

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
7 February 2008 (07.02.2008)

PCT

(10) International Publication Number  
**WO 2008/016591 A2**

(51) International Patent Classification:  
*G06Q 30/00* (2006.01) *G06F 17/30* (2006.01)

(74) Agent: **LAW OFFICE OF ROBERT O. BOLAN**; P.O.  
Box 36, Bellevue, WA 98009 (US).

(21) International Application Number:  
PCT/US2007/017084

(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(22) International Filing Date: 30 July 2007 (30.07.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
11/497,085 31 July 2006 (31.07.2006) US

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(71) Applicant (*for all designated States except US*): **YAHOO! INC.** [US/US]; 701 First Avenue, Sunnyvale, CA 94089 (US).

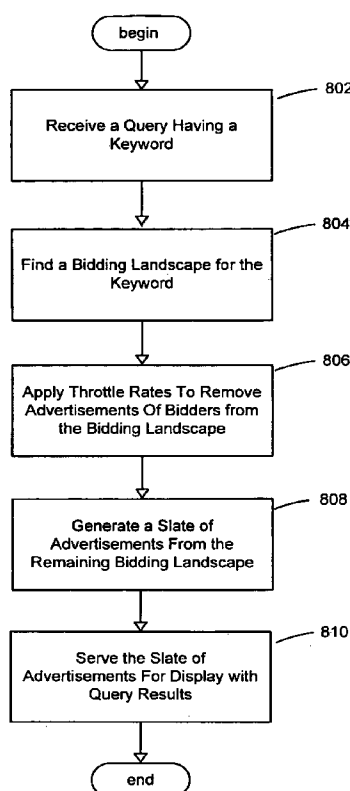
(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **MENDELE-VITCH, Ofer** [IL/US]; 3064 Whisper Wave Circle, Redwood City, CA 94065 (US). **TOMLIN, John, Anthony** [US/US]; 850 Hanover Avenue Sunnyvale, Sunnyvale, CA 94087 (US).

Published:

— *without international search report and to be republished upon receipt of that report*

(54) Title: SYSTEM AND METHOD FOR SCHEDULING ONLINE KEYWORD AUCTIONS SUBJECT TO BUDGET CONSTRAINTS



(57) Abstract: An improved system and method for scheduling online keyword auctions subject to budget constraints is provided. A linear programming model of slates of advertisements may be created for predicting the volume and order in which queries may appear throughout the day for use in allocating bidders to auctions to optimize revenue of an auctioneer. Each slate of advertisements may represent a candidate set of advertisements in order of optimal revenue to an auctioneer. Linear programming using column generation with the keyword as a constraint and a bidder's budget as a constraint may be applied to generate a column that may be added to a linear programming model of slates of advertisements to determine optimal revenue to an auctioneer. Upon receiving a query request, a slate of advertisements that may provide optimal revenue to the auctioneer may be output for sending to a web browser for display.

WO 2008/016591 A2

**SYSTEM AND METHOD FOR SCHEDULING ONLINE KEYWORD  
AUCTIONS SUBJECT TO BUDGET CONSTRAINTS**

5                                   **CROSS-REFERENCE TO RELATED APPLICATIONS**

The present invention is related to the following United States patent application, filed concurrently herewith and incorporated herein in its entirety:

                  "System and Method for Optimizing Throttle Rates Of  
10 Bidders In Online Keyword Auctions Subject To Budget  
Constraints," Attorney Docket No. 1270.

**FIELD OF THE INVENTION**

15           The invention relates generally to computer systems, and more particularly to an improved system and method for scheduling online keyword auctions subject to budget constraints.

20                                   **BACKGROUND OF THE INVENTION**

Most theoretical analysis of online keyword auction mechanisms has neglected the practical aspect of limited budgets for the buyers. Several publications describe on-line algorithms for conducting sponsored search auctions,  
25 sometimes with budget constraints. However, these approaches apply approximation algorithms that unfortunately are unable to predict or use forecast query data. As a result, various implementations of online keyword auctions

may only ensure that daily budget limits for buyers are not exceeded at the expense of negatively impacting the auctioneer's objective.

For instance, an implementation may use a throttling  
5 rate for budgeting. In this case, a buyer may only be permitted to participate in a percentage of auctions in which the buyer may actually wish to bid so that the buyer's daily spend may not exceed the buyer's daily budget. If the buyer's daily spend may in fact exceed the daily budget,  
10 then the buyer may become completely throttled and no longer participate in bidding for auctions that day. This may result in removing more and more buyers from auctions as the day progresses than may be necessary, considering spend and budget over the course of a day.

15 A different implementation including the highest bidders may be combined with throttling so that each buyer may continue to participate in each auction as long as a buyer's remaining daily budget may not be exceeded. However, such an implementation may also fail to provide the  
20 optimal objective for an auctioneer. At some point in the day, a buyer that may be able to bid on a variety of keyword auctions may actually spend the entire daily budget as the highest bidder on frequently occurring keywords, and thereby be removed as an available buyer for bidding on less  
25 frequently occurring keywords. Thus, this greedy approach may also result in removing more buyers from auctions as the

day progresses than may be necessary considering pricing and frequency of keywords over the course of a day.

What is needed is a system and method that may optimize the objective for an online auctioneer while ensuring that  
5 spending by buyers remains within their specified budget constraints. Such a system and method should be able to take into consideration sequencing of daily queries and budgeting by buyers throughout a day. Such a system and method should be able to support an auctioneer's objective  
10 to maximize revenue and/or to maximize overall "social" value of the auctioned keywords to the bidders.

#### SUMMARY OF THE INVENTION

Briefly, the present invention may provide a system and  
15 method for scheduling online keyword auctions subject to budget constraints. In various embodiments, a client having a web browser may be operably coupled to a query processing server for sending a query request. The query processing server may include a model generator for creating a linear  
20 programming model used to provide a candidate set of advertisements for keywords of query requests. The query processing server may also include an operably coupled linear programming analysis engine for optimizing the linear programming model offline to generate slates of  
25 advertisements for keywords of a query request and to generate a frequency for each slate to indicate how often

the slate of advertisements should be displayed. The query processing server may then choose a slate of advertisements online in accordance with the generated frequencies to provide a slate of advertisements accompanying search  
5 results of a query request to the web browser.

In an embodiment, the linear programming analysis engine may associate with each slate of advertisements an indicator of priority or value, and an expected traffic volume. In such an embodiment, the query processing server  
10 may choose a slate of advertisements online in accordance with the expected traffic priorities and values prescribed.

The query processing server may also be operably coupled to a database of advertisements that may include any type of advertisements that may be associated with an  
15 advertisement ID. In an embodiment, several bidders may be associated with an advertisement ID. The database of advertisements may also include a collection of advertisement slates that may be generated as part of the linear programming model. Each of the advertisement slates  
20 may represent an ordered candidate set of advertisements for keywords of a query request.

The present invention may provide a framework for predicting the volume and order in which queries may appear throughout the day for use in allocating bidders to auctions  
25 to optimize revenue of an auctioneer. A linear programming model of slates of advertisements may first be created

offline along with frequencies indicating how often each slate of advertisements should be displayed. Each slate of advertisements may represent an ordered candidate set of advertisements, where the ordering may be determined in whole or in part by the bids of the buyers according to the rules set by the auctioneer. To do so, a subset of queries and bidders may be selected; an estimate of the number of queries may be obtained for particular time-slots; a proportional budget may be calculated for each bidder for each time-slot; and ranked slates of advertisements may be determined for the subset of queries. Linear programming using column generation with the forecast keyword occurrences as a constraint and the bidders' budgets as a constraint may be applied to generate columns that may be added to the linear programming model of slates of advertisements in order to produce the optimal objective to an auctioneer. Upon receiving a query request, a slate of advertisements may be chosen online according to the previously generated frequencies, and the chosen slate of advertisements that may provide an optimal objective to the auctioneer may then be output for sending to a web browser for display.

