Expert Systems with Applications 41 (2014) 1593–1600

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Identifying web sessions with simulated annealing

Tomás Arce [a,1], Pablo E. Román [b], Juan Velásquez [c], Víctor Parada [d,*]

[a] Departamento de Ingeniería Informática, Universidad de Santiago de Chile, Av. Ecuador 3659, Estación Central, Santiago, Chile
[b] Center of Mathematical Modelling (CMM) UMI CNRS 2807, Universidad de Chile, Av. Blanco Encalada 2120, Piso 7, Santiago, Chile
[c] Departamento de Ingeniería Industrial, Universidad de Chile, República 701, Santiago, Chile
[d] Departamento de Ingeniería Informática, Universidad de Santiago de Chile, Av. Ecuador 3659, Estación Central, Santiago, Chile

ARTICLE INFO

Keywords:
Web usage mining
Web session
Simulated annealing

ABSTRACT

Delivery of efficient service through a web site makes it compulsory in the redesigning stage to take into account the behavior of the users, which can be studied by means of a web log file that partially records information about user visits. The reconstruction of all of the sequences of pages that are visited by users who browse a web site is known as the web sessionization problem, and it has been formulated by means of an integer programming model; however, because a web log can accumulate a large amount of information, it is necessary to reconstruct the sessions over a period of weeks or months, thus the solution to this problem requires a long computational processing time. This paper presents a heuristic approach based on simulated annealing for the sessionization problem. Using this approach, it has been possible to reduce the processing time up to 166 times compared to the time that is required for the integer programming model. Furthermore, the metaheuristic solution finds new optimum values, which achieve increases on the order of 17% in the best cases.

1. Introduction

The Internet has become a flourishing source of data for different research fields related to social networks, such as sociology (Lin, Jheng, & Yu, 2012; Nohuddin et al., 2012), marketing (Fong, Zhou, Hui, Tang, & Hong, 2012; Wang, Ting, & Wu, 2013) and computer science (Devi, Devi, Rani, & Rao, 2012; Yin & Guo, 2013). Also, the massive usage of the Internet drives many lucrative businesses such as e-commerce. Thus, it is increasingly necessary for web sites to be designed with a structure that makes it easy for the user to obtain the service (Wang & Ren, 2009). To that end, it is necessary to study the behavior of the users which is partially kept in a privacy-compliant data file saved on the web server known as a web log. Legislation in several countries forbids the storage of personal information in order to safeguard personal privacy (Mayer & Mitchell, 2012). In order to follow national laws, internet companies must rely on the anonymity of web logs for extracting information about their user preferences and browsing behavior (Velásquez, 2013; Velásquez & Palade, 2008). The generated browsing sequences represent an input source for discovering the behavior patterns of users who visit a web site (Cooley, Mobasher, & Srivastava, 1999; Kosala & Blockeel, 2000; Tao, Hong, Lin, & Chiu, 2009).

Every time a user requests a web page or some resource contained on it such as images, videos, and sounds, a new record is created in the web log. The generated information allows the detection of the most visited pages, the common page access sequences, the users' preferred contents and even the generation of user profiles (Choi & Lee, 2009; Nasraoui, Soliman, Saka, Badia, & Germain, 2008). Such detection is called web usage mining oriented toward extracting knowledge from web logs (Román, L'Huillier, & Velásquez, 2010), which requires the extraction of individual sequences of a user's web interaction while maintaining anonymity (Mayer & Mitchell, 2012). The following data are typically recorded in the file: the IP address from which the inquiry is made, the time and date of the inquiry, the requested resource, a code indicating the result of the operation, the number of bytes transferred in the request, a string that contains the address of the site from which the present request was originated and the browser and operating system that was used. The generated browsing sequences represent an input source for discovering the behavior patterns of users who visit a web site (Cooley et al., 1999; Kosala & Blockeel, 2000; Tao et al., 2009).

With expert assistance the information stored in the web log can be used to support decision-making that can help in restructuring a web site to improve access to the preferred contents; the information can also be used to detect market segments based on the buying behavior of the users and to implement resource suggestion systems for the users. The sequence of individual interactions is called a session and the reconstruction of all of the page

* Corresponding author. Tel.: +56 2 7180900.
E-mail addresses: tomas.arcec@usach.cl (T. Arce), proman@dii.uchile.cl (P.E. Román), jvelasqu@dii.uchile.cl (J. Velásquez), victor.parada@usach.cl (V. Parada).
[1] Tel.: +56 2 7180900.

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.08.056

sequences visited by the users during their browsing through a web site is known as the web sessionization problem (WSP) (Berendt, Mobasher, Spiliopoulou, & Wiltshire, 2001; Chitraa & Selvdoos, 2010; Huynh & Miller, 2009; Poženel, Mahnic, & Kukar, 2010; Velásquez & Palade, 2008). A correct reconstruction of the sessions must take into account that the validity of the generated patterns depends largely on the credibility of the sessions obtained (Berendt et al., 2001). In spite of that, the reconstruction of sessions is not a trivial process because of factors that hinder reconstruction, such as proxy servers or the use of cache memory in the client's web browser.

