



[12] 发明专利申请公开说明书

[21] 申请号 03812522.6

[43] 公开日 2005年8月17日

[11] 公开号 CN 1656483A

[22] 申请日 2003.3.31 [21] 申请号 03812522.6
 [30] 优先权
 [32] 2002.4.1 [33] US [31] 60/369,460
 [86] 国际申请 PCT/US2003/009528 2003.3.31
 [87] 国际公布 WO2003/085561 英 2003.10.16
 [85] 进入国家阶段日期 2004.11.30
 [71] 申请人 奥弗图尔服务公司
 地址 美国加利福尼亚州
 [72] 发明人 斯科特·罗伊 纳林德·P·辛格

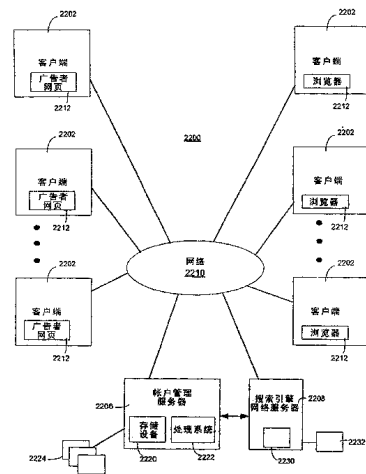
[74] 专利代理机构 北京东方亿思知识产权代理有限
 责任公司
 代理人 王 怡

权利要求书4页 说明书26页 附图22页

[54] 发明名称 与广告者支出成比例地显示付费搜索编目

[57] 摘要

在一种按位置付费的数据库搜索系统(2200)中,其中广告者(2202)付费以将他们的搜索编目包括在一个数据库中,该数据库响应于来自搜索者的查询而提供搜索结果(2212),广告者(2202)决定他/她在搜索词上花多少钱。搜索提供者(2208)与各个广告者所花费的钱数成比例地来显示广告者们的编目(2212)。这允许广告者向所述数据库搜索系统(2200)订购,决定在预定的时间段内为订购支付多少钱。搜索提供者(2208)可以为广告者(2202)推荐最佳开支量(2206)。



1. 一种用于数据库搜索系统的商业方法，在所述数据库搜索系统中搜索者向数据库提交搜索查询，并接收至少包括某些广告者资助的搜索编目
5 在内的搜索编目，所述商业方法包括：

以指定价格向各个广告者出售指定数量的搜索者约定；以及

接下来，响应于搜索查询并且与所述各个广告者的指定数量的搜索者约定成比例地，来提供各个广告者的搜索编目。

2. 如权利要求 1 所述的商业方法，其中，出售指定数量的搜索者约定的步骤包括：
10

出售所述各个广告者的搜索编目的搜索者印次。

3. 如权利要求 1 所述的商业方法，其中，出售指定数量的搜索者约定的步骤包括：

出售所述各个广告者的搜索编目的搜索者点进。

4. 如权利要求 1 所述的商业方法，其中，出售指定数量的搜索者约定的步骤包括：
15

出售在点进各个广告者的搜索编目后，在各个广告者的网站处的点进后搜索者动作。

5. 如权利要求 1 所述的商业方法，其中，出售指定数量的搜索者约定的步骤包括：
20

出售所述各个广告者的搜索编目中在预定时间段内的搜索者点进和搜索者印次之一。

6. 如权利要求 1 所述的商业方法，还包括：

接收搜索请求；

25 对于每一个单独的广告者，确定已收到的搜索者约定的各自的数量与
所述每一个单独的广告者应当收到的搜索者约定的各自的数量之间各自的
差异值；并且

根据所确定的差异值，返回搜索编目。

7. 如权利要求 1 所述的商业方法，其中，出售指定数量的搜索者约定

的步骤包括：

至少确定以下两者之一：

各个广告者的每次点击价钱，和
所述各个广告者的估计点击数量。

5 8. 如权利要求 7 所述的商业方法，还包括：

基于市场的历史数据来确定各个广告者的点进率。

9. 如权利要求 8 所述的商业方法，其中，确定点进率的步骤包括：

将所述点进率估计为所述各个广告者的搜索编目已获得的点击数与该
搜索编目已获得的印次数之比。

10 10. 如权利要求 1 所述的商业方法，还包括：

向各个广告者推荐最佳广告开支；以及
在不同的市场之间分配广告者广告开支。

11. 一种存储有计算机可读计算机代码的计算机可读存储介质，所述
代码被配置来实现一种用于数据库搜索系统的方法，在所述数据库搜索系
15 统中搜索者向数据库提交搜索查询，并接收至少包括某些广告者资助的搜
索编目在内的搜索编目，所述计算机可读计算机代码包括：

第一计算机可读代码，用于以指定价格向各个广告者提供指定数量的
搜索者约定，所述搜索者约定至少包括搜索者印次和搜索者点进之一；以
及

20 第二计算机可读代码，用于响应于随后收到的搜索查询并且与所述各
个广告者的指定数量的搜索者约定成比例地，来提供各个广告者的搜索编
目。

12. 如权利要求 11 所述的计算机可读存储介质，还包括：

第三程序代码，用于为各个广告者确定每次点进价钱；和

25 第四程序代码，用于确定可交付给所述各个广告者的点进的估计数。

13. 如权利要求 12 所述的计算机可读存储介质，其中，所述第三计算
机程序代码被配置来对所述各个广告者，将所述每次点进价钱确定为估计
总广告销售和估计点进率的函数。

14. 如权利要求 12 所述的计算机可读存储介质，还包括第五计算机程

序代码，其被配置为基于市场的历史数据来确定所述点进率和所述估计总广告销售。

5 15. 如权利要求 12 所述的计算机可读存储介质，其中所述第四计算机程序代码被配置来将所述的点进的估计数确定为所述各个广告者的广告开支对估计总广告销售之比的函数。

16. 如权利要求 11 所述的计算机可读存储介质，还包括：

第三程序代码，用于基于各个广告者的总广告预算、所述广告者的每次点进收益和外部回报率来优化所述广告者的广告开支。

17. 如权利要求 16 所述的计算机可读存储介质，还包括：

10 第四计算机程序代码，用于为所述各个广告者确定预算缩放因子。

18. 如权利要求 17 所述的计算机可读存储介质，其中所述第四计算机程序代码被配置为：

首先求解所有的零约束和流量约束；

15 确定将要响应于搜索查询而提供的搜索结果页上未被处于流量限的广告者所占据的空白空间的数量，以及当前不在零界限或流量界限上的广告者的数量；以及

求解方程系统，以确定所述各个广告者的预算约束。

20 19. 一种运行在计算机网络上的装置，用于响应于用户通过远程输入设备所输入的关键字而生成结果列表，所述装置包括与所述计算机网络相连的计算机系统，并且所述装置上存储有：