Advantageously, the present invention may effectively use a forecast of the frequency and sequence of keywords occurring throughout a day for optimizing the objective of an auctioneer. By scheduling bidders to auctions, the

present invention may also provide improved coverage for multi-keyword bidders. Other advantages will become apparent from the following detailed description when taken in conjunction with the drawings, in which:

5

**BRIEF DESCRIPTION OF THE DRAWINGS**

FIG. 1 is a block diagram generally representing a computer system into which the present invention may be incorporated;

10        FIG. 2 is a block diagram generally representing an exemplary architecture of system components for scheduling online keyword auctions subject to budget constraints, in accordance with an aspect of the present invention;

15        FIG. 3 is a flowchart for generally representing the steps undertaken in one embodiment for scheduling online advertising auctions subject to budget constraints by applying linear programming using column generation, in accordance with an aspect of the present invention;

20        FIG. 4 is a flowchart for generally representing the steps undertaken in one embodiment for applying linear programming using column generation to determine a relative frequency for each slate to provide optimal revenue, in accordance with an aspect of the present invention;

25        FIG. 5 is a flowchart for generally representing the steps undertaken in one embodiment for determining one or more slates of advertisements that may improve the objective

by generating one or more columns of the linear programming model, in accordance with an aspect of the present invention;

FIG. 6 is a flowchart for generally representing the steps undertaken in one embodiment for responding to queries applying the results of a linear programming model of advertising auctions subject to budget constraints, in accordance with an aspect of the present invention;

FIG. 7 is a flowchart for generally representing the steps undertaken in one embodiment for determining throttle rates that may be used in scheduling online advertising auctions subject to budget constraints by applying linear programming using column generation, in accordance with an aspect of the present invention; and

FIG. 8 is a flowchart for generally representing the steps undertaken in one embodiment for responding to queries by applying throttle rates determined using the results of a linear programming model of advertising auctions subject to budget constraints, in accordance with an aspect of the present invention.

#### DETAILED DESCRIPTION

##### EXEMPLARY OPERATING ENVIRONMENT

FIG. 1 illustrates suitable components in an exemplary embodiment of a general purpose computing system. The exemplary embodiment is only one example of suitable



components and is not intended to suggest any limitation as to the scope of use or functionality of the invention.

Neither should the configuration of components be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary embodiment of a computer system. The invention may be operational with numerous other general purpose or special purpose computing system environments or configurations.

10       The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, and so forth, which perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in local and/or remote computer storage media including memory storage devices.

25       With reference to FIG. 1, an exemplary system for implementing the invention may include a general purpose computer system 100. Components of the computer system 100 may include, but are not limited to, a CPU or central processing unit 102, a system memory 104, and a system bus

120 that couples various system components including the system memory 104 to the processing unit 102. The system bus 120 may be any of several types of bus structures including a memory bus or memory controller, a peripheral  
5 bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA)  
10 local bus, and Peripheral Component Interconnect (PCI) bus . also known as Mezzanine bus.

The computer system 100 may include a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by the computer system  
15 100 and includes both volatile and nonvolatile media. For example, computer-readable media may include volatile and nonvolatile computer storage media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or  
20 other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or  
25 any other medium which can be used to store the desired information and which can accessed by the computer system

100. Communication media may include computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. For instance, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media.

The system memory 104 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 106 and random access memory (RAM) 110. A basic input/output system 108 (BIOS), containing the basic routines that help to transfer information between elements within computer system 100, such as during start-up, is typically stored in ROM 106. Additionally, RAM 110 may contain operating system 112, application programs 114, other executable code 116 and program data 118. RAM 110 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by CPU 102.

The computer system 100 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, FIG. 1 illustrates a hard disk drive 122 that reads from or writes to non-

removable, nonvolatile magnetic media, and storage device 134 that may be an optical disk drive or a magnetic disk drive that reads from or writes to a removable, a nonvolatile storage medium 144 such as an optical disk or magnetic disk. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary computer system 100 include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 122 and the storage device 134 may be typically connected to the system bus 120 through an interface such as storage interface 124.

The drives and their associated computer storage media, discussed above and illustrated in FIG. 1, provide storage of computer-readable instructions, executable code, data structures, program modules and other data for the computer system 100. In FIG. 1, for example, hard disk drive 122 is illustrated as storing operating system 112, application programs 114, other executable code 116 and program data 118. A user may enter commands and information into the computer system 100 through an input device 140 such as a keyboard and pointing device, commonly referred to as mouse, trackball or touch pad tablet, electronic digitizer, or a microphone. Other input devices may include a joystick, game pad, satellite dish, scanner, and so forth. These and

other input devices are often connected to CPU 102 through an input interface 130 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A display 138 or other type of video device may also be connected to the system bus 120 via an interface, such as a video interface 128. In addition, an output device 142, such as speakers or a printer, may be connected to the system bus 120 through an output interface 132 or the like computers.

The computer system 100 may operate in a networked environment using a network 136 to one or more remote computers, such as a remote computer 146. The remote computer 146 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer system 100. The network 136 depicted in FIG. 1 may include a local area network (LAN), a wide area network (WAN), or other type of network. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet. In a networked environment, executable code and application programs may be stored in the remote computer. By way of example, and not limitation, FIG. 1 illustrates remote executable code 148 as residing on remote computer 146. It will be appreciated that the network connections

shown are exemplary and other means of establishing a communications link between the computers may be used.

SCHEDULING ONLINE KEYWORD AUCTIONS SUBJECT TO BUDGET

5 CONSTRAINTS

The present invention is generally directed towards a system and method for scheduling online keyword auctions subject to budget constraints. A linear programming model of slates of advertisements may be created offline for  
10 predicting the frequency and sequence of keywords occurring throughout a day for use in online scheduling of bidders to auctions that may optimize revenue of an auctioneer. Each slate of advertisements may represent a candidate set of advertisements in order of optimal revenue to an auctioneer.  
15 Linear programming using column generation with the keyword as a constraint and a bidder's budget as a constraint may be applied to generate columns that may be added to the linear programming model of slates of advertisements in order to determine optimal revenue to an auctioneer. Upon receiving  
20 a query request, a slate of advertisements may be chosen online according to the generated frequencies, and the chosen slate of advertisements may then be output for sending to a web browser for display.

As will be seen, the framework described may also apply  
25 when the budget constraints for one or more bidders may require those bidders to be removed from a set of bidders.

In this case, subsequent bidders may be moved up the order in a slate of advertisements. As will be understood, the various block diagrams, flow charts and scenarios described herein are only examples, and there are many other scenarios  
5 to which the present invention will apply.

Turning to FIG. 2 of the drawings, there is shown a block diagram generally representing an exemplary architecture of system components for scheduling online keyword auctions subject to budget constraints. Those  
10 skilled in the art will appreciate that the functionality implemented within the blocks illustrated in the diagram may be implemented as separate components or the functionality of several or all of the blocks may be implemented within a single component. For example, the functionality for the  
15 client query handler 206 may be included in the same component as the web browser 204. Or the functionality of the model generator 216 may be implemented as a separate component on another server. Moreover, those skilled in the art will appreciate that the functionality implemented  
20 within the blocks illustrated in the diagram may be executed on a single computer or distributed across a plurality of computers for execution.

In various embodiments, a client computer 202 may be operably coupled to one or more servers 210 by a network  
25 208. The client computer 202 may be a computer such as computer system 100 of FIG. 1. The network 208 may be any

type of network such as a local area network (LAN), a wide area network (WAN), or other type of network. A web browser 204 may execute on the client computer 202 and may include functionality for receiving a search request which may be  
5 input by a user entering a query. The web browser 204 may be operably coupled to a client query handler 206 that may include functionality for receiving a query entered by a user and for sending a query request to a server to obtain a list of search results. In general, the web browser 204  
10 and the client query handler 206 may be any type of interpreted or executable software code such as a kernel component, an application program, a script, a linked library, an object with methods, and so forth.

The server 210 may be any type of computer system or  
15 computing device such as computer system 100 of FIG. 1. In general, the server 210 may provide services for query processing and may include services for providing a list of auctioned advertisements to accompany the search results of query processing. In particular, the server 210 may include  
20 a server query handler 212 for receiving and responding to query requests, a model generator 214 for creating a linear programming model used to provide a candidate set of advertisements for keywords of query requests, and a linear program analysis engine, or optimizer, 216 for choosing  
25 slates of advertisements for keywords of the queries expected for processing. Each of these modules may also be



any type of executable software code such as a kernel component, an application program, a linked library, an object with methods, or other type of executable software code.

