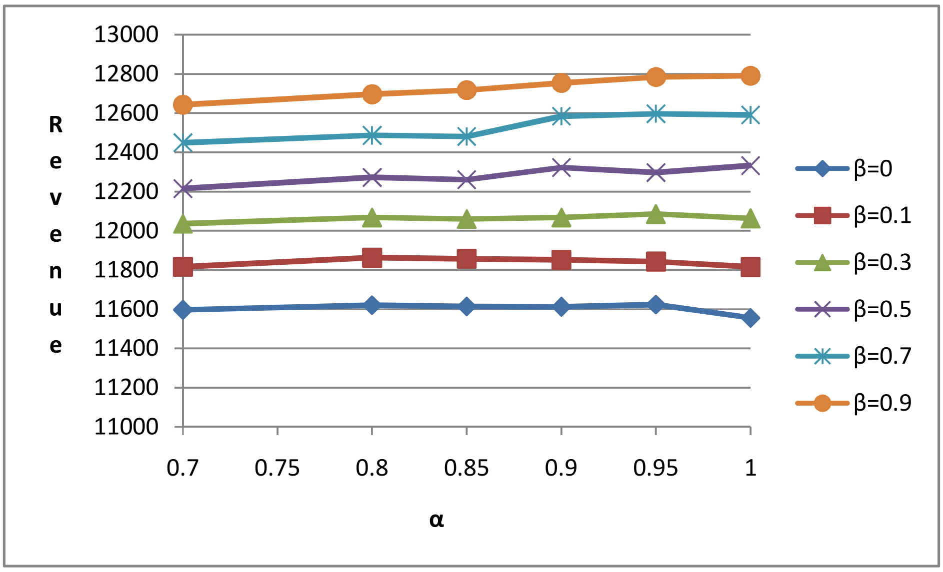


Figure B-3 Revenue vs. (α, β) when A = 2 , B = 2 and Number of users = 100

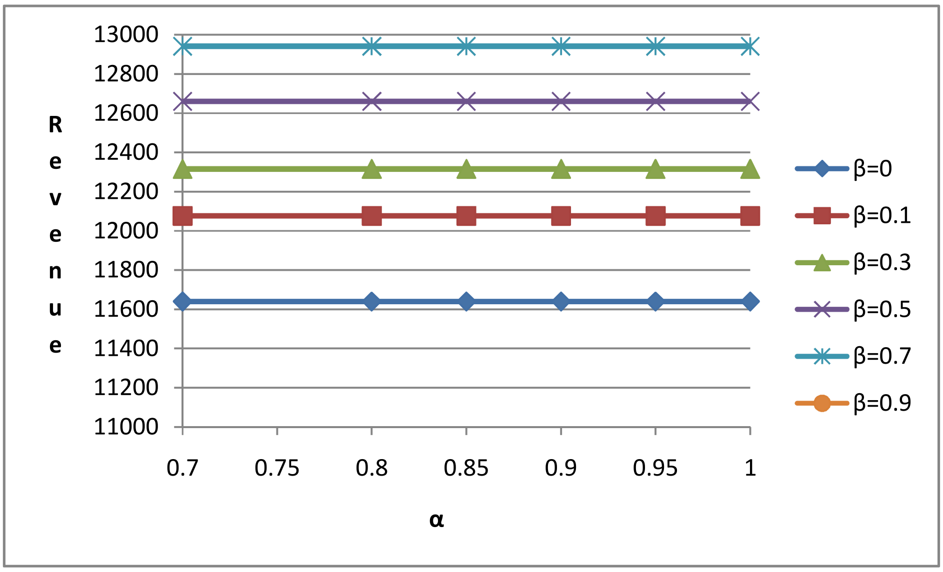


Figure B-5 Revenue vs. (α, β) when $\mathrm{A} = 2$, $\mathrm{B} = 3$ and Number of users = 100