包含多个编目的数据库，其中每个编目与一个广告者相关联；

编程代码，用于响应于用户输入关键字而生成结果列表，所述结果列表包括具有与所述用户输入的所述关键字发生匹配的关联关键字的编目；和

25 用于管理广告者订购的编程代码，包括接受指定了用户约定数量的广告者订购订单、维护广告者订购帐户、以及当广告者编目包括在所生成的搜索列表中时调整各个广告者的订购帐户。

20. 如权利要求 19 所述的装置，其中，所述计算机系统响应于各个广告者提供的补偿来存储所述用户约定的数量。

21. 如权利要求 20 所述的装置，其中，所述计算机系统确定将作为对所述各个广告者提供的所述补偿的回报而提供的点进的估计数，所述的点进的估计数与所述各个广告者提供的所述补偿对所有广告者提供的总补偿之比有关。

5 22. 一种用于位置付费的数据库搜索系统的订购方法，所述方法包括：

以指定的价钱提供给广告者们指定数量的搜索者约定；

与一个或多个订购广告者建立订购帐户；

从搜索者处接收搜索请求；

10 响应于所述搜索请求，提供包括订购广告者的搜索编目在内的搜索结果；以及

响应于提供了所述搜索结果，调整所述订购广告者的所述订购帐户。

23. 如权利要求 22 所述的订购方法，还包括：

基于各个广告者的广告预算、每次点进收益和外部回报率中的至少一

15 项，来向所述各个广告者推荐最佳广告者支出量。

24. 如权利要求 23 所述的订购方法，还包括：

为所述各个广告者确定预算缩放因子，以限制所述广告者在多个市场中的开支。

与广告者支出成比例地显示付费搜索编目

- 5 本专利申请按美国 35 U.S.C § 119(e)要求享受以下专利申请的申请日，该申请是在 2002 年 4 月 1 日提交的美国临时专利申请 No. 60/369,460，该专利申请的内容作为参考全部包括在本说明书中。

背景技术

- 10 位置付费（pay-for-placement）搜索引擎就像电话黄页的在线版本一样运行：用户进行搜索，而系统显示与该用户的查询相匹配的付费广告者编目（listing）。因为屏幕的可用空间有限，所以用户一般只能看见与其查询相匹配的编目中的一小部分。因此，为了建立一个可行的系统，搜索引擎提供者需要决定每个编目多长时间显示一次，收取多少费用，以及各个
15 编目应当出现的顺序。

- 本领域的当前状态就是 Overture 的搜索引擎，可以从 www.overture.com 获得。Overture 使用一种被称为竞价排名（bid-for-rank）的方案，在该方案中，Overture 基于每个广告者愿意为每次点击支付多少费用，来按照点击和订购清单向各个广告者收费。广告者可以以每次点击 5 美分为最小出
20 价，报出他所希望的价格。许多网站显示 Overture 的搜索结果，由于每个网站都从列表的顶端开始显示不同数量的搜索结果，因此有一种很强的刺激要靠近顶端。出价最高的广告者总是出现，而随着其出价的降低，广告者出现的频率则越来越低。理论上，提供最高服务质量的广告者出价最高，并且出现在列表的顶端。而实际上，系统几乎不可能运行得这么好。
25 上述方案原理上很简单，但是它有很多问题使用户和广告者都受到阻碍。

 从用户的角度看，问题在于竞价排名方案可能导致不相关的、不想要的搜索编目靠近顶端。广告者完全控制着他们的编目出现的顺序，并且聪明的广告者可以利用这种自由带来的好处，让用户花钱而自己免费地显示。例如，假设有一个出售宠物雪貂的广告者。该广告者会在像“雪貂”

和“宠物雪貂”这样非常特定的搜索词的情况下出现，也会在像“宠物”这样更常见的词的情况下出现。当用户搜索“宠物雪貂”时，雪貂广告者出现在列表的顶端是合理的，因为所述用户几乎肯定是在找该广告者所出售的东西。但对于像“宠物”这样更常见的词而言，该雪貂广告者就不应该出现在靠近顶端的地方了。输入“宠物”的用户更有可能是找狗或猫，而不是雪貂。

不幸的是，如果雪貂广告者聪明的话，他将建立这样一种编目，就像“特别棒的宠物雪貂！超便宜！”。他可能愿意为这个编目出很高的价格——一个将他带到列表顶端的价格——因为他知道当用户点击该编目时，这个用户清楚地看到这是关于雪貂的。在该编目中有足够多的信息，使得广告者很有可能做成一笔买卖。在竞价排名系统中，广告者并不为处于列表顶端而付费；只有在用户实际点击他的编目时，广告者才付费。这个编目如此具体，以致于只有当用户对雪貂感兴趣的时候才会点击它，所以即使该编目在像“宠物”这样常见的词的情况下出现，对广告者也几乎没有风险。这样一来，针对“宠物”的第一个搜索结果是一个只和少数用户有关的编目。对于其它人而言，这个编目是没用的，并且整个搜索体验比较糟糕。雪貂现象在位置付费搜索引擎中比较普遍。而电话黄页就不会受到这个问题的困扰，因为广告者被迫基于他们所占的页面大小来付费。

从广告者的角度看，竞价排名系统的问题在于它比较复杂，广告者很难知道他获得了什么。当广告者为一个特定的排名出价时，他无法预先计算出他有可能得到多少次点击，或者有可能花掉多少钱，或者是否可能在某个其它排名上获得更高的收益。他甚至不能确定能否得到他想要的排名，因为另一个广告者可能在他之后给出一个更高的价格。如果广告者预算是固定的，那么他必须不断地监视他的支出，以确保不会超出预算，同时仍要保持他的出价足够地高，以获得最大可能数量的点击数。通常，他必须针对 50 或 100 个不同的搜索词来处理所有这些不确定因素，而这些搜索词中的每一个都有自己的出价。

例如，假设一个广告者现在为搜索词“鲜鱼”出价 1.00 美元，以获得

排名第 2 的位置。他可以保持他的位置，或将他的出价增加到 1.20 美元以前进到第 1 位，或将他的出价降低到 0.80 美元以下降到第 3 位。为了做出上述决定，他需要知道在以上三个排名的每一个位置上，他可能得到多少次点击。他几乎不可能得到这种信息。事实上，在本领域的当前状态下，

5 搜索引擎提供者甚至不能肯定地说：他在第 1 排名处将比在第 3 排名处获得更多的点击。即使广告者确定有必要出价到第 1 位，也不能保证在几个小时以后其它广告者中的一个不会出价超过他。或者，同样可能的是，广告者可能出价到第 1 位，接着几天之后发现他支付过多了，因为在他之后的广告者们已经放弃了。广告者必须不断地监视他的出价和位置，以确保