5       The server 210 may be operably coupled to a database of advertisements such as ad store 218 that may include any type of advertisements 220 that may be associated with an ad ID 214. In an embodiment, several bidders 222 may be associated with an ad ID 214 for one or more advertisements  
10       220. The ad store 218 may also include a collection of ad slates that may be generated as part of the linear programming model, each ad slate representing an ordered candidate set of advertisements for keywords of a query request.

15       There are many applications which may use the present invention for scheduling online auctions subject to budget constraints. For example, online search advertising applications may use the present invention to schedule keyword auctions subject to bidders' budget constraints. Or  
20       online search advertising applications may use the present invention to schedule keyword auctions by expected revenue rather than by bid. For any of these applications, advertisement auctions may be scheduled that optimize the objective of the auctioneer.

25       FIG. 3 presents a flowchart for generally representing the steps undertaken in one embodiment for scheduling online

advertising auctions subject to budget constraints by applying linear programming using column generation. In an embodiment, consider  $M = \{\Theta, B\}$  to be an auction marketplace, where  $\Theta = \{q_1, q_2, \dots, q_N\}$  may be a set of possible queries and  $B = \{b_1, b_2, \dots, b_M\}$  may be a set of all bidders. Also consider the "bidding state" of the auction marketplace to be defined by a matrix  $A(t)$ , where  $A_{i,j}(t)$  may be the bid amount that the  $j$ -th bidder may be bidding on the  $i$ -th query  $q_i$ . Assuming a static bidding state ( $A(t)=A$ ), consider the daily budget limit that may be specified by each bidder  $b_j$  to be defined as  $d_j$ . Note that in an embodiment,  $d_j$  can represent the daily budget for a campaign, where a buyer may be associated with multiple campaigns. Thus  $d_j$  may represent a daily spend limit by a bidder (or campaign) across multiple queries. If a daily budget limit may not be specified for a bidder, then assume that  $d_j = \infty$ .

For each query  $q_i$ , consider  $R_{i,j}(t) = A_{i,j} \cdot w_{i,j}(t)$  to be the ranking function used to rank the  $j$ -th offer in an auction instance, where  $w_{i,j}(t)$  may be a time-dependent weighting factor for the  $i$ -th query and  $j$ -th bidder. The ranking function  $R_{i,j}(t)$  may be equal to zero for any bidder  $b_j$  that may not participate in an auction instance. Furthermore,  $w_{i,j}(t)$  may be separated into two components as follows:

$$w_{i,j}(t) = Q_{i,j}(t) \cdot B_{i,j}(t),$$

where  $Q_{i,j}(t)$  may be defined to be the quality component of  $w_{i,j}$  and  $B_{i,j}(t)$  may be defined to be the

budgeting component. For example, in an online auction  $Q_{i,j}(t) = 1$  may result in a bid ordering of advertisements, and  $Q_{i,j}(t) = \text{CTR}_{i,j}$ , where CTR represents the click-through rate, may result in a revenue ordering of the

5 advertisements. A linear programming model may be created for this defined marketplace as further described below.

At step 302, a set of queries bid upon by a set of bidders may be selected from the expected query set. For example, queries received for a previously occurring day may  
10 be selected and a set of bidders who have bid on those queries may be selected. At step 304, an estimate of the number of queries may be obtained for each time-slot in a time period. In an embodiment, there may be twenty-four hour-long timeslots defined for a 24 hour day. In various  
15 other embodiments, the timeslot may be fifteen minutes long, thirty minutes long, a period of a day and so forth.

Once an estimate of the number of queries may be obtained for each time-slot, a proportional budget for each bidder may be calculated for each time slot at step 306. In  
20 an embodiment, a proportional budget may be calculated for each bidder by scaling a known budget over a period of time, such as scaling a daily budget over a period of 24 hours, and over a set of queries. Furthermore, calculating a proportional budget may include adjusting a scaled amount by  
25 subtracting accrued spending for a given time period.

At step 308, ranked slates of ads may be determined for the subset of queries. For each query  $q_i$ , the "bidding landscape" may be defined in an embodiment as a set of bidder indices  $L_i = \{j_p : R_{i,j_p} > 0, p=1 \dots P_i\}$ , where the indices  $j_p$  may  
 5 be sorted by the value of  $R_{i,j}$  in descending order, and  $P_i$  may be the number of bidders in the landscape.

Furthermore, a slate of ads may be defined that may be a subset of the bidding landscape  $L_i$ . Each bidding landscape  $L_i$  may be mapped into a set of slates  $L_i^k$ , each being a unique  
 10 subset of  $L_i$  which can be obtained by deleting members of  $L_i$  while maintaining the ordering and then truncating, if necessary, to  $P_i^k$  members. More formally, The  $k^{th}$  slate for ad  $i$ , may include a unique subset (of length  $P_i^k$ ) of the indices of  $L_i$ , and may be defined as  $L_i^k = \{j_{k_l} : j_{k_l} \in L_i, 1 \leq P_i^k \leq P\}$ ,  
 15 where  $P$  may be the maximum number of slots available for advertising on a page, such as a web browser. The indices in  $L_i^k$  may be ordered as in  $L_i$ , such as in order of ranking  $R_{i,j}$ . In an embodiment, if there may be less than  $P+1$  members, an additional dummy member may be added to  $L_i^k$  for  
 20 the purpose of computing second-bid prices.

At step 310, the estimated click-through-rate may be determined for ad positions for keywords of each query. For a query  $q_i$ , consider  $T_{i,j}(p)$  to denote the expected click-

through-rate ("CTR") for a bidder  $j$  who may be ranked at slot  $p$  on a page.

In general, the data collected in steps 302 through 310 may be stored for use by the linear programming analysis engine to apply linear programming using column generation to determine the relative frequency for each slate to provide optimal revenue. At step 312, linear programming using column generation may be applied to determine the relative frequency for each slate to provide optimal revenue.

FIG 4. presents a flowchart for generally representing the steps undertaken in one embodiment for applying linear programming using column generation to determine a relative frequency for each slate to provide optimal revenue. In general, the data collected from steps 302 through 310 for the linear programming model may be read from storage by the linear programming module in order to formulate and build the model as follows. Consider  $x_{ik}$  to represent the number of times to show slate  $k$  for keyword  $i$  in a given time slot. Note that any choice for  $x_{ik}$  can be represented as particular values for  $B_{i,j}(t)$ . In general, values for  $x_{ik} \geq 0$  may be found, such that revenue for a time-slot may be maximized and spending may be less than the budget for all bidders  $j$ .

The expected revenue-per-search (rps) may be defined in a 2<sup>nd</sup> bid pricing model for a slate and a query as:

$$rps(L_i^k) = \sum_{l=1}^{P_i^k} T_{i,j_{k,l}}(l) \cdot A_{i,j_{k,l}} \cdot \frac{Q_{i,j_{k,l}}(l)}{Q_{i,j_{k,l}}(t)}$$

The total revenue for a time-slot, over all queries, may therefore be defined as:

$$Rev = \sum_k \sum_{i=1}^N rps(L_i^k) x_{ik}.$$

5 The daily spend for each bidder  $j$  may be represented

as  $Spend_j = \sum_k \sum_{i=1}^N \delta_j(i, k) \cdot x_{ik}$ , where  $\delta_j(i, k) = T_{i,j_{k,p}}(p) \cdot U_{i,j_{k,p}}$ , and  $U_{i,j_{k,p}}$  may be the price bidder  $j$  pays every time his ad may be clicked at position  $p$  for the  $k$ -th slate of  $q_i$ , defined as:

$$U_{i,j_{k,p}} = \begin{cases} A_{i,j_{k,p}} \cdot \frac{Q_{i,j_{k,p}}(t)}{Q_{i,j_{k,p}}(t)} & 0 < p < P_i^k \leq P \\ 0 & \text{otherwise} \end{cases}$$

10 Linear programming with column generation may be used to find the optimal allocation. For example, consider  $d_j$  to be the budget for bidder  $j$ ; consider  $v_i$  to be the expected number of occurrences of the  $i$ -th keyword in the time-slot; consider  $c_{ijk} = \delta_j(i, k)$  to be the expected cost to bidder  $j$  of  
 15 if slate  $k$  may be shown for keyword  $i$ ; and  $r_{ik} = rps(L_i^k)$  to be the expected revenue on slate  $k$  for keyword  $i$ . For a budget defined as  $\sum_i \sum_k c_{ijk} x_{ik} \leq d_j \quad \forall j = 1, \dots, M$ , and an inventory defined as  $\sum_k x_{ik} \leq v_i \quad \forall i = 1, \dots, N$ ; the relative frequency for each slate to provide optimal revenue  
 20 may be determined by maximizing  $\sum_i \sum_k r_{ik} x_{ik}$ .