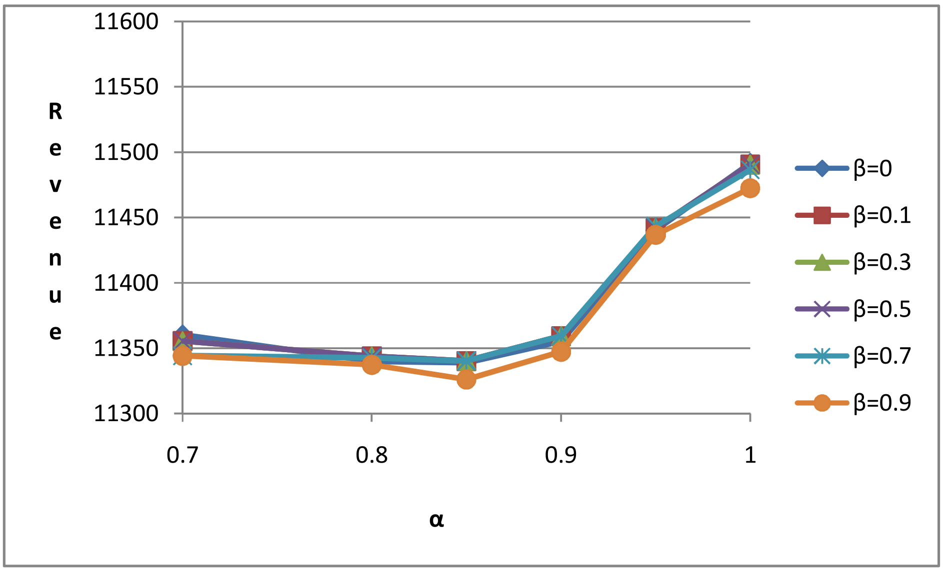


Figure B-6 Revenue vs. (α, β) when $\mathrm{A} = 3$, $\mathrm{B} = 1$ and Number of users = 100

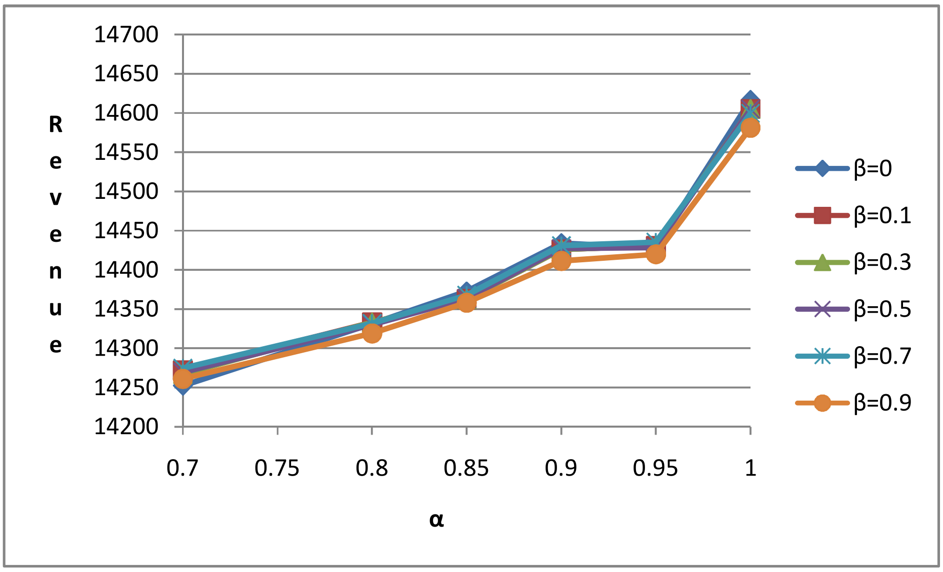


Figure B-7 Revenue vs. (α, β) when $\mathrm{A} = 3$, $\mathrm{B} = 2$ and Number of users $= 100$