10 他获得了想要的东西，而不会支付过多。本领域的当前状态就是使用电子竞价代理来跟踪出价。例子如下：www.gotoast.com、www.did-it.com 和 www.pay-per-click-bid-managers.com。然而，这些例子的表现都不足够好，因为它们仅仅周期性地运行，而且它们通常不能从搜索引擎提供者那里获得所需要的信息，例如广告者在不同的排名处可能得到多少次点击这

15 样的信息。

如果广告者的预算是固定的，那么他还必须不断地监视他的支出，以确保他将来能够满足他的预算。例如假设广告者在下一个月有 1000 美元可供支出。如果他将其出价设定为 1.00 美元，并且第一天有 50 个用户点击他的编目，那么他必须压低他的出价，因为他的支出率太高了。按 50

20 美元/天算，他将在月底之前花掉他的全部预算。相反，如果第一天只有 10 个用户点击他的编目，那么他一定要抬高他的出价，因为他的支出率太低了。出价和预算之间的互动是复杂的，没有不断的调整很难合适。这也

将导致不好的搜索结果，这是因为用户看到的不是最佳的最相关广告者，而仅仅是当前还没有超出预算的那些广告者。

25 所有这些问题在广告者对多个搜索词竞价时变得更加复杂。每个词得到不同数量的搜索和点击，并且需要不同的出价。广告者必须按照某种最优化其总收益的方式，在各个搜索词之间分配他的资金。目前还没有很好的工具来帮助广告者完成这项工作，甚至最好的竞价代理也没有尝试在多个词之间升高或降低出价，以匹配固定预算。

许多广告者偏向于使用这种系统的一种替代。这种精巧的出价结构需要非常多的微观管理，并且它仅涉及广告者所实际关心的两个问题：他要付多少钱，以及他可以获得什么回报。广告者希望搜索引擎提供者告诉他：花了 1000 美元，他就可以在下一个月里买到 1000 次点击；或者他花两倍的价钱就可以获得两倍的点击，或者花一半的钱就获得一半的点击。这种模型与其它的广告方法更加一致，就像电话黄页和因特网横幅广告一样。当信息被提炼成花销和点击数时，很明显，整个出价和排名的概念就不再必要了。广告者并不真的关心他在搜索结果列表中的排名。他关心的是他出钱后可以获得多少次点击，并且这些点击中有多少将转化为销售。理想情况下，他可以仅仅基于这样的信息来作出购买决定——付多少钱，以及获得多少次点击作为回报——而将有关他的编目何时何地出现等所有其它细节问题留给搜索引擎提供者。基于这种想法的系统为搜索引擎提供者提供了充分的自由来决定响应于用户的查询应当显示哪些编目，因而解决了雪貂问题，同时对于广告者来说要简单得多。

15

发明内容

本发明在保持了竞价排名方案的最优特点的同时，消除了出价和排名：使用拍卖来自动地设定价格，并且允许广告者挑选他们应当出现的搜索词。另外，这些实施方式自动地优化了每个广告者在他所出现的搜索词之间的支出，因而不再需要竞价代理。

这些实施方式背后的想法是很简单的。每个广告者决定他想要在一个搜索词上花多少钱，并且搜索提供者与各个广告者所花费的钱数成比例地，来显示广告者们的编目。假设有 N 个广告者为显示 M 个编目的搜索结果页上的空间竞争。如果有 T 次搜索，则搜索提供者就有 TM 个总印次（impression）在广告者之间分配。一个印次就是呈现给搜索者的搜索结果其中的一个搜索编目的显示。如果广告者愿意花费的钱数是 a_i ，则他得到的印次数为：

$$IMPRESSIONS_i = TM \frac{a_i}{a_1 + \dots + a_N} \quad (\equiv t_i) \quad (1)$$

他可以预期的点击数为：

$$CLICK_i = t_i r_i \quad (= c_i) \quad (2)$$

r_i 项是广告者的点进率 (click-through rate)，即当广告者出现在搜索中时，用户点击他的编目的概率。

为了向每个人分配正确数量的印次，提供者保留每个广告者得到的印次数量的运行总数 s_i 。当用户进行搜索时，提供者计算这个数量和广告者应当已经得到的印次数量之间的差异。这个差异是：

$$DISCREPANCY_i = s_i - t_i \frac{S}{T}$$

S 是已经发生的搜索的总数。提供者按照所述差异来对广告者排序，并且返回顶端的 M 个编目。返回编目的顺序可以是随机的，或按照点进率 r_i 来排序的，或按照编目的感知质量来排序的，或者按照某种其它标准来排序的。这种算法确保了每个广告者在总共 T 次搜索后获得正确数量的印次，同时随时间平均地分配印次。

一个广告者可以购买的印次的数量是有限制的。由于在一种实施方式中，在任何搜索结果中，任何一个广告者都不可以出现一次以上，因此他具有的总印次不可能超过 T 。令 A 等于所有广告者花费的总钱数，则使一个广告者达到这种界限的钱数是：

$$a_i > \frac{A}{M} \quad (3)$$

我们将这个界限称为流量限。如果某个广告者超出了这个流量限，则搜索引擎提供者让多余的印次保持空白。如果 $N \leq M$ ，则始终至少有一个处于流量限的广告者，并且市场崩溃。在这种情形下，搜索引擎提供者可以减小它所返回的编目的数量，使得 $M < N$ ，并且广告者们有动力去竞争。或者，提供者可以总是将编目的数量设置为广告者数量的几分之一，例如 $M = 0.5N$ 。

在优选的实施方式中，广告者为点击或点进而不是印次来付费，并且价格在销售时是固定的。点击或点进是搜索者浏览广告者的搜索编目并点击其相关超级链接，或者选择浏览该搜索编目的动作。搜索者的网络浏览器随后被重新定向到与该搜索编目相关联的统一资源定位符 (URL)。当某个广告者想要购买点击时，搜索引擎提供者给出每次点击价钱 (cost-per-click)，该价钱取决于他预计用户多长时间点击一次所述广告者的编

目、以及他预计总共要卖出多少广告。提供者报出的每次点击价钱就是：

$$CPC_i = \frac{\hat{A}}{TM\hat{r}_i}$$

在这个方程中， \hat{A} 是搜索引擎提供者对他预计 T 次搜索所卖出的广告总数 A 的估值， \hat{r}_i 是他对所述广告者的点进率的估值。一个广告者的每次点击价钱随着他的预期点进率的下降而上升。如果每个人都有相同的点进率，
5 则每个人都支付相同的 CPC。搜索引擎提供者将他预计交付的点击数表示为：

$$ESTIMATED_CLICKS_i = TM\hat{r}_i \frac{a_i}{\hat{A}}$$

为了完成买卖，广告者决定他想要以所指示的每次点击价钱来花费的数目
10 a_i 。提供者按照这个价格来交付点击，直到广告者达到了他的开支界限，或者用户结束了总共 T 次搜索。