Using a conventional column-generation approach (See "A Column Generation Approach for Combinatorial Auctions", Workshop on *Mathematics of the Internet: E-Auction and Markets* Institute for Mathematics and its Applications

5 (2001) by Brenda Dietrich and John J. Forrest), an initial subset of slates,  $L_i \in L_i^k$ , may be generated at step 402 and the corresponding linear program may be solved at step 404, and then columns may be generated as needed using dual values of a linear program at step 406. For instance,

10 consider  $\pi_j$  to be the marginal value for bidder  $j$ 's budget, more specifically, the simplex multipliers for the  $j^{th}$  constraint of  $\sum_i \sum_k c_{ijk} x_{ik} \leq d_j \quad \forall j = 1, \dots, M$ , and consider  $\gamma_i$  to be the marginal value for the  $i^{th}$  keyword, more specifically, the simplex multipliers for the  $i^{th}$

15 constraint of  $\sum_k x_{ik} \leq v_i \quad \forall i = 1, \dots, N$ , then a column corresponding to slate  $L_i^k$  (and hence to variable  $x_{ik}$ ) may be profitably introduced into the linear programming model if  $r_{ik} - \sum_{j \in L_i^k} c_{ijk} \pi_j - \gamma_i > 0$ .

Accordingly, for each keyword  $i$ ,  $r_{ik} - \sum_{j \in L_i^k} c_{ijk} \pi_j$  may be

20 maximized over the legal slates  $L_i^k$ . If a slate may be found such that  $r_{ik} - \sum_{j \in L_i^k} c_{ijk} \pi_j - \gamma_i > 0$ , the corresponding slate and its

variable may be introduced into the linear programming model. If no such slate exists for any  $i$ , then an optimal solution may have been obtained. Those skilled in the art will appreciate that in an alternate embodiment,  $\sum_{j \in L_i^k} c_{ijk} \pi_j - r_{ij}$

5 may be equivalently minimized over the legal slates  $L_i^k$ .

Rather than generate every slate  $L_i^k$  *a priori*, after an initial subset of slates,  $L_i \in L_i^k$ , may be generated, then columns may be generated as needed using the dual values  $\pi_j$  and  $\gamma_i$  of a linear program. For instance, considering the

10 coefficients  $r_{ik} = rps(L_i^k) = \sum_{l=1}^{P_i^k} T_{i,j_{kl}}(l) \cdot A_{i,j_{kl+1}} \cdot \frac{Q_{i,j_{kl+1}}(t)}{Q_{i,j_{kl}}(t)}$  and

$$c_{ijk,k} = T_{i,j_{kp}}(p) \cdot A_{i,j_{kp+1}} \cdot \frac{Q_{i,j_{kp+1}}(t)}{Q_{i,j_{kp}}(t)} \text{ in } r_{ik} - \sum_{j \in L_i^k} c_{ijk} \pi_j - \gamma_i > 0,$$

$$F_{ik}(\pi) = \sum_{l=1}^{P_i^k} T_{i,j_{kl}}(l) \cdot A_{i,j_{kl+1}} \cdot \frac{Q_{i,j_{kl+1}}(t)}{Q_{i,j_{kl}}(t)} \cdot (1 - \pi_{j_{kl}}) \text{ may be maximized for a given}$$

$\pi_j$  over slates  $L_i^k$ .

FIG. 5 presents a flowchart for generally representing  
 15 the steps undertaken in one embodiment for determining one or more slates of advertisements that may improve the objective by generating one or more columns of the linear programming model. A keyword may be obtained at step 502, for instance, from a query. At step 504, a subset of  $L_i$  may  
 20 be generated for a keyword  $i$  of a query, and it may be



determined at step 506 whether the subset may provide an improved solution. In an embodiment, the subset of  $L_i$  may provide an improved solution if upon evaluating

$$F_{ik}(\pi) = \sum_{l=1}^{r_i^*} T_{i,j_{k,l}}(l) \cdot A_{i,j_{k,l}} \cdot \frac{Q_{i,j_{k,l}}(t)}{Q_{i,j_k}(t)} \cdot (1 - \pi_{j_k}), \quad F_{ik}(\pi) > \gamma_i. \quad \text{If so, the}$$

5 column(s) corresponding to the subset of  $L_i$  may be added to the linear programming model at step 508. If the condition of  $F_{ik}(\pi) > \gamma_i$  may not be satisfied, then it may be determined at step 510 whether the keyword may be the last keyword included in the query. If not, then processing may continue  
10 at step 502.

If all keywords have been processed, then it may be determined at step 512 whether any improving slate may have been found, that is some  $L_i^*$  may be found for which  $F_{ik}(\pi) > \gamma_i$ . If so, the augmented linear program incorporating the  
15 additional columns generated at step 508 may be solved, and processing may continue to step 502. If it may be determined at step 512 that no improving slate may be found, then processing may be finished since there may not be found any new column satisfying the condition of  $F_{ik}(\pi) > \gamma_i$ , and the  
20 linear programming model may provide an optimal solution.

Thus, the present invention may reduce the columns of the linear programming model to a manageable size by using a subset of possible combinations of advertisements which can be shown for each keyword. Once created, advertisement

slates and frequencies may also be available for caching.

In other embodiments, the linear programming analysis engine may associate with each slate of advertisements an indicator of priority or value, and an expected traffic volume. In

5 such embodiments, the query processing server may choose a slate of advertisements online in accordance with the expected traffic priorities and values prescribed. Moreover, the framework described may also apply when the budget constraints for one or more bidders may require those  
10 bidders to be removed from a set of bidders. In this case, subsequent bidders may be moved up the order in a slate of advertisements.

FIG. 6 presents a flowchart for generally representing the steps undertaken in one embodiment for responding to  
15 queries applying the results of a linear programming model of advertising auctions subject to budget constraints. Note that FIG. 6 may present a flow chart for one embodiment of the process of serving the slates of ads generated by the linear programming model. The slates of advertisements  
20 generated by the linear programming model may be stored as in 226 for use by the ad server. A query having a keyword may be received at step 602, and slates of advertisements for the keyword may be found at step 604 along with their associated frequencies. A random, or pseudo-random number,  
25 may be generated at step 606 with the same distribution as the specified frequencies. And the generated random number

may be used at step 608 to select a slate of advertisements to show with the query results. The selected slate of advertisements may be served at step 610 for display with the query results.

5           In addition to a client-server application that may implement the steps described in conjunction with FIG. 6 for serving a slate of advertisements, the present invention may also be applied for optimizing throttling rates used in applications to remove advertisements of bidders from a  
10   slate of advertisements when the bidders have spent an expected amount of their budget. Advantageously, applying the present invention for optimizing throttling rates may provide an advertising allocation that may requires less online computing resources by a server than direct  
15   application of the linear programming model of the present invention. For example, in an embodiment of a direct application of the linear programming model of the present invention, a server may be configured to maintain access to a database of all the slates of advertisements generated in  
20   an optimal solution, as well as their frequencies for display. In various applications that may use throttling of bidders, the optimal slates of advertisements along with the frequency data may be post-processed to obtain "throttling rates" which may then be applied to the holding out of a  
25   bidder from the bidder landscape, either on a query independent basis or on a query-bidder pair basis.

FIG. 7 presents a flowchart for generally representing the steps undertaken in one embodiment for determining throttle rates that may be used in scheduling online advertising auctions subject to budget constraints by applying linear programming using column generation. At step 702, a model of slates of advertisements within bidders' budgets may be created using the method described in conjunction with FIGS. 3-5 above. At step 704, throttle rates may be determined using the model. To do so, the linear programming model of the present invention may be extended to take into account the expected prices paid by the bidders. For example, the linear programming model may provide a set of slates for each query, together with the number of times the  $k$ -th slate for query  $i$  may be shown, denoted  $x_{ik}$ . The  $i$ -th query may be assumed to appear a total of  $q_i$  times. Each slate may have a maximum number of bidders appearing, such as 12 in one embodiment, and a bidder  $j$  may be considered to have an expected pay-out cost of  $p_{ikj}$  for appearing in slate  $k$  for query  $i$ . Consider  $Q_j$  to be the set of queries on which bidder  $j$  has bid, and consider  $S_{ij}$  to be the set of slates for query  $i$  in which bidder  $j$  appears.