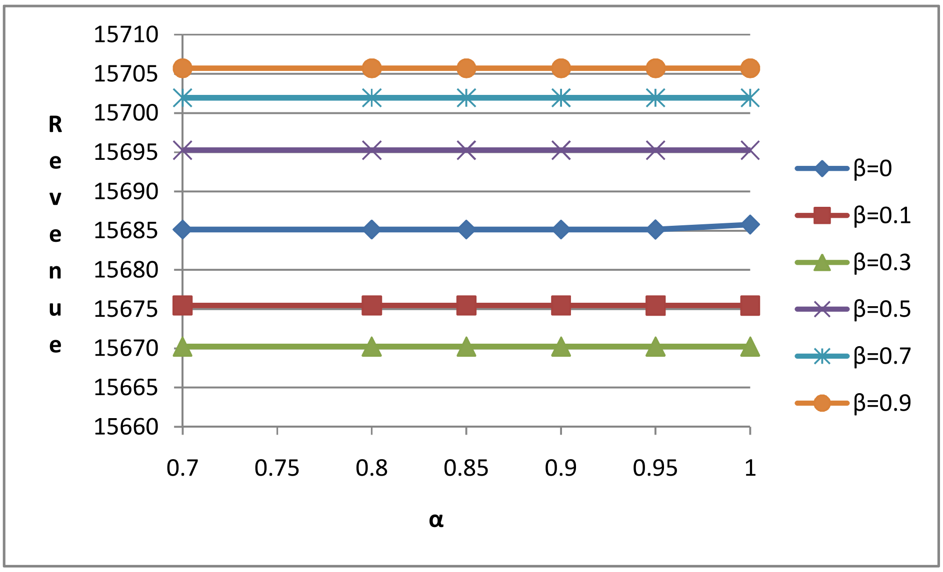


Figure B-8 Revenue vs. (α, β) when $\mathrm{A} = 3$, $\mathrm{B} = 3$ and Number of users $= 100$

REFERENCES

[1] ZenithOptimedia. "Advertising Expenditure Forecasts July." Press Release, 2009

[2] http://www.zenithoptimedia.com

[3] http://secondlife.com/

[4] http://www.lively.com

[5] G. Adomavicius, and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 734-749, 2005

[6] S.M. Bae, S. C. Park, S. H. Ha "Fuzzy Web Ad Selector Based on Web Usage Mining", *IEEE Intelligent Systems*, vol. 18 (6) pp.62 -69, 2003

[7] P. Kazienko, "Multi-agent System for Web Advertising", *9th International Conference on Knowledge-Based In-telligent Information and Engineering Sys.*, *LNAI* 3682, pp. 507-513, 2005

[8] N. Zhou, Y. Chen, and H. Zhang, "Study on Personalized Recommen-dation Model of Internet Advertise-ment", *IFIP International Federation for Information Processing,* Springer Boston, vol. 252, pp.176-183, 2008

[9] R. R. Yager, "Targeted E-commerce Marketing Using Fuzzy Intelligent Agents", *IEEE Intelligent Systems,* vol. 15 (6), pp.42-45, 2000

[10]M. Langheinrich, A. Nakamura, N. Abe, T. Kamba and Y.Koseki. "Unintrusive customization techniques for Web Advertising."*Computer Networks 31*,pp. 1259-1272, 1999

[11] Tomlin, J.A. "An entropy approach to unintrusive targeted advertising on the Web."*Computer Networks*, pp.767-774, 2000

[12] Nakamura A. and, Abe N. "Improvements to the Linear Programming Based Scheduling of Web Advertisements."*Electronic Commerce Research 5*, pp.75-98, 2005

[13] Adler M., Gibbons P.,and Matias Y. "Scheduling-Space Sharing for Advertising."*Journal of Scheduling 5*, pp.103-119,2002.

[14] Amiri A. and, Menon S. “Efficient Scheduling of Banner Advertisements.”*ACM Transactions on Internet Technology 3*, pp.334-346,2003.

[15] Kumar S, Jacob V. and Sriskandarajah C. “Hybrid Genetic Algorithms for Scheduling Advertising On a Web Page.”*Proceedings of the Twenty-Second International Conference on Information Systems(ICIS).* New Orleans, LA, 2001.

[16] Kumar S., Dawande M., and Mookerjee V.S. “Optimal Scheduling and Placement of Internet Banner Advertisements.”*IEEE Transactions on Knowledge and Data Engineering 19*, pp.1571-1584,2007.

[17] P. Chatterjee, D.L. Hoffman and T.P. Novak, “Modeling the Clickstream: Implications for Web-based Advertising Efforts,” *Vanderbilt University Sloan Center for Internet Retailing*, May, 1998 (available at *http://elab.vanderbilt.edu/researchpapers.htm*).

[18] Novak T.P. and Hoffman D.L. “New Metrics for New Media: Toward the Development of Web Measurement Standards” World Wide Web Journal, Winter, 2(1), pp. 213-246, 1997