The techniques that solve the WSP from a web log must deal mainly with the nonexistence of a clear identification of the users. Matching the IP address as a single criterion is not sufficient, because when multiple users have access to a site through a proxy server, they are registered in the web log with the proxy server's address. To partially remove the identification error, the techniques use the IP address and the web browser that was used by the web site visitor as identification criteria (Pirolli, Pitkow, & Rao, 1996). Other techniques directly track user operations by using cookies with client-side scripts obtaining accurate sessions, but they are not recommended since they violate user privacy laws (Mayer & Mitchell, 2012). Therefore, the web usage mining on web logs is a safe way to analyze user preferences, as long as we take into account that in general a high quality sessionization is not possible by means of machine learning techniques that are known to be subject to data error (Román et al., 2010).

Current improvement of the web usage mining processing relies on accurately solving the WSP, for which several heuristics have been proposed. The most widely used technique to tackle the WSP is the time heuristic, which considers that the sessions have a time limit (Catledge & Pitkow, 1995; Cooley, Mobasher, & Srivastava, 1997; Huynh & Miller, 2009). This technique groups the records according to the IP address and the web browser that was employed by the user; it arranges the records of each resultant group in a temporal order and then obtains the sessions, ensuring that the first and last records of a given session do not exceed a time limit. The major drawback of this algorithm consists of the null consideration of the hyperlink topology of the web site. A revised version of this heuristic considers the site's link structure together with a temporal criterion (Pirolli et al., 1996), in addition to the criteria that the consecutive records of the same session must refer to pages between which there is a link. Nowadays, with modern web browser navigation it becomes hard to track the information due to the multiple ways that pages are cached, loaded, and navigated. Multitab navigation, back button browsing and history jumps are commonly not reflected on web logs (Román et al., 2010), so hyperlink topology restriction is relaxed from being a strong restriction on session properties.

A recent approach addresses the WSP as an optimization problem by defining an integer programming model (Román, Dell, & Velásquez, 2010). All of the possible reconstructions of sessions from a given web log constitute the feasible solution space of the problem. The choice of a specific reconstruction implies a search for the reconstruction that has a maximum value in terms of a specific objective function. Web hyperlink topology and time sequencing are incorporated as restrictions in the model. Relaxed browsing behavior could even be easily included (Dell, Román, & Velásquez, 2009). However, because a web log in a single day can accumulate an enormous number of records and the supporting decision making requires the reconstruction of sessions over a period of weeks or months, the number of variables of the integer programming models is huge for the currently available capacity for solving integer programming problems. Solving even small instances of the problem requires several hours of computing time. The performance is even worse when considering more common browsing behaviors such as parallel tabs and back buttons. The ideal situation would consider the existence of an algorithm that can process the information in real time by requiring a short computing time.

A novel algorithm for solving the WSP was presented in Bayir, Toroloslu, Demirbas, and Cosar (2012). The algorithm is based on graph modeling of the sessions that are constructed considering maximal path length, hyperlink topology and back button browsing. They found improved accuracy in recovering sessions relative to previous models. However, their major drawback comes from the theoretical justification of the model affecting its reproducibility.

Instead of generating the optimum solution by means of an algorithm that solves the integer programming problem, we obtain a good quality solution using a small amount of computing time, by means of Simulated Annealing (SA) which is a metaheuristic that emulates the physicochemical process that takes place in the cooling of pure substances, systematically generating a new solution from the current solution and allowing, at some instant, the choice of a *poor solution*, with a certain probability that decreases with time (Kirkpatrick, Gelatt, & Vecchi, 1983; Talbi, 2009). This method has been shown to be very efficient for solving problems belonging to the *NP-Hard* class, such as the design of electronic circuits, the reconstruction of images, the generation of roads, set partition problems and planning problems (Suman & Kumar, 2005). This paper presents an approach that is based on simulated annealing for the WSP for the purpose of reconstructing each session of the users that visit a web site.

The second section presents the WSP representation under an SA, identifying the elements needed for the experimental phase. The third section presents the main results, and in the last section, the main conclusions are given.

2. Methods and materials

2.1. The web sessionization problem

The model founded on integer programming (Dell, Roman, & Velasquez, 2008) is based on ensuring that a session is constituted by a set of records that share the same IP address and the same web browser. For consecutive records within a session, the constraints ensure the maintenance of the link structure of the site and that the session does not go beyond a certain time limit. Thus, if a record is found directly after another record within the same session, then the following are true:

- Both share the same IP address and web browser.
- There is a link between the referenced page in both records.
- The time difference between the recorded application for both records does not exceed a certain *mtp* value or time window.

Let

- r and r' be records of a web log;
- $o$be the order of a record in a given session, $o = 1,2,\ldots,O$, where is the maximum size that a session can have;
- s be the identifier of a session;
- C_o be the value of the coefficient of the objective function when there is a record assigned to position o;
- x_{ros} be the binary decision variable, which has a value of 1 when record r is assigned to position o in a given session s and a value of 0 in any other case;
- $Bpage(r)$ be the set of records that can be directly before record r in the same session, according to the criteria of having the same IP address, corresponding to the same web browser, having a link between consecutive records and maintaining the time window between consecutive records;

- *first* be the set of records that are always found in the same position in a session. It is possible to calculate this set by first getting $bpage_r$ for all of the web log records. Therefore, a record $r \in first$ if $bpage(r) = \varnothing$.