Then, in an embodiment, effective throttle rates may be derived by comparing the number of times a bidder appears in some slate, for which they were eligible, with the number of times that the queries occurred which were used to generate

the slates. Accordingly, an effective throttle rate for bidder  $j$  may then be defined by the following equation:

$$TR_j^0 = 1 - \frac{\sum_i \sum_{k \in S_{ij}} x_{ik}}{\sum_{i \in Q_j} q_i}, \text{ which may be zero if } j \text{ participates}$$

in all the slates for which  $j$  may be eligible, and 1.0 if  $j$  may appear in none of them because  $j$  has been completely throttled out. Considering the additional expected cost of participating in a slate, the average price that bidder  $j$  may pay per impression for participating in slates for query  $k$  may be defined as:

$$P_{ij} = \frac{\sum_{k \in S_{ij}} p_{ikj} x_{ik}}{\sum_{k \in S_{ij}} x_{ik}}, \text{ where the dependency of the } p_{ikj} \text{ on the}$$

other bidders for the term may be ignored.

The effective throttle rate may then be computed by comparing the expected total price paid with the maximum that could have been spent at the rate  $P_{ij}$  for the relevant queries by using the following equation:

$$TR_j = 1 - \frac{\sum_{i \in Q_j} \sum_{k \in S_{ij}} p_{ikj} x_{ik}}{\sum_{i \in Q_j} q_i P_{ij}}. \text{ Notice that this equation may be equivalent to } TR_j^0 = 1 - \frac{\sum_i \sum_{k \in S_{ij}} x_{ik}}{\sum_{i \in Q_j} q_i} \text{ when the } p_{ikj} \text{ may be all equal.}$$

In another embodiment, an effective throttle rate may be derived by applying the linear programming model to provide finer-grained throttling by query and bidder. To do so, the comparison of the number of times a bidder appears in some slate, for which they were eligible, with the number

of times that the queries occurred which were used to generate the slates, defined as  $TR_j^0 = 1 - \frac{\sum_i \sum_{k \in S_{ij}} x_{ik}}{\sum_{i \in Q_j} q_i}$ , may be applied to an individual query  $k$  by using the following equation:

$$5 \quad QBTR_{ij} = 1 - \frac{\sum_{k \in S_{ij}} x_{ik}}{q_i}.$$

These throttle rates for a query-bidder pair more closely approximate the linear programming model and should produce solution values intermediate between the effective throttle rate for all queries and the linear programming  
10 solution. Note that, in this case, it may no longer be necessary to weight by the expected cost. However, the data storage requirements for this approach may be greater, and it may require more computing resources to integrate throttle rates for a query-bidder pair in existing server  
15 implementations for advertising auctions.

Returning to FIG. 7, the throttle rates may be output at step 706. In an embodiment, the throttle rates output may be used by a server in responding to queries by applying throttle rates to remove advertisements in the bidding  
20 landscape for the query of bidders who may otherwise be expected to exceed their budget.

FIG. 8 presents a flowchart for generally representing the steps undertaken in one embodiment for responding to queries by applying throttle rates determined using the

results of a linear programming model of advertising auctions subject to budget constraints. A query having a keyword may be received at step 802, and a bidding landscape for the keyword may be found at step 804. In an embodiment, 5 the bidding landscape may have been generated using the method described in conjunction with FIG. 4. In another embodiment, the bidding landscape may have been generated using another method, such as a greedy technique of choosing the bids of advertisements based upon the highest bids for 10 the keyword. Throttle rates may then be applied at step 806 to remove advertisements of bidders from the bidding landscape. Next, a slate of advertisements may be generated from the remaining bidding landscape at step 808. And the slate of advertisements may then be served at step 810 for 15 display with the query results.

As can be seen from the foregoing detailed description, the present invention provides an improved system and method for scheduling online keyword auctions subject to budget constraints. Such a system and method may efficiently 20 schedule bidders to auctions to optimize revenue of an auctioneer. The system and method may also apply broadly to online search advertising applications and may be used, for example, to schedule keyword auctions by expected revenue rather than by bid. Moreover, the present invention may 25 also be applied for optimizing throttling rates used in applications to remove advertisements of bidders from a

slate of advertisements when the bidders may be expected to exceed their budget. As a result, the system and method provide significant advantages and benefits needed in contemporary computing and in online applications.

5           While the invention is susceptible to various modifications and alternative constructions, certain illustrated embodiments thereof are shown in the drawings and have been described above in detail. It should be understood, however, that there is no intention to limit the  
10 invention to the specific forms disclosed, but on the contrary, the intention is to cover all modifications, alternative constructions, and equivalents falling within the spirit and scope of the invention.



WHAT IS CLAIMED IS:

1. A computer system for scheduling online  
5 advertisements, comprising:  
a model generator for creating a linear programming  
model used to provide candidate sets of advertisements,  
based on the results of an online auction, for keywords of  
query requests;  
10 a linear programming analysis engine for determining an  
optimal set of the slates of advertisements for keywords of  
a query request being processed; and  
a query processing server operably coupled to the  
linear programming analysis engine for providing slates of  
15 auctioned advertisements accompanying search results of  
query processing.
2. The system of claim 1 further comprising an  
advertisement store operably coupled to the linear  
20 programming analysis engine for storing advertisements that  
may be presented as a slate of advertisements.
3. The system of claim 1 further comprising a query  
handler operably coupled to the query processing server for  
25 receiving and responding to query requests.

4. A computer-readable medium having computer-executable components comprising the system of claim 1.

5. A computer-implemented method for scheduling  
5 online auctions, comprising:  
receiving a query having the keyword;  
finding slates of advertisements for the keyword and  
frequencies for displaying each slate of advertisements;  
selecting a slate of advertisements for display with  
10 results of the query; and  
outputting the slate of advertisements for display with  
the results of the query.

6. The method of claim 5 further comprising creating  
15 the linear programming model of slates of advertisements,  
each slate representing a candidate set of advertisements in  
order determined in whole, or in part, by the bids of the  
advertisers on the keywords.

20 7. The method of claim 6 wherein creating the linear  
programming model of slates of advertisements comprises  
selecting a subset of queries and bidders.

8. The method of claim 6 wherein creating the linear  
25 programming model of slates of advertisements comprises

obtaining an estimate of the number of queries for each of a plurality of time-slots.

9. The method of claim 6 wherein creating the linear  
5 programming model of slates of advertisements comprises calculating a proportional budget for each bidder for each time-slot.

10. The method of claim 6 wherein creating the linear  
10 programming model of slates of advertisements comprises determining ranked slates of advertisements for the subset of queries.

11. The method of claim 6 wherein creating the linear  
15 programming model of slates of advertisements comprises estimating click through rates for advertisement positions in a slate of advertisements for the keyword of the query.

12. The method of claim 10 wherein determining ranked  
20 slates of advertisements for the subset of queries comprises determining a set of bidder indices that may be ranked in descending order using a ranking function with a weighting factor for the subset of queries and a set of bidders.

25 13. The method of claim 10 wherein determining ranked slates of advertisements for the subset of queries comprises

determining a number of slots available for advertising on a display page.

14. The method of claim 6 wherein creating the linear programming model of slates of advertisements comprises applying linear programming using the keyword counts as a constraint and bidders' budgets as a constraint to generate columns that may be added to a linear programming model.

10 15. The method of claim 14 wherein applying linear programming using the keyword counts as a constraint and the bidders' budgets as a constraint to generate the columns that may be added to the linear programming model of slates of advertisements comprises determining the expected cost to  
15 the bidder for showing a slate of advertisements for the keyword in a time-slot.

16. The method of claim 14 wherein applying linear programming using the keyword counts as constraints and the  
20 bidders' budgets as a constraint to generate the column that may be added to the linear programming model of slates of advertisements comprises determining the expected revenue to the auctioneer for showing a slate of advertisements for the keyword in a time-slot.

25

17. The method of claim 5 wherein outputting the slate of advertisements for display with the results of the query comprises including the slate of advertisements in a web page for display to a user.