搜索引擎提供者通过调整其估值 \hat{A} 来控制价格。当他的估值尽可能准确时，即当 $\hat{A}=A$ 时，他赚得最多。如果估值 \hat{A} 太小的话，则点击估值过大，并且广告者在还没有达到各自界限的情况下就用光了 \hat{A} 。在这种情形
15 下，提供者应当通过增大 \hat{A} 来提升价格。广告者将趋向于花得更少，但是提供者得到的总数 \hat{A} 将会上升。如果估值 \hat{A} 太大，则点击估值过小，并且广告者在 T 次搜索结束之前就达到了总限 A 。在这种情形下，提供者应当通过降低 \hat{A} 来降低价格。广告者将趋向于花得更多，并且提供者得到的总数 A 将再次上升。如果估值 \hat{A} 刚好合适，则每个广告者支付其各自的开支
20 界限 a_i ，同时没有任何未售出的点击。同样地，当所有估计的点进率都尽可能准确时，搜索引擎提供者就做得最好。

对总开支和点进率做出良好估计的一种方法是使用历史数据。例如假设整个市场的历史点进率是 CTR。那么提供者可以使用以下公式来估计各个编目的点进率：

$$25 \quad r_i = \frac{u_i + kCTR}{s_i + k}$$

u_i 项是一个编目已得到的点击数， s_i 是印次数。对于一个新编目而言，这些数量都为 0，并且所述估值默认为总点进率 CTR。对于一个已得到很多

点击和印次的编目而言，所述估值接近于观测频率。常数 k 起到缩放因子的作用，用于确定所述估值在这两个值之间变化得有多快。当广告者决定为其编目购买更多的点击时，提供者基于 r_i 的最新估值而报出每次点击价钱。提供者也可以按照其它方式来估计点进率，例如通过比较新编目和有历史数据的其它编目。

提供者还有许多其它方式来完成与广告者签约的过程。例如，他可以每周或每月进行一次拍卖来确定数量 a_i ，或者他可以允许广告者不断地调整它们。本发明并不限于任何特定的方案。

显示搜索结果的每个网站都是不同的。在本领域的当前状态下，编目的数量在 1 到 20 之间变动，有些网站显示完整的编目，而其它网站只显示标题。由于这些差别，优选实施方式将搜索词和网站的每一种组合都视为一个独立的市场。一种替换方式是将一个搜索词从任何地方得到的所有印次都视为单独的一个市场。在这种情形下，流量限公式就不同于方程 3 了，但是其它方程和算法仍保持不变。本领域的普通技术人员可以毫无困难地推导出适于这种替代方式的新的流量限公式。

广告者一般要为很多不同的搜索词花钱。本发明的一个优点就是提供者可以推荐广告者应当在不同的市场之间所花的最佳数额并在其间自动分配他的钱。如果广告者遵循提供者的推荐，那么可以保证他最大化预期收益。这一部分的剩余内容将描述从单个市场中的单个广告者的情形开始，直到多个市场中的多个广告者的全部问题，所必要的公式和算法。

为了优化广告者的开支，提供者需要知道三件事情：广告者的总预算 b_i ，他的每次点击收益 p_i ，以及外部回报率 R （如果广告者们在其它地方花钱的话，该外部回报率 R 对所有的广告者都是可用的）。于是，广告者在单个市场中购买付费编目的预期收益就是：

$$25 \quad \text{PROFIT}_i = (p_i c_i - a_i) + R(b_i - a_i) \quad (\equiv f_i) \quad (4)$$

第一项是广告者从其付费编目中获得的收益；第二项是他投资于其它地方而获得的收益。如果外部回报率 R 比较高，则广告者应当在点击上少花预算。使其收益最大化的 a_i 值是：

$$a_i = -d_i + \sqrt{\frac{T}{1+R} v_i d_i} \quad (5)$$

量 d_i 是其它广告者在所述市场中花费的总钱数； $v_i = r_i p_i$ 是所述广告者的市场值。一个广告者应当花费的数额随着其点进率或每次点击收益的下降而减少。如果 a_i 小于 0，或者小于提供者的最小开支量，或者大于流量限，或者大于广告者的预算，则它被限制到适当的值。

- 5 当有多个市场时，最优解将单独地处理它们。广告者的最佳开支由方程 5 针对每个市场的一种特例给出。数量 T 、 v_i 和 d_i 在每一个市场中全都可以不同。市场之间仅有的交互因素就是广告者的预算界限，

$$\sum_j a_{ij} \leq b_i。$$

- 这是对所有的市场求和。为了实现这个界限，广告者的市场值包含了一个预算缩放因子， $v_{ij} = \lambda_i r_{ij} p_{ij}$ 。注意，利用单个 λ_i 值在每个市场中统一地缩放广告者的值。如果该广告者不受预算限制，则 $\lambda_i = 1$ 。如果他受到预算限制，则 λ_i 被设定为恰好花光其预算的那个值。任何一种好的数字方程求解器都可以找到这个值。例子包括在 www.ece.nwu.edu/OTC 上的非线性方程求解器的在线导读，以及 *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Dennis and Schnabel, ISBN 0898713641。
- 15

- 当在单个市场中有多个广告者时，问题就更复杂了。如果任何一个广告者改变他愿意支付的数额，则其他广告者的最优解都会改变。一种可能的算法是对每一个广告者进行叠代求解，并且希望叠代过程收敛。如果这些叠代过程真的收敛，则最终答案将是一个固定点，其中没有一个广告者可以增加其收益，除非另外一个广告者改变了他愿意支付的数额。使用方程 4 对 f_i 的定义，这个固定点满足以下方程系统：
- 20

$$\frac{\partial f_i}{\partial a_i} = 0 \quad i = 1 \dots N \quad (6)$$

- 该固定点不是对于任何广告者的全局最大值，这是因为如果一个广告者改变他支出的数额，则其他广告者的收益有可能上升。另外，由于相对于他可以控制的一个量——他自己的开支——来最大化每个广告者的收益，所以它是所述问题的正确解。
- 25