The integer programming model is the following:

$$\text{ISP}: \text{Maximize } z = \sum_{ros} C_o x_{ros} \tag{1a}$$

s.t.

$$\sum_{os} x_{ros} = 1, \quad \forall r \tag{1b}$$

$$\sum_{r} x_{ros} \leq 1, \quad \forall o, s \tag{1c}$$

$$x_{r,o+1,s} \leq \sum_{r' \in bpage(r)} x_{r'os}, \quad \forall r, o, s \tag{1d}$$

$$x_{ros} = 0, 1 \quad \forall r, o, s \tag{1e}$$

$$x_{ros} = 0 \quad \forall r, \in first, o > 1, s \tag{1f}$$

The first constraint establishes that a record can only be in a single session and in a given order within it. The second constraint establishes that there can only be a unique record in a given session and a position within it. The third constraint establishes the correct arrangement of the records of the same session, complying with the time, site structure and the same IP address and web browser combination conditions. The value of C_o is used to give a bonus to sessions of a given size. For example, by setting a value of $C_o = 1$ when $o = 2$ and $C_o = 0 \; \forall \; o \neq 2$, we are favoring a large number of size 2 sessions in the final reconstruction. The value of the objective function for a sessionization is calculated taking into account the length of each of the generated sessions (l) and then adding the value of C_o that corresponds to each position of the session. For example, for a single session of length $l = 5$, the value of the objective function is the following:

$$\sum C_o = C_1 + C_2 + C_3 + C_4 + C_5 \tag{2}$$

Several measures can be used to reproduce the sessions. Functions C_o^1 through C_o^4 are used in this paper and are presented in Table 1. All of the functions are monotonically increasing in o with the purpose of maximizing small-size sessions or else maximizing large-size sessions.

2.2. Quality of the reconstruction

The quality of the reconstruction is measured with respect to the power law for the size of the sessions on a web site (Huberman, Pirolli, Pitkow, & Lukose, 1998; Oliveira et al., 2006). This law states that most visits to a web site are concentrated on a small number of pages and the rest of the pages receive a smaller number of visits. Thus, the quality of the reconstruction can be verified by revising the coefficient of correlation r^2 and the standard error S, obtained from a linear regression performed on the logarithm of the size of the session and the number of sessions generated by the reconstruction.

Table 1
Different coefficients C_o for the objective function.

C_o
$C_o^1 = log(o)$
$C_o^2 = 1.5log(o) + (o-3)^2/12o$
$C_o^3 = o$
$C_o^4 = o^2$

2.3. Simulated annealing (SA)

The SA algorithm is a metaheuristic optimization procedure that is inspired by the cooling process of molten metals, and it has been used to find good quality solutions to various combinatorial optimization problems (Kirkpatrick et al., 1983; Talbi, 2009; Černý, 1985). In the metallurgical process, the initial state is the molten metal, and the temperature is decreased in a controlled manner, allowing the formation of intermediate equilibrium states until a solid crystalline structure is achieved. During the process, an internal form of the metal's energy is decreased. Following this analogy, to solve an optimization problem, the solution space is visited, varying a parameter T that corresponds to the temperature so that, at a high value of T, the probability of accepting solutions worse than the current solution is high. As T decreases, the algorithm is gradually converted into a local search algorithm. Regulation of the acceptance probability takes place by means of a Boltzmann distribution. The adequate representation of an optimization problem that is solved by SA requires defining a rule to generate neighboring solutions of the current solution, the function that must be optimized and an initial solution. The proper algorithm parameters must also be defined: the initial value of T, a way of decreasing it gradually and the number of iterations that are performed over the internal cycle $N(t)$. The procedure for solving a maximization problem is presented in the pseudocode depicted in Fig. 1.

2.4. Representation of a solution with SA

A solution of a WSP is represented using a matrix $X_{(mxn)}$, where m is equal to the maximum size of a session and n is equal to the number of records contained in the web log partition. Then, each element x_{ij} of the matrix corresponds to the identification number of the record that is in position i of session j. Thus, for a given solution, the column index is the identifier of the reconstructed session.