5

18. A computer-readable medium having computer-executable instructions for performing the method of claim 5.

10 19. A computer system for scheduling online auctions, comprising:

means for generating a subset of a list of advertisements for a keyword of a query;

15 means for determining whether the subset may provide optimal revenue to an auctioneer;

means for adding to a linear programming model a column corresponding to the subset of the list of advertisements; and

20 means for outputting the subset of a list of advertisements for the keyword of the query.

20. The computer system of claim 19 wherein means for determining whether the subset may provide optimal revenue to the auctioneer comprises using dual values of a linear program model, the dual values represented by a marginal

25

value for a bidder's budget and a marginal value of the keyword.

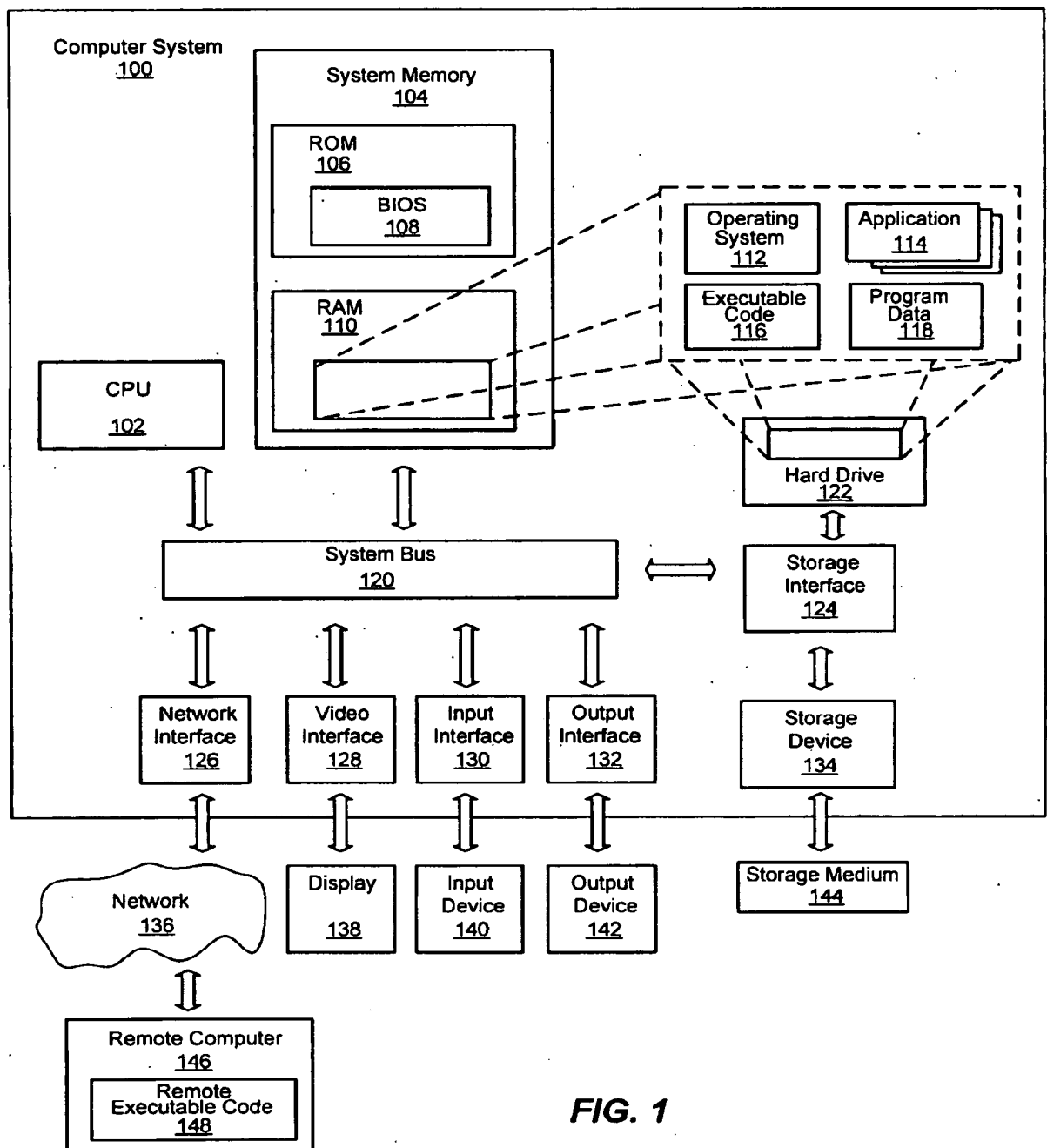
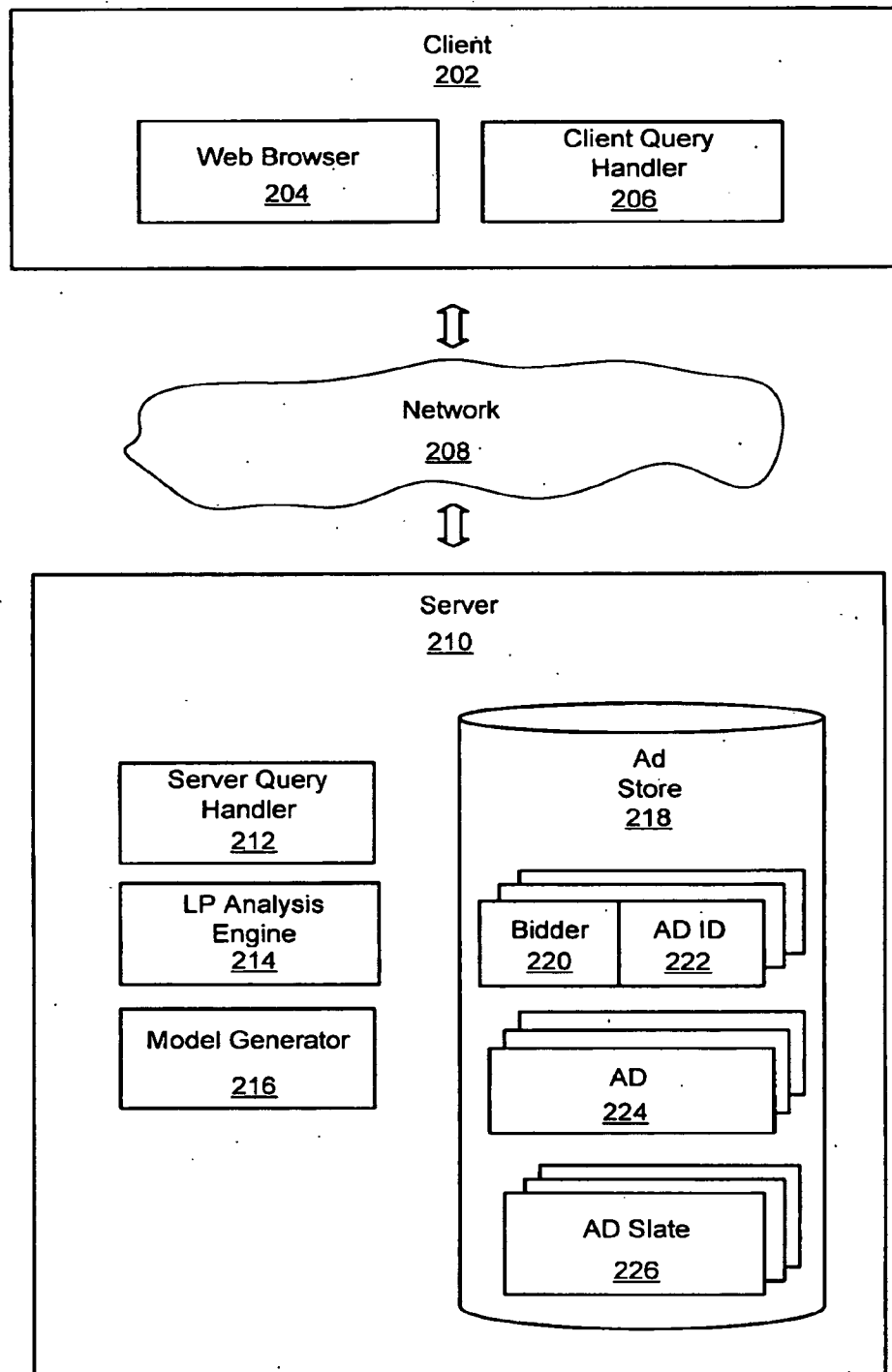
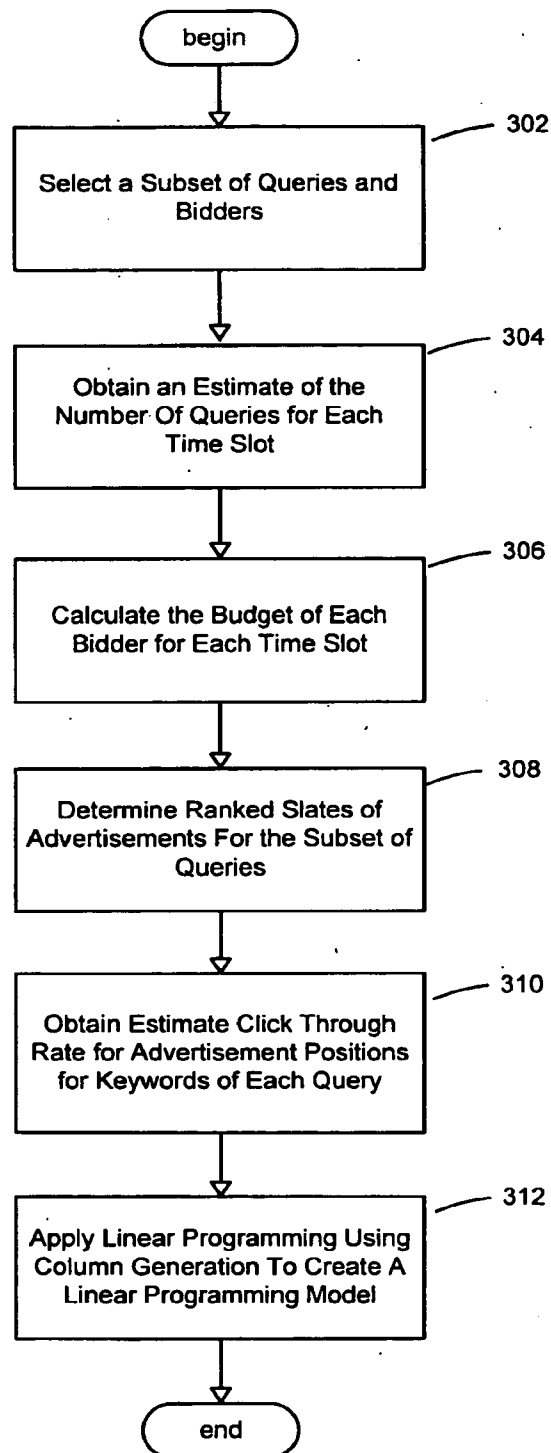
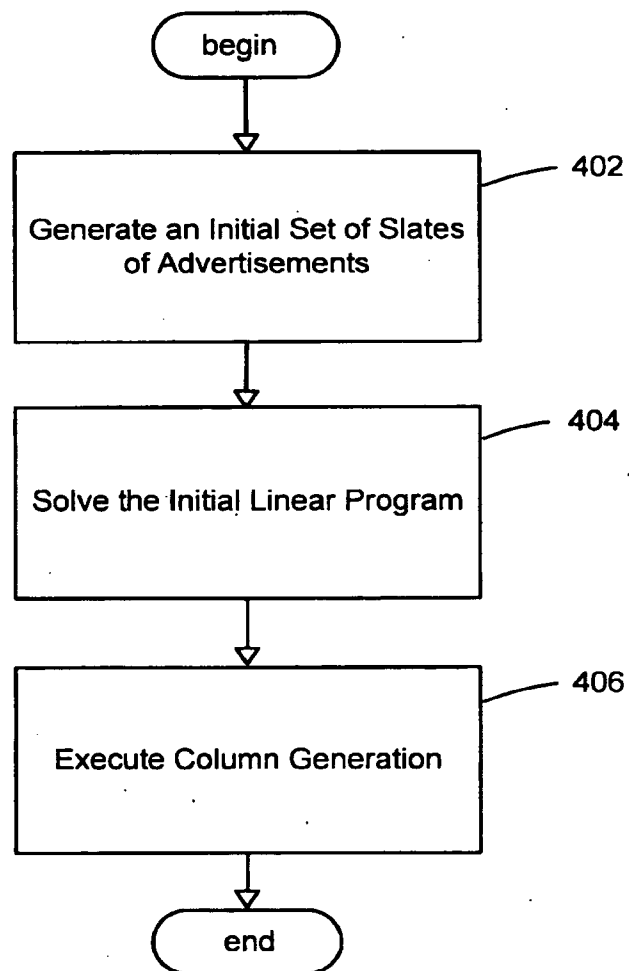