可以以封闭形式来求解方程 6 的系统。每个广告者的解是：

$$a_i = A \left(1 - \frac{v_L}{v_i} \right) \quad (7)$$

所有广告者加在一起花费的总钱数 A 是：

$$A = \frac{T}{1+R} v_L \quad (8)$$

量 v_L 是一个广告者在方程 7 变为负值并且该广告者不应花任何钱之前，可能有的最低市场值。它的值与所有广告者的值的调和平均有关，

$$5 \quad v_L = \bar{v} \left(1 - \frac{1}{N} \right) \quad (9)$$

其中所述调和平均值 \bar{v} 是，

$$\bar{v} = \frac{N}{\sum 1/v_i} \quad (10)$$

另一个感兴趣的量是 v_U ，即一个广告者在受到流量限制之前可能具有的最大值。它的值是，

$$10 \quad v_U = v_L / \left(1 - \frac{1}{M} \right) \quad (11)$$

方程 7-11 是这些实施方式的基本公式。

当使用方程 7 计算最佳开支量时，可能出现某些广告者具有的最佳量小于 0 或大于流量限的情况。这些解在现实世界中是不可能的。找到有效解的算法就在超出边界的广告者中挑选一个，将他约束到适当的界限，从
15 所述问题中除去他，并且对所有其它人重新计算解。这个过程不断重复，直到所述算法求出一个有效解。每次叠代中所重新计算的值对 v_L 使用略微不同的公式，

$$v_L = \bar{v} \left(1 - \frac{m}{Mn} \right) \quad (12)$$

在这个方程中， m 是搜索结果页上未被处于流量限的广告者占据的空白空
20 间的数量， n 是既不处于零界限也不处于流量限的自由广告者的数量。对自由广告者计算调和平均值 \bar{v} 。如果所有的广告者都是自由的，则 $m=M$ ， $n=N$ ，并且解等于先前方程 9 的公式。

用于挑选除去哪个广告者的算法要考虑或为负数或高于流量限的总的有符号超支：

$$25 \quad E = \sum e_i \quad e_i = \begin{cases} a_i & \text{如果 } a_i < 0 \\ a_i - A/M & \text{如果 } a_i > A/M \\ 0 & \text{其它} \end{cases} \quad (13)$$

如果 $E < 0$ ，则低于零界限的广告者的超支占优势，所述算法就除去具有最低市场值的广告者。如果 $E \geq 0$ ，则高于流量限的广告者的超支占优势，所述算法就将具有最高市场值的广告者限制到流量限。倾向于增加流量限来决定边界情形 $E=0$ ，以确保市场在广告者比搜索结果页上的空间还少的情况下也绝不会崩溃。

5 当有多个市场和多个广告者时，最优解将单独地处理它们。对于单个广告者，每个广告者的市场值包含了限制其在所有市场上的开支的预算缩放因子。如果广告者不受预算限制，则他的预算因子 $\lambda_i = 1$ 。如果广告者有预算限制，则 $\lambda_i < 1$ 。通过重新给广告者们编号使得一开始的 k 个广告者是受预算限制的，满足预算约束的算法将求解非线性方程的联立系统，

$$\sum a_{ij} = b_i \quad i = 1 \dots k \quad (14)$$

自由变量是处于预算界限的那些广告者的 λ_i 。通过方程 7 和 12 给出 a_{ij} 的值，被约束到界限 0 和 A_j/M_j 。任何好的方程求解器都可以数字求解这个系统。

15 有一些实际问题。首先，由于广告者的市场值随着其预算缩放因子的改变而改变，因此当我们求解方程 14 的系统时，他的最佳开支可能跨越零界限或流量限。因此很重要的一点就是一直使用方程 7 和 12 来计算 a_{ij} ，即使广告者一开始在某个市场中受到约束。通过一直使用所述方程，广告者的总最佳开支在 0 和其最大值之间平滑地变化。

20 其次，因为市场约束在求解 λ_i 时可能改变，所以算法必须进行叠代，直到得出稳定的解。所述算法是：先对每个市场中的所有零约束和流量约束求解，求出 m 和 n 的值，然后对预算约束求解，保持 m 和 n 值不变。如果在求解预算约束时，任何广告者跨越了零界限或流量限，就重复所述过程。

25 第三，所述算法在求解预算约束时必须判断哪些广告者包括在方程 14 中。该算法包括任何高于预算的广告者、以及任何 $\lambda_i < 1$ 的广告者。这种组合给出了处于其预算界限的广告者的工作集。这个工作集有可能是错的，其中有些广告者即使在 $\lambda_i = 1$ 时也实际上低于预算。在这种情形中，方程 14 在当前的工作集下没有解。好的方程求解器将尽可能地求解具有很多预算

约束的情况。在下次叠代中，所述算法将除去不属于所述工作集的广告者，并对方程 14 求解。

5 最后，存在这样的区域，其中广告者的总最佳开支并不随着他的预算比例改变而改变。当广告者在每一个市场中都受到约束时，就会发生这种情况。图 21 图示了该问题。这个图描绘了一个广告者的总最佳开支如何作为其预算比例的函数而变化。曲线图在左侧是平的，因为广告者的最佳开支在每一个市场中都低于零界限，并且它在右侧也是平的，这是因为他的最佳开支在每一个市场中都高于流量限。这些平坦区域可能造成数字方程求解器出现问题。

10 所述解决方案就是要提供两个点，即 $\min \lambda_i$ 和 $\max \lambda_i$ 作为方程求解器对 λ_i 的边界约束。这些边界取代了通常的界限 0 和 1。在单个市场中， $\min \lambda_i$ 是广告者到达零界限的点，而 $\max \lambda_i$ 是广告者到达流量限的点。这些值是：

$$\begin{aligned} \min \lambda_i &= \lambda_i \left(\frac{v_L - \beta}{v_i - \beta} \right) \\ \max \lambda_i &= \lambda_i \left(\frac{v_U - \beta}{v_i - \beta} \right) \end{aligned} \quad (15)$$

15 β 的值是：

$$\beta = \begin{cases} 0 & \text{如果广告者受到约束} \\ \bar{v}/n & \text{如果广告者是自由的} \end{cases}$$

20 所期望的边界约束是这些值在所有市场当中的最小值和最大值。有可能即使有了这些限制，方程求解器在开支曲线图的中部仍将遇到平坦区域。当广告者在一些市场中低于零界限，而在其余市场中大于流量限时，这些平坦区域几乎不会发生。在这种情形下，所述算法将以落在平坦区域以外的一个 λ_i 值来重新启动方程求解器，使其可以继续进行。

某个广告者可以放弃自动优化，而亲自调整他的开支。除了 v_L 的公式发生改变以外，上述所有方程继续对其它广告者成立。如果 C 是支付固定数额的广告者所贡献的总钱数，则

$$25 \quad v_L = \frac{1}{2}v_0 + \frac{1}{2}\sqrt{v_0^2 + \frac{4C(1+R)\bar{v}}{Tn}} \quad (16)$$

量 v_0 是从方程 12 得出的旧的零界限值，

$$v_0 = \bar{v} \left(1 - \frac{m}{Mn} \right)$$

注意，当 $C=0$ 时， v_L 成为 v_0 。另一个感兴趣的检验是 $n=1$ 和 $m=M$ 。这种情形下， $v_0=0$ ，并且方程 16 回到方程 5 的初始的那一个广告者结果。