2.5. Generation of an initial solution

The initial solution generated uses a variation of the *Links* Heuristic (Pirolli et al., 1996), ensuring that all of the consecutive records within a session do not exceed a maximum time window. For two records, r_1 and r_2, to be located consecutively in the same session, the following must be satisfied:

```
SA algorithm
Input: Min f(x) s.t. x ∈ Ω;
Output: x*;
Generate an initial solution x0 ∈ Ω; x* = x0;
Define T > 0; t = 0;
Repeat
    Repeat
        Generate solution xj neighboring xi;
        δ = f(xj) - f(xi);
        If δ > 0 then xi = xj;
        Otherwise If random (0, 1) < exp (-δ / T) then xi = xj;
        If xi < x* then x* = x0;
    Until N(t) iterations are completed
    t = t +1; T = T(t);
Until the stop criterion is reached.
```

Fig. 1. SA for a maximization problem.

```
Function GeneratesInitialSolution ()
    MTP    : Maximum time difference between consecutive records.
    SMAX   : Maximum length of a session.
    newSession=1;
    session=1;
    For each record r in the web log
        If (newSession = 1)then
            Create a new session with record r in the first position.
        Or else
            If
            (The size of the session is Not equal to SMAX)
            AND
            (The last record of the session, r - 1, has
            the same address as record r)
            AND
            (There is a link between r - 1 and r)
            AND
            (r.timeStamp – (r – 1).timeStamp ≤ MTP) then
                Add r to the current session.
            Or else
                newSession =1;
                r --;
            End If
        End If
    End For
End Function
```

Fig. 2. Algorithm for generating the initial solution.

- Both records r_1 and r_2 have the same IP address.
- Record r_1 has a *link* with r_2.
- The records have visiting times within a predefined time window *mtp*.
- A predefined maximum session time is not exceeded.

Fig. 2 presents the algorithm for generating the initial solution. In the algorithm, the records are assigned to a session following the viability conditions that are described above.

2.6. Generation of the neighboring solution

The generation of a neighboring solution consists of three steps: choosing a session randomly, deleting the chosen session and reassigning each of the records of the deleted session. In the last step, there are two possibilities; the record is either assigned to other existing sessions that comply with the maximum number of records per session, the maximum time difference and the existence of a link between consecutive records, or it is assigned to a new session where the record occupies the first position. A record can be relocated as follows:

- At the beginning of the session. The chosen record r_e must have a link to the first record r_1, and the time difference between r_e and r_1 must not exceed the maximum time window *mtp*.
- In the middle of the session. If r_i and r_{i+1} are two consecutive records within the session, there must be a *link* that goes from r_i to r_e and another one that goes from r_e to r_{i+1} for the chosen record r_e to be located between r_i and r_{i+1}; the time difference between r_i, r_e and between r_e, r_{i+1} must not be greater than the maximum time window *mtp*.
- At the end of the session. The existence of a link between the chosen record r_e and the last record r_f must be verified, and the time difference between r_f and r_e must not exceed the time window *mtp*.

2.7. Evaluation function

The evaluation function considered is the proper objective function (1a) of the integer programming problem, keeping in mind the four possibilities for C_o that are presented in Table 1.

2.8. Selection of SA parameters

With the aim of searching for the adequate adaptation of SA for the WSP, we set the parameters according to the following criteria for each case:

- T_0. A scheme has been adopted with a variable T_0 that allows changing the execution time of the algorithm depending on the effort needed for the search, measured according to the size of the instance to be solved. In this way, T_0 is chosen based on the maximum approximate difference in the objective function (Aarts & Lenstra, 1997), as follows:

$$\Delta F^{max} = f_{max} - f_{min} \tag{3}$$

In the problem's context, the greatest decrease of C_o occurs when the largest session is fragmented into smaller unit sessions. The difference gives Eq. (4) as a result:

$$\Delta F^{Max} = \sum_{o=1}^{L} C_o - LC_1 \tag{4}$$

where L is equal to the largest session of the initial solution generated.

Considering the natural logarithm of the acceptance probability p, $ln(p) = -\Delta F^{max}/T_0$, with $p = 0.99$, we get T_0 such that when a solution gets worse by a magnitude of ΔF^{max}, it will be accepted with a probability of 0.99. Then, T_0 is calculated from $T_0 = -\Delta F^{max}/0.01$.

- Cooling function. A geometric temperature drop is used, given by $Ti = -\alpha\ Ti - 1$, with $\alpha \in (0, 1)$.
- The number of iterations of the internal cycle. The number of iterations is dynamic as the search continues. If a neighboring solution that improves the current solution is generated, the cooling function is applied to decrease T (Cardoso, Salcedo, & Azevedo, 1994; Ravi & Shukla, 2009). To avoid stagnation of the SA at a single T level, the maximum number of iterations equal to the number of possible neighboring solutions of a given solution is considered. An approximation of this number is given by the number of sessions at each level of T; when a neighboring solution is generated, the number of possible movements is equivalent to the number of sessions that can be chosen for their records to be reassigned.
- T_f A fixed value is used that is found during the calibration process.

2.9. Algorithm for WSP

The internal cycle operates with a constant value of T visits; each time, a neighbor solution of the current solution is generated by means of a reassignment of a record from one session to another. It accepts this solution automatically when a better sessionization is generated and accepts it with a specific probability when the sessionization is worse than the current one.

2.10. Equipment used

The implementation of SA was written in the C language and was compiled with *GCC* version 4.3.2. The computer used for both the implementation and the execution of the experiment had an Intel Core 2 Duo 1.8-Ghz processor, with 1 GB of RAM, an 80-GB hard disc, and the *Linux* Operating System, *kernel* version 2.6.26, *Debian* version 5.0 distribution.