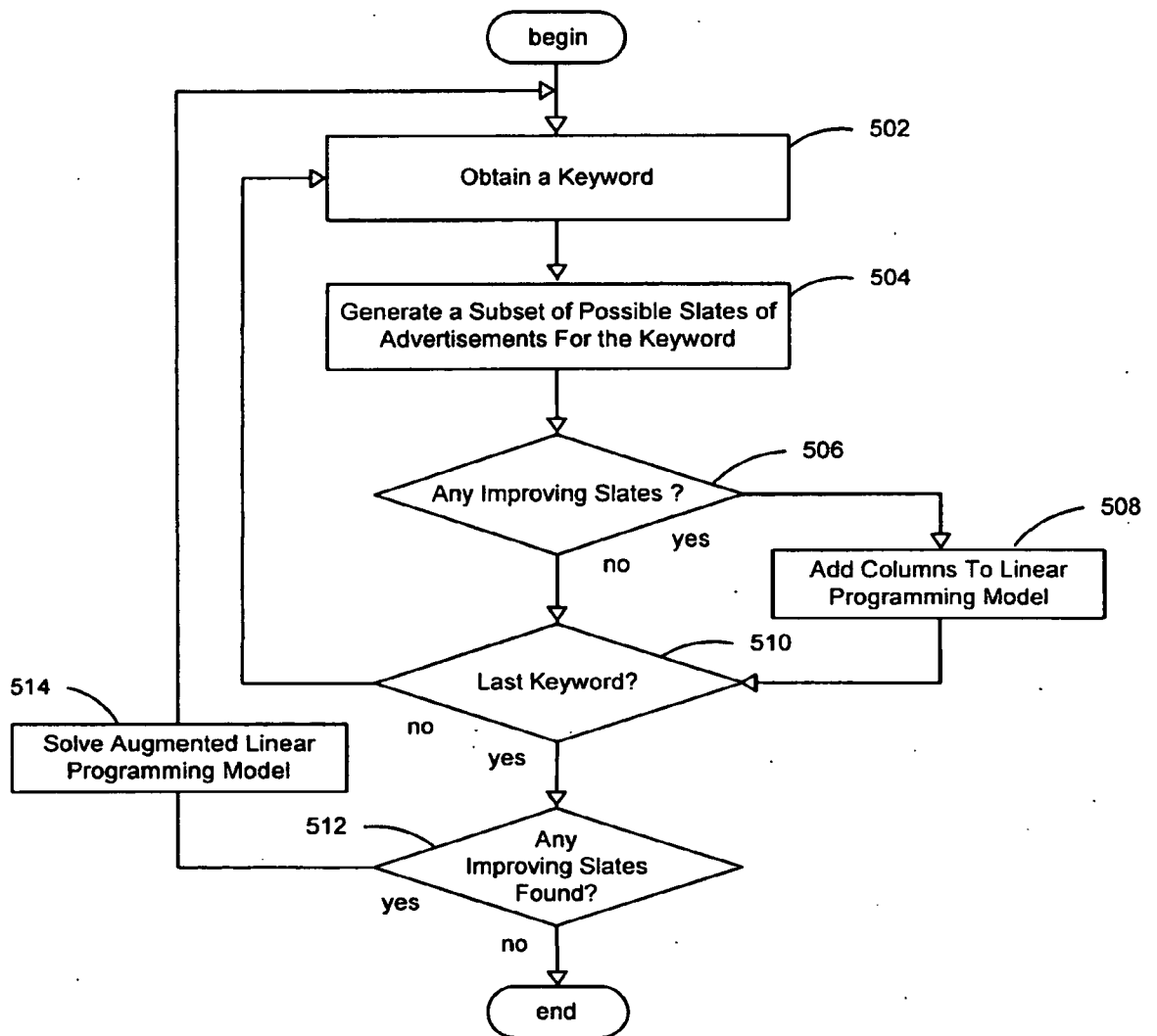
FIG. 1

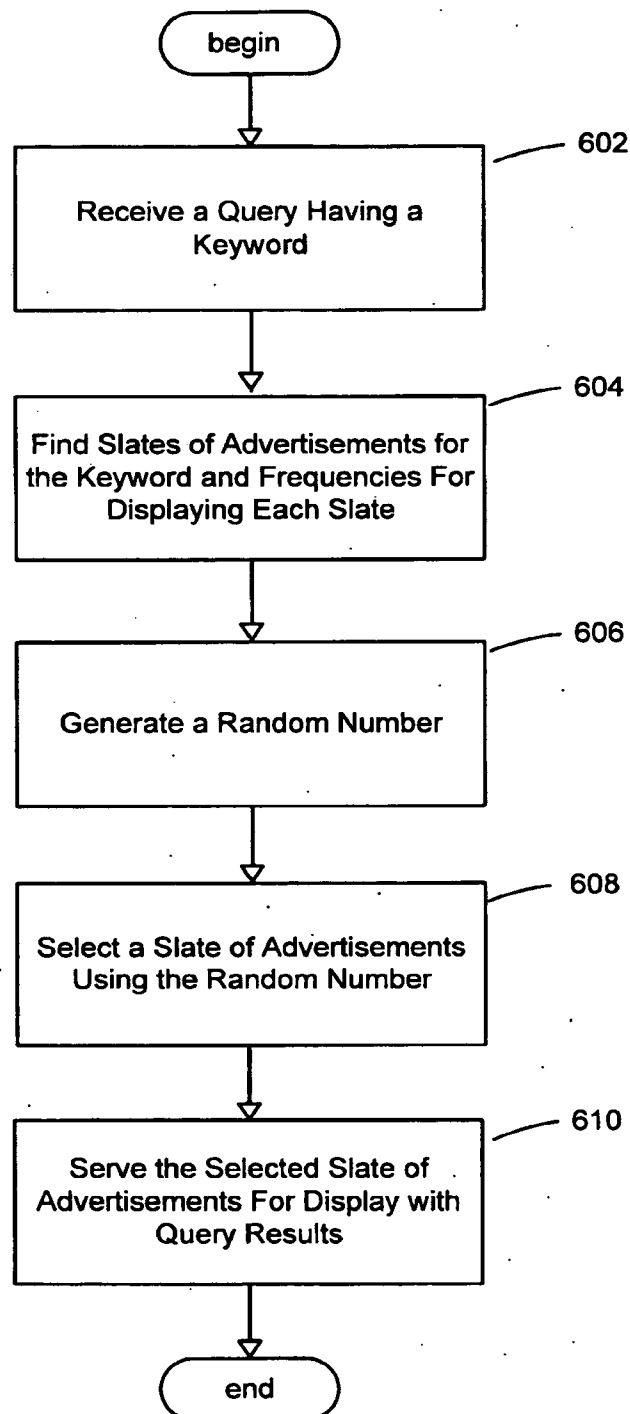
**FIG. 2**

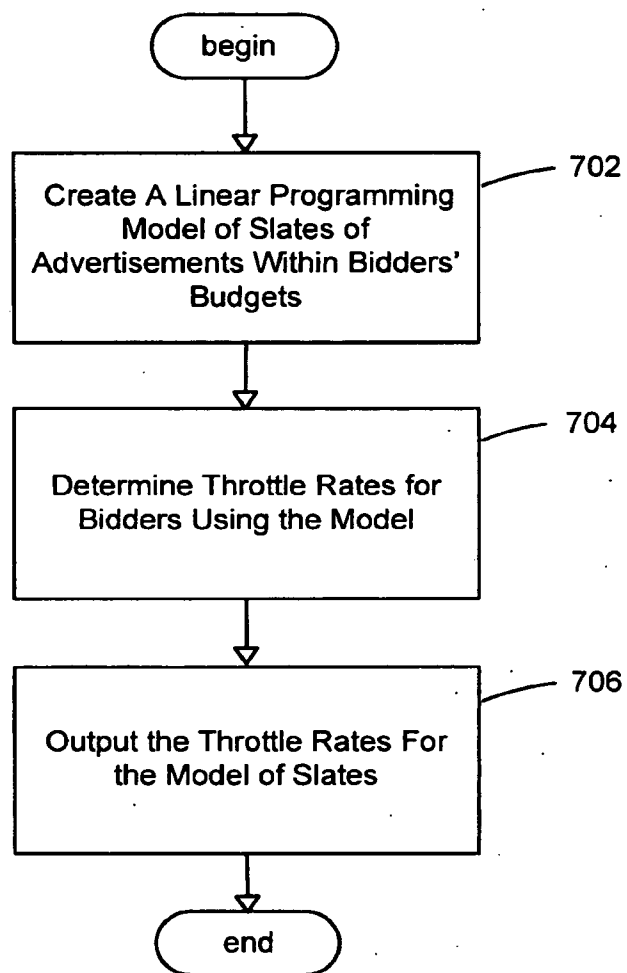


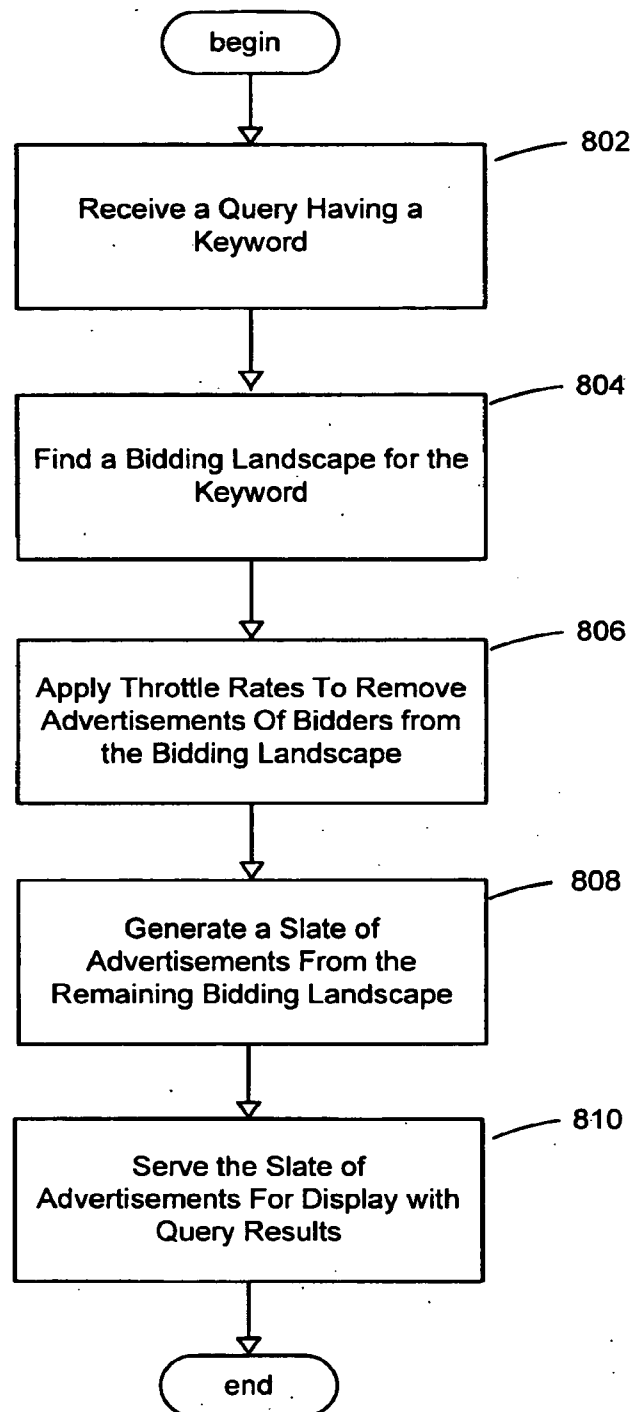
**FIG. 3**

**FIG. 4**

**FIG. 5**

**FIG. 6**

**FIG. 7**

**FIG. 8**