- 5 在一种实施方式中，对于位置付费数据库搜索系统实施一种订购方法。该方法包括一个搜索提供者，该提供者以指定的价钱向广告者们提供大量的搜索者约定（engagement）。搜索者约定可以是印次、点击或点进、点进后动作，或者其它类型的约定。按照这种方式，广告者可以选择其愿意支付的数额或点击数，并由此订购。订购可以是针对一段固定的时
- 10 间，例如一个月，或者可以以任何双方协商的方式来安排。搜索提供者可以为不同的广告者，或者依据搜索编目的数量或者广告者所参与的市场，来提供不同的比率。

- 所述方法还包括与订购广告者建立订购帐户。可以向所述预定帐户收取广告者所支付的费用，并随后在发生搜索者约定时自动地从该帐户中扣除。
- 15

- 位置付费数据库搜索系统接下来将从搜索者那里收到搜索请求。响应于这些搜索请求，所述数据库搜索系统将提供搜索结果。如果搜索查询与搜索编目匹配的话，有些搜索结果可以是订购广告者的编目。如果以搜索编目页面提供了订购广告者搜索编目，则据此对订购帐户进行调整。可以
- 20 这样来完成上述工作，即通过记入帐户中订购广告者所支付的印次或点进的数量，并且为提供给搜索者的每一个搜索编目减去 1。也可以使用其它种类的订购管理。

附图说明

- 25 图 1-20 是示出了用于找到每个广告者的最佳开支的具体算法的流程图；

图 21 是示出了广告者的总最佳开支作为广告者的预算比例的函数而变化的曲线图；以及

图 22 是图示了包括计算机数据库搜索系统在内的网络的一种实施方

式的框图。

具体实施方式

5 现在参考附图，图 1-20 形成的流程图给出了用于找到每个广告者的最佳开支的具体算法。该算法的输入是：

	SEARCHES[M]	在一个市场 M 中的搜索数量
	SPACES[M]	一个搜索结果页上的空间数量
	ROI[M]	外部回报率
	PROFIT_PER_CLICK[A,M]	广告者的每次点击收益
10	CLICK_RATE[A,M]	广告者的点进率
	BUDGET[A]	广告者的预算。

该算法的输出是：

	SPEND[A,M]	广告者的最佳开支量
	CONSTRAINT[A,M]	广告者的约束状态
15	LAMBDA[A]	广告者的预算缩放因子。

20 在整篇详细的说明中，变量 A 都表示广告者，而变量 M 都表示市场。由于除 BUDGET[A]和 LAMBDA[A]之外的每一个量都取决于当前的市场 M，所以附图中一般都省略 M 以提高可读性。同样，附图中没有显示出错情况、浮点边界条件、或者用于缓存和优化的时机。本领域的普通技术人员将会毫无困难地理解伪代码，并且编写出可实现其的高效计算机程序。

25 这里所描述的实施方式可以实现为计算机可读程序代码，用于运行一台或多台处理设备以及相关的数据存储设备。在一种具体实施方式中，所公开的方法和装置可被实现为 C++程序代码，用于控制数据库管理系统或搜索引擎。在另一种实施方式中，所述方法和装置可以实现为存储了计算机可读程序代码的数据存储介质、执行这里所描述的功能的数据处理装置或者任何其它适当的设备。

图 1 示出了最顶层方法的一种实施方式。这种实施方式中的方法是一个循环过程：首先求解每个市场中的零约束和流量约束，然后求解预算约束。当函数 IS_SOLVED 指示当前的解满足所有约束时，所述循环过程结

束。在最后的步骤中，该算法在输出变量 SPEND[A,M]中记录每个广告者的最佳开支。

最顶层方法是一个标记为 Solve 的过程，其从框 100 开始。在框 102，调用 INITIALIZE_BUDGET-SCALES 过程。这个过程初始化所述广告者的预算缩放因子，即 lambda。这个过程的一种实施方式如图 9 所示。循环过程从框 104 开始。在框 106，初始化循环变量。在框 108，调用 SOLVE_MARKET 过程。这个过程的一种实施方式如图 2 所示。这个过程满足某个市场中的零约束和流量约束。循环在框 110 处继续，直到所有的市场 M 都被处理。在框 112，退出所述循环过程并调用 SOLVE_BUDGETS 过程。这个过程的一种实施方式如图 10 所示。这个过程满足每个广告者的预算约束。包括框 104、106、108、110、112 和 114 在内的所述循环过程继续处理，直到 IS_SOLVED 过程返回真值。这个过程的一种实施方式如图 17 所示。这一过程确定是否所有的零约束、流量约束和预算约束都被求解。如果没有，则控制返回框 104。如果是，则在框 116 调用 COMPUTE_SPENDING 过程。这个过程计算每个广告者的最佳开支。这一过程的一种实施方式如图 18 所示。然后所述 Solve 过程在框 118 结束。

图 2-8 示出了求解零约束和流量约束的步骤。输出是 CONSTRAINT [A,M]值，这些值指示了广告者在市场中是零约束的、流量约束的、还是不受约束的。最顶层循环位于图 2 中。该算法首先将当前市场中的所有约束初始化为 NONE，然后叠加约束，直到 MARKET_IS_SOLVED 指示当前解有效为止。在每次叠代中，所述算法调用 COMPUTE_MARKET_PARAMETERS，以计算表征市场的方程 7-12 中的量。

图 2 是图示了 SOLVE_MARKET 过程的一种实施方式的流程图。该过程从框 200 开始。在框 202 调用 INITIALIZE_CONSTRAINTS 过程。此过程的一种实施方式如图 3 所示。这个过程初始化市场中的广告者约束。接下来，在框 204 进入循环过程。在框 206 调用 COMPUTE_MARKET_PARAMETERS 过程。这一过程计算构成市场的各

个量。这个过程的一种实施方式如图 4 所示。还在框 206 调用 ADD_CONSTRAINT 过程。这一过程向市场加入最重要流量 (most significant traffic) 或零常数。这个过程的一种实施方式如图 6 所示。在框 208, 检验由 MARKET_IS_SOLVED 过程返回的值。这个过程的一种实施方式如图 8 所示。这个过程确定是否市场的所有零约束和流量约束都已被求解。如果这个过程没有返回真值, 则循环操作继续。否则, 该 SOLVE_MARKET 过程在框 210 结束。

图 3 示出了初始化市场约束的算法。该算法是一个将每个广告者的约束设置为 NONE 的循环。该 INITIALIZE_CONSTRAINTS 过程从框 300 开始。在框 302 进入循环过程。在框 304, 对于所考虑的每一个广告者, 将按广告者索引的 Constraint 数组的值初始化为 NONE 值。循环在框 306 继续, 直到所有的广告者都已被处理。该过程在框 308 结束。