2.11. Test instances

To verify the operation of the implementation of the proposed solution, a web log generated by a web site that considered a storage period of one month was used. The site has 172 pages and 1228 links between them. The web log was previously preprocessed to delete the requirements of multimedia objects, access to web mail and error records. After the preprocessing, the total number of records is 102,303. Of the 16,985 IP addresses contained in the file, 16,785 have less than 50 records that represent 98% of the total. In turn, 14,265 IP addresses visit 3 or fewer different pages, which represent 84% of the total IP addresses.

Reconstructing sessions using the mathematical model and starting from a web log leads to an oversized requirement of the hardware resources. However, because a session is composed of records that share the same IP address, it is possible to partition the web log into smaller chunks. In this way, the possible solutions space is reduced, and the problem's natural restrictions are applied to divide it into multiple smaller size problems. By applying the algorithm to each chunk and then integrating the partial results, the resource requirement is reduced considerably. Not all of the partitions have the same interest; there are sets of records that are more interesting for the sessionization, given the possibility that the recorded requests correspond to multiple user accesses that are masked under a single address (which corresponds to the proxy server or the firewall that is being used). To identify these sets, two measures of difficulty are used that are established in relation to a given IP address, the diversity of the record and the number of records. The record diversity of a given IP address is measured in terms of the *entropy of the IP address* (Dell et al., 2008), defined as the following:

$$\sum_{i \in P(IP)} (f_i/n_R) log_b(n_R/f_i) \tag{5}$$

where f_i corresponds to the number of times in which page i was accessed by a given IP address, n_R is the number of records containing that IP address, b is the number of different pages that have been visited from a given IP address and P(IP) is the number of pages visited from the IP address. The entropy takes values between 0 and 1. When the entropy is close to 0, most of the records for a given address correspond to accesses to a single page; when it is close to 1, all of the pages accessed by the address are visited with the same frequency.

A large diversity for an IP address is a good indicator of the difficulty of the records associated with that address, but it is not a sufficient condition, because there can be a large diversity with very few records. That is why, together with entropy, a criterion of a minimum number of records per IP address is used; an IP address with high entropy and a large number of records ensures that the records associated with that address are referred to different pages and that they are numerous, capturing what happens in a web log when multiple users gain access to the site by means of proxy servers.

To construct the test instances, the records with IP addresses that have entropy greater than or equal to 0.5 and a number of records in the web log greater than or equal to 50 are chosen. For the web log used, the number of records chosen under this criterion was 17,709.

2.12. Metrics used

To evaluate and analyze the results obtained, the following metrics are used:

- Percent difference of the objective function, Gap:

$$Gap = |f(x) - f^*(x)|/f^*(x) \tag{6}$$

where $f^*(x)$ is the reference value for the comparison.

- Power Law fit. The quality of the reconstruction is verified by means of the coefficients r^2 and S, obtained from a linear regression over the logarithm of the size of a session and the number of sessions. The values $0 \leqslant r^2 \leqslant 1$ and S allow the evaluation of the degree of fit of the number of sessions versus the size of the sessions with the Power Law. When r^2 takes values close to 1, the model fits perfectly with the Power Law. On the other hand, S measures the standard error of the model in terms of the difference between the real value and the estimated value. The closer it is to 0, the better the model's fit is with real data (Montgomery & Runger, 2006).

2.13. Execution of the experiment

To reconstruct the sessions with SA, we used the values of $\alpha = 0.99$ and $T_f = 0.08$ for the parameters. For each set of instances, an execution of the algorithm considers 10 attempts for each way of assigning coefficient C_o of the objective function. An attempt must be understood as the execution of the algorithm over all of the chunks of the web log.

The assignment functions of C_o presented in Table 1 are considered for the experiment. Subsequently there are 40 attempts for the set of ISP instances and 40 attempts for the set of instances of the metaheuristic, with a total of 80 attempts. Counting the application of the solution to each chunk, a total of 21,200 executions of the SA are performed.

3. Results

The execution of SA obtains the following for each assignment function of C_o and each set of test instances:

- Average attempt: The reconstruction closest to the average value of the objective function calculated from the 10 attempts executed.
- Best attempt: A reconstruction based on the best attempt of the 10 executed, in terms of the value of the objective function.
- Best reconstruction: A reconstruction that is obtained by choosing, for each chunk, the best sessionization in terms of the value of the objective function. The solutions of each of the best chunks are joined to form the best possible solution from the attempts that are executed up to some point. This reconstruction is obtained from the processing of the results, after the execution of the implementation.

Furthermore, SA provides information on the value of the objective function that obtained the solution, the computer time and the quality of the reconstruction of the sessions in terms of the Power Law in relation to the size of the sessions.

3.1. General results

The general results that are obtained during the execution of the algorithm are presented in Table 2. The smallest average number of sessions is obtained with C_o^1. The largest session average is obtained with C_o^4. The function C_o^1 yields the shortest processing times, with an average of 2.15 min per attempt and a total test time for the execution of the 10 attempts of 21.52 min. Function C_o^4 takes an average of 3.86 min per attempt, with a total test time of 38.58 min.

Table 3 presents the best attempts for each of the objective functions. The lowest number of sessions was 11,312, obtained

Table 2
Results of SA for WSP.

C_o	Maximum # of sessions	Minimum # of sessions	Average # of sessions	Standard deviation	z_{max}	z_{min}	$z_{promedio}$	Average computer time [min]	Total test time [min]
*Co*1	11321	11312	11317.7	2.83	6481.3	6460.4	6469.9	2.15	21.52
*Co*2	11334	11321	11326.3	4.42	13990.0	13918.5	13957.2	2.41	24.14
*Co*3	11349	11320	11332.5	7.76	32561.0	32391.0	32489.8	2.47	24.74
*Co*4	11352	11335	11340.5	4.81	134926.0	131012.0	132746.8	3.86	38.58

Table 3
Results of the best attempts for each of the objective functions.

C_o	Id attempt	z	r^2	S	No. of sessions	Computer time [min]
C_o^1	10	6481.3	0.943	0.577	11312	2.15
C_o^2	1	13990.0	0.931	0.666	11327	2.41
C_o^3	7	32561.0	0.919	0.707	11326	2.48
C_o^4	6	134926.0	0.941	0.581	11338	3.86

Table 4
Evolution of the value of the objective function for different executions of SA.

	*Co*1			*Co*2			*Co*3			*Co*4		
Attempt	z	Δz	Gap [%]	z	Δz	Gap [%]	z	Δz	Gap [%]	z	Δz	Gap [%]
1	6460.4	–	–	13990.0	–	–	32541.0	–	–	134012.0	–	–
2	6490.7	30.29	0.467	14032.4	42.44	0.302	32688.0	147.0	0.450	136611.0	2599.0	1.90
3	6505.0	14.23	0.219	14041.4	9.03	0.064	32744.0	56.0	0.171	137883.0	1272.0	0.93
4	6511.2	6.25	0.096	14052.4	10.93	0.078	32770.0	26.0	0.079	138246.0	363.0	0.26
5	6514.3	3.12	0.048	14063.7	11.33	0.081	32794.0	24.0	0.073	138811.0	565.0	0.41
6	6519.6	5.30	0.081	14068.6	4.92	0.035	32839.0	45.0	0.137	139885.0	1074.0	0.77
7	6521.4	1.77	0.027	14072.1	3.50	0.025	32847.0	8.0	0.024	140056.0	171.0	0.12
8	6524.0	2.57	0.039	14080.2	8.02	0.057	32893.0	46.0	0.140	140056.0	0.0	0.00
9	6525.1	1.10	0.017	14080.2	0.00	0.000	32898.0	5.0	0.015	140126.0	70.0	0.05
10	6528.0	2.97	0.046	14080.3	0.18	0.001	32898.0	0.0	0.000	140126.0	0.0	0.00

Table 5
Results obtained with SA and CPLEX for WSP.

C_o	z (ISP)	z (SA) av. attempt	z (SA) best value	Gap [%]
C_o^1	5953.3	6470.4	6528.0	9.66
C_o^2	13229.4	13956.5	14080.3	6.43
C_o^3	30783.0	32482.0	32898.0	6.87
C_o^4	119450.0	132485.0	140126.0	17.31

with C_o^1, and the largest number of sessions, 11,338, was obtained with C_o^4. The shortest and longest times occur with C_o^1 and C_o^4, with 2.15 and 3.86 min, respectively. The best fit with the Power Law occurs with C_o^1.

3.2. Evolution of the best reconstruction

Because the solution is based on a random search method, it is possible that a new execution over a specific chunk will find a better solution than one found in a previous execution. Executing SA several times over the same web log, it is possible to choose the best solution for each of the chunks. Because the sessionization of each chunk is independent of the others, the *best reconstruction* can be generated based on a certain number of attempts. This way of operating allows refining the quality of a sessionization by means of successive executions. Table 4 shows the value of the objective function for the best possible reconstruction up to a given attempt. The value of z is presented for every function C_o^1 through C_o^4, as is the difference in z that is produced by the new execution of SA with respect to the previous execution.

It can be seen that the values of the objective function tend to converge at a given point, causing the largest increase in the first four attempts, with more than 70%. Specifically, the increases remain until attempt number 10, except for C_o^3 and C_o^4, which converge after the eighth attempt. The best refining is produced with C_o^4, which shows an increase of 4.56%.

If a unique reconstruction is sought that is stable in terms of the value of the objective function, it is seen that the road to follow is to execute SA more than once with C_o^4. Alternatively, if efficiency is sought while sacrificing the stability of the sessionization obtained, it is recommended to execute SA only once.

3.3. Comparison with the exact method

The integer programming model was solved with CPLEX («IBM – IBM – Mathematical Programming», 2011) version 10.1.0, together with GAMS («GAMS», s.f.), a software system used for defining optimization models. The experiment was executed on a computer with 1.6 Ghz and 2 GB RAM. CPLEX was executed with a time limit of 300 s for each chunk. When this limit was reached, the software returned the best solution found up to that moment for this chunk.

Table 5 shows the value of the objective function for each of the assignment functions. The results show that SA improves the optimum found by CPLEX for all of the assignment functions, with an increase with respect to the optimum value that goes from 9.66% to 17.31%, respectively.