图 4 示出了用于计算由方程 7-12 给出的各个市场参数的方法的一种实施方式。该方法从框 400 开始。在框 402, 调用 COMPUTE_SUMS 过程。COMPUTE_SUMS 计算初始量 FREE_SPACES、FREE_ADVERTISERS 和 Z, 其中 Z 是方程 10 中给出的调和平均值的分母。这些量将出现在剩余的公式中。所述方法使用由方程 12 给出的求 V_L 的公式, 但如果某些广告者手动调整其开支, 则也同样可以使用更一般的方程 16。这个过程 COMPUTE_SUMS 的一种实施方式如图 5 所示。在框 404 计算 V_{Bar} 的值, 而在框 406 计算 V_L 的值。在框 408, 定义变量 TOTAL_MARKET_SPEND 的值。该方法在框 410 处结束。

图 5 示出了用于计算 FREE_ADVERTISERS、FREE_SPACES 和 Z 的过程的一种实施方式。FREE_ADVERTISERS 对不受约束的广告者的数量计数; FREE_SPACES 对搜索结果页上未受流量约束的广告者占据的空白空间的数量计数; Z 是每个自由广告者的市场值的倒数之和。该过程使用对所有广告者的循环来计算这些量。

所述过程从框 500 开始。在框 503, 初始化 FREE_ADVERTISERS、FREE_SPACES 和 Z。在框 504, 使用广告者作为循环索引进入循环过程。在框 506, 确定对某个广告者的约束是否等于 NONE。如果是的, 则

在框 508 递增 FREE_ADVERTISERS 和 Z 的值。控制随后前进到框 514。否则，如果对该广告者的约束 (Constraint) 的值等于流量 (Traffic) (框 510)，则在框 512 递减 FREE_SPACES 的值。在框 514，循环操作继续，直到所有的广告者都已被处理。然后，该过程在框 516 结束。

5 图 6 示出了用于向 CONSTRAINT[A,M]值中增加最重要流量或零约束的算法。如果求解出市场了，则不做什么。否则，所述算法根据 EXCESS_SPEND 的符号来增加零约束或流量约束。如果超支是负数，则该算法为具有最小市场值的自由广告者增加零约束。如果超支是 0 或正数，则该算法为具有最大市场值的自由广告者增加流量约束。

10 所述过程从框 600 开始。在框 602，该过程通过调用 MARKET_SOLVED 过程来检验是否已求解出市场。这个过程的一种实施方式如图 8 所示。如果已求解出市场，则该过程结束，并且控制返回到调用进程。如果没有，则在框 604 调用 EXCESS_SPEND 过程。这个过程的一种实施方式如图 7 所示。如果 EXCESS_SPEND 返回的值小于 0，则在框 606，所述算法为具有最小市场值的自由广告者 A 增加零约束。如果 EXCESS_SPEND 返回的值是 0 或者正数，则在框 608，所述算法为具有最大市场值的自由广告者 A 增加流量约束。所述过程在框 610 处结束。

图 7 示出了用于计算由上面的方程 13 给出的总的有符号超支的算法。该算法加入来自其值小于 V_L 或大于 V_U 的每一个广告者的数额，
20 求出超支的运行总和。该算法忽略任何受到零约束或流量约束的广告者。

所述过程从框 700 开始。在框 702，变量 EXCESS_SPEND 的值被初始化为 0。在框 704，用广告者 A 作为循环索引来启动循环操作。在循环过程中，在框 706，确定当前索引的广告者的值是否小于 V_L ，即最小的可能自由值。如果是这样的，则在框 708，如果对广告者 A 没有任何约束，则按照图 7 中框 708 所示的值来递增变量 EXCESS_SPEND。然后控制前进到框 714。如果在框 710 处，当前索引的广告者的值大于 V_U ，即最大的可能自由值，则控制前进到框 712。在框 712 处，如果对广告者 A 没有任何约束，则按照图 7 中框 712 所示的值来递增变量 EXCESS_SPEND。然后控制前进到框 714。循环操作不断重复，直到所有
25

的广告者都已被处理。所述过程在框 716 结束。

图 8 示出了用于确定当前的 CONSTRAINT[A,M]值是否为有效解的算法。对于每一个自由广告者，该算法进行验证，以发现他的值是否小于最小的可能自由值 V_L ，或者大于最大的可能自由值 V_U 。只有当所有的自由广告者都落入这些边界内时，当前的解才是有效的。

所述过程从框 800 开始。在框 802，用广告者 A 作为索引来启动循环操作。在框 804，将广告者 A 的值与 V_L 进行比较。如果该值小于 V_L ，则在框 806，若对广告者 A 没有任何约束，则由该过程返回 NO 值。控制前进到框 812。如果在框 804 处广告者 A 的值不小于 V_L ，那么就在框 808 相对于 V_U 来检验这个值。如果它超过了 V_U ，则在框 810，若对广告者 A 没有任何约束，则该过程返回 NO 值。控制前进到框 812。如果还剩下其他的广告者，则循环操作在框 802 继续。如果所有的广告者都已被处理，而所述过程没有返回 NO 值，则控制退出循环，并且所述过程在框 814 返回 YES 值，指示了市场已被解出。所述过程在框 816 结束。

图 9 示出了用于初始化预算缩放因子的算法。该算法是一个将每个广告者的预算缩放因子设置为 1 的循环过程。该过程在 SOLVE 算法开始时被调用一次。

所述过程从框 900 开始。在框 902，使用广告者 A 作为循环索引而进入循环操作。在框 904，广告者的预算缩放因子 LAMBDA 被初始化为值 1。在框 906，所述循环操作继续，直到所有的广告者都已被处理。所述过程随后在框 908 处结束。

图 10-15 示出了用于求解预算缩放因子 LAMBDA[A]的步骤。图 10 中是最顶层的算法。该算法由计算处于预算界限的广告者的工作组开始。然后，该算法建立一个变量向量传递给方程求解器，其中所述工作组中的每个广告者对应于其中一个变量。每个变量都具有相关的上边界和下边界。SET_TO_ZERO 函数是一个外部方程求解器，其调整 LAMBDA[A]值，使得每个广告者的总开支刚好和他的预算匹配。任何一种方程求解器，只要它能够求解带有边界约束的非线性方程系统，就可以用在这里。方程求解器的输入是向量目标函数 BUDGET_ERROR、变量的数量 N、待调整的变

量向量、以及边界约束。当函数运行时，SET_TO_ZERO 调用 BUDGET_ERROR(I)来评估目标函数的第 I 分量。如果它找到了解，则它以对每个 I，BUDGET_ERROR(I)都等于 0 来结束。如果没有解，则它以处于上边界的 LAMBDA[A]值中的一个或多个值来结束，并且所述算法在下次叠代时将从所述工作组中除去对应的广告者。

图 10 图示了用于求解预算缩放因子的 SOLVE_BUDGETS 过程的一种实施方式。该过程从框 1000 开始。在框 1002，变量 N 被初始化为 0，并且变量 WORKING_SET 被设置等于 BUDGET_LIMITED_ADVERTISERS 过程的结果。这一过程的一种实施方式如图 12 所示。在框 1004，使用广告者 A 作为循环索引而开始循环操作。在框 1006，变量 N 递增，向量 VARIABLES 中按变量 N 索引的条目被设置等于向量 LAMBDA 中用于当前广告者的条目的标号 (reference)。调用 MIN_LAMBDA 过程，以确定变量 LOWER_BOUND 的值。MIN_LAMBDA 计算当前被索引的广告者在到处受到零约束之前的最小 lambda。这一过程的一种实施方式如图 13 所示。然后使用 MAX_LAMBDA 过程来确定变量 UPPER_BOUND 的值。MAX_LAMBDA 计算当前被索引的广告者在到处受到流量约束之前的最大 lambda。这一过程的一种实施方式如图 14 所示。在框 1008，该循环操作继续，直到所有的广告者 A 都已被处理。

在完成了所述循环操作后，在框 1010，如上所述调用 SET_TO_ZERO 过程来求解由输入 (BUDGET_ERROR 函数) 所定义的方程系统。这一函数的一种实施方式如图 11 所示。SOLVE_BUDGETS 过程在框 1012 结束。

图 11 示出了用于计算所述工作组中超过或低于预算的广告者的数量的过程。该过程从工作组中取出第 I 个广告者，并且返回他当前的总开支和他的预算界限之间的差。

过程 BUDGET_ERROR 从框 1100 开始。在框 1102，取出第 I 个广告者。在框 1104，计算由 TOTAL_SPEND 过程返回的结果和广告者 A 的 BUDGET_LIMIT 之间的差，将其作为 BUDGET_ERROR 的值。TOTAL_SPEND 过程的一种实施方式如图 16 所示。所述过程在框 1106 结

束。

图 12 示出了用于计算处于其预算界限的广告者的工作组的过程的一种实施方式。所述工作组包括超出其预算、或者具有小于最大可能值的 LAMBDA[A]的所有这些广告者。该算法将满足这两个预算界限条件之一的每一个广告者都添加到工作组中。

所述过程从框 1200 开始。在框 1202 处，初始化变量 BUDGET_LIMITED_ADVERTISERS。在框 1204 使用广告者 A 作为循环索引而进入循环操作。在框 1206，确定 TOTAL_SPEND 过程对于广告者 A 的当前值是否超过了广告者 A 的预算。TOTAL_SPEND 过程的一种实施方式如图 16 所示。如果在框 1206 检验的情况为真，则在框 1208，用广告者 A 的预算来递增变量 BUDGET_LIMITED_ADVERTISERS 的值。如果不是真，则在框 1210，确定当前被索引的广告者的 lambda 值是否小于 MAX_LAMBDA 对于广告者 A 的当前值。MAX_LAMBDA 过程的一种实施方式如图 14 所示。如果是小于，则在框 1212，用广告者 A 的预算来递增变量 BUDGET_LIMITED_ADVERTISERS 的值。否则，控制前进到框 1214。

在框 1214，如果还有其它广告者，则控制返回到框 1204。否则，如果所有的广告者都已被处理，则所述过程在框 1216 结束。

图 13 示出了用于计算一个广告者其最佳开支在每个市场中变为负数之前可能具有的最小 LAMBDA[A]的过程的一种实施方式。对于每个市场，该算法使用方程 15 来计算广告者的最小 LAMBDA[A]。在所有市场当中的最小值是在任何一个市场内的最小值。

该过程从框 1300 开始。在框 1302，变量 MIN_LAMBDA 的值被初始化为 1。在框 1304 使用市场 M 作为循环索引而进入循环操作。在框 1306、1308，实施上面的方程 15 以确定所述广告者的最小 lambda。在框 1310，如果还有市场未被处理，则控制返回框 1304。否则，该过程在框 1312 结束。

图 14 示出了一个用于计算一个广告者其最佳开支在每个市场中达到流量限之前可能具有的最大 LAMBDA[A]的过程。对于每个市场，该算法

使用方程 15 来计算广告者的最大 $LAMBDA[A]$ 。在所有市场当中的最大值是在任何一个市场内的最大值。

所述过程从框 1400 开始。在框 1402，变量 MAX_LAMBDA 被初始化为 0。在框 1404 使用市场 M 作为循环索引而开始循环操作。在框 1406、
5 1408，实施上面的方程 15 以确定所述广告者的最大 $lambda$ 。在框 1410 循环继续，直到所有的市场都已被处理。所述过程在框 1412 结束。

图 15 示出了用于确定是否有任何违反预算约束的过程 $BUDGETS_ARE_SOLVED$ 的一种实施方式。该算法进行检查以确认每一个广告者在没有超出其预算的情况下都具有 $LAMBDA[A]$ 的最大可能值。
10 只有当所有的广告者都满足这个条件的时候，该过程才返回 YES。

所述过程从框 1502 开始。从框 1502 开始对每个广告者 A 进行循环。在框 1504，将过程 $TOTAL_SPEND$ 对广告者 A 返回的结果与广告者 A 的预算进行比较。过程 $TOTAL_SPEND$ 的一种实施方式如图 16 所示。如果所述结果大于预算，则所述过程在框 1506 返回 NO 值。否则，在框
15 1508，如果过程 $TOTAL_SPEND$ 返回的结果小于广告者的预算，则在框 1510，若所述广告者的 $lambda$ 小于 MAX_LAMBDA 过程对于该广告者返回的值，那么所述过程将返回 NO 值。 MAX_LAMBDA 过程的一种实施方式如图 14 所示。循环在框 1512 处继续，直到所有的广告者都已被处理。如果在整个循环的叠代过程中，没有一个广告者返回了 NO 值，则在框
20 1514 返回 YES 值，并且过程在框 1516 处结束。

图 16 示出了用于计算广告者对所有市场的总最佳开支的过程 $TOTAL_SPEND$ 的一种实施方式。它使用 $OPTIMAL_SPEND(A,M)$ 来计算每个市场中的数额，并将各个结果累加到运行总值中。

所述过程从框 1600 开始。在框 1602，变量 $TOTAL_SPEND$ 的值被初始化为 0。在包括框 1604、1606、1608 在内的循环中，对于每个市场 M ，
25 用 $OPTIMAL_SPEND$ 过程所返回的值来递增变量 $TOTAL_SPEND$ 的值。这一过程的一种实施方式如图 19 所示。在处理了所有的市场后，该过程在框 1610 处结束。

图 17 示出了用于确定当前的 $CONSTRAINT[A,M]$ 和 $LAMBDA[A]$ 值

是否满足所有的零约束、流量约束和预算约束的过程 IS_SOLVED 的一种实施方式。该算法首先进行检查以确认每一个市场都满足其零约束和流量约束；然后它进行检查以确认每一个广告者都满足其预算约束。只有当所有这些条件都为真时，该算法才返回 YES。