Table 6 shows the processing time of CPLEX, t_{ISP}, and the average execution time for each SA attempt, t_{SA}. SA has a significantly

Table 6
Computer time for ISP and SA.

C_o	t_{ISP} [min]	t_{SA} [min]	Speedup
C_o^1	359.1	2.15	166.78
C_o^2	372.5	2.40	154.90
C_o^3	379.9	2.47	153.51
C_o^4	96.3	3.86	24.96

Table 7
Quality of the reconstruction of sessions based on Power Law fit.

Method	r^2	S
Time heuristic	0.915	0.663
ISP-CPLEX model (C_o^2)	0.978	0.383
SA (Average attempt, C_o^4)	0.943	0.587
SA (Best attempt, C_o^1)	0.943	0.577
SA (Best reconstruction, C_o^4)	0.943	0.573

shorter execution time than the integer programming model in all of the assignment functions of C_o. The speedup reaches 166.78 for C_o^2, which is the maximum value of this indicator. The smallest speedup is obtained with C_o^4, with a 24.96-fold reduction in processing time.

To study the quality of the reconstruction based on the fit with the Power Law, we obtain the values of r^2 and S for the time heuristic. In this case, the records are arranged temporarily in increasing order for each group. Each record is chosen and added to a session in such a way that the consecutive records of a given session do not have a time difference greater than 300 s. Table 7 shows the best-fitting values obtained by each of the methods studied.

Based on the above values, the best fit is obtained with the ISP model solved with CPLEX using C_o^2. Both of the reconstructions that are obtained with ISP and those obtained by SA exceed the time heuristic in terms of that metric, which presents the worst values in both the standard error and in r^2.

4. Discussion and conclusions

The implementation of Simulated Annealing to solve the WSP improves the value of the objective function for each of the assignment functions, with a gap that varies between 9.66% and 17.31%. The speed for obtaining the results is one of the main strengths of SA over the exact method. In the best case, the reduction is of the order of 166 times the execution time of the exact method.

Executing SA several times on the same set of instances allows new solutions to be obtained that are better than those that can be obtained with a single execution. The study of the magnitude of the increments in relation to the number of executions of the solution yielded an interesting convergence of the solutions to a given value of the objective function. That allows the achievement of a potentially stable and unique reconstruction depending on the number of executions, compared with only a single execution of the solution. Clearly, this implies an increase in processing time that must be evaluated according to the accuracy required for the reconstruction of the sessions.

From the standpoint of the sessionization problem, the fit of the reconstructions obtained by SA in terms of the Power Law for the size of the sessions is better using CPLEX but comes at a high computational cost.

Acknowledgements

This research was partially supported by the Millennium Institute Complex Engineering Systems ICM: P-05-004-F, FBO16.

References

Aarts, E., & Lenstra, J. K. (Eds.). (1997). *Local search in combinatorial optimization*. John Wiley & Sons, Inc.

Bayir, M. A., Toroloslu, I. H., Demirbas, M., & Cosar, A. (2012). Discovering better navigation sequences for the session construction problem. *Data & Knowledge Engineering, 73*, 58–72.

Berendt, B., Mobasher, B., Spiliopoulou, M., & Wiltshire, J. (2001). Measuring the accuracy of sessionizers for web usage analysis. In *Proceedings of the workshop on web mining at the first SIAM international conference on data mining*, 7–14, Chicago, IL.

Cardoso, M., Salcedo, R., & Azevedo, S. (1994). Nonequilibrium simulated annealing: A faster approach to combinatorial minimization. *Industrial Engineering Chemical Research, 33*, 1908–1918.

Catledge, L. D., & Pitkow, J. E. (1995). Characterizing browsing strategies in the world-wide web. *Computer Networks and ISDN Systems, 27*(6), 1065–1073.

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications, 45*(1), 41–51.

Chitraa, V., & Selvdoos, A. (2010). A survey on preprocessing methods for web usage data. *IJCSIS International Journal of Computer Science and Information Security, 7*(3), 78–83.

Choi, J., & Lee, G. (2009). New techniques for data preprocessing based on usage logs for efficient web user profiling at client side. In *Proceedings of the 2009 IEEE/WIC/ACM international joint conference on web intelligence and intelligent agent technology* (Vol. 3, pp. 54–57). Washington, D.C.

Cooley, R., Mobasher, B., & Srivastava, J. (1997). Web mining: information and pattern discovery on the world wide web. In *Proceedings of the 9th international conference on tools with artificial intelligence* (pp. 558–567). Newport Beach, California.

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems, 1*(1), 5–32.

Dell, R., Roman, P., & Velasquez, J. (2008). Web user session reconstruction using integer programming. In *IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology* (Vol. 1, pp. 385–388). Sydney. doi:http://dx.doi.org/10.1109/WIIAT.2008.181.

Dell, R., Román, P., & Velásquez, J. (2009). Web User Session Reconstruction with back button browsing. In *Knowledge-based and intelligent information and engineering systems, lecture notes in computer science* (Vol. 5711, pp. 326–332). doi:http://dx.doi.org/10.1007/978-3-642-04595-0_40.

Devi, B. N., Devi, Y. R., Rani, B. P., & Rao, R. R. (2012). Design and implementation of web usage mining intelligent system in the field of e-commerce. *Procedia Engineering, 30*, 20–27.

Fong, A. C. M., Zhou, B., Hui, S., Tang, T., & Hong, G. (2012). Generation of personalized ontology based on consumer emotion and behavior analysis. *IEEE Transactions on Affective Computing, 3*(2), 152–164.

GAMS. *GAMS Home Page*. Retrieved from http://www.gams.com/.

Huberman Pirolli Pitkow & Lukose (1998). Strong regularities in world wide web surfing. *Science, 280*(5360), 95–97.

Huynh, T., & Miller, J. (2009). Empirical observations on the session timeout threshold. *Information Processing & Management, 45*(5), 513–528.

IBM – IBM – Mathematical programming: linear programming, mixed-integer programming and quadratic programming – IBM ILOG CPLEX optimizer – software. Retrieved from: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

Kirkpatrick, S., Gelatt, D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. *SIGKDD Explorations, 2*, 1–15.

Lin, S., Jheng, Y., & Yu, C. (2012). Combining ranking concept and social network analysis to detect collusive groups in online auctions. *Expert Systems with Applications, 39*(10), 9079–9086.

Mayer, J. R. & Mitchell, J. C. (2012). Third-party web tracking: Policy and technology, security and privacy. In *IEEE symposium on security and privacy* (pp. 413–427). San Francisco.

Montgomery, D. C., & Runger, G. C. (2006). *Applied statistics and probability for engineers* (4th ed.). Wiley.

Nasraoui, O., Soliman, M., Saka, E., Badia, A., & Germain, R. (2008). A web usage mining framework for mining evolving user profiles in dynamic web sites. *IEEE Transactions on Knowledge and Data Engineering, 20*(2), 202–215.

Nohuddin, P. N. E., Coenen, F., Christley, R., Setzkorn, C., Patel, Y., & Williams, S. (2012). Finding "interesting" trends in social networks using frequent pattern mining and self organizing maps. *Knowledge-Based Systems, 29*, 104–113.

Oliveira, J. G., Dezso, Z., Goh, K.-I., Kondor, I., Barabasi, A.-L., & Vazquez (2006). Modeling bursts and heavy tails in human dynamics. *Physical Review E – Statistical, Nonlinear and Soft Matter Physics, 73*(3 Pt 2), 036127.

Pirolli, P., Pitkow, J., & Rao, R. (1996). Silk from a sow's ear: Extracting usable structures from the web. In *Proceedings of the SIGCHI conference on human factors in computing systems: common ground* (pp. 118–125). Vancouver.

Poženel, M., Mahnic, V., & Kukar, M. (2010). Separation of interleaved web sessions with heuristic search. In *Proceedings of the ICDM '10, IEEE international conference on data mining* (pp. 411–420). Sidney.

Ravi, S., & Shukla, S. (Eds.). (2009). *Fundamental problems in computing: An analysis of several heuristics for the traveling salesman problem*. Dordrecht, Netherlands: Springer Netherlands.

Román, P. E., Dell, R. F., & Velásquez, J. D. (2010). Advanced technique in web data pre-processing and cleaning. In J. D. Velásquez & L. Jain (Eds.). *Advances in techniques in web intelligence-1* (Vol. 311, pp. 19–48). Heidelberg: Springer Verlag.

Román, P. E., L'Huillier, G., & Velásquez, J. D. (2010). Web usage mining. In J. D. Velásquez & L. Jain (Eds.). *Advances in techniques in web intelligence-1* (Vol. 311, pp. 143–165). Heidelberg: Springer Verlag.

Suman, B., & Kumar, P. (2005). A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society, 57*(10), 1143–1160.

Talbi, E. G. (2009). *Metaheuristics: From design to implementation*. John Wiley and Sons.

Tao, Y. H., Hong, T. P., Lin, W. Y., & Chiu, W. Y. (2009). A practical extension of web usage mining with intentional browsing data toward usage. *Expert Systems with Applications, 36*(2), 3937–3945.

Velásquez, J. D. (2013). Web mining and privacy concerns: some important legal issues to be consider before applying any data and information extraction technique in web-based environments. *Expert Systems with Applications, 40*(13), 5228–5239.

Velásquez, J. D., & Palade, V. (2008). *Adaptive web sites: A knowledge extraction from web data approach*. Amsterdam: IOS Press.

Wang, T., & Ren, Y. (2009). Research on personalized recommendation based on web usage mining using collaborative filtering technique. *WSEAS Transactions on Information Science and Applications, 6*(1), 62–72.

Wang, K., Ting, I., & Wu, H. (2013). Discovering interest groups for marketing in virtual communities: An integrated approach. *Journal of Business Research, 66*(9), 1360–1366.

Yin, P., & Guo, Y. (2013). Optimization of multi-criteria website structure based on enhanced tabu search and web usage mining. *Applied Mathematics and Computation, 219*(24), 11082–11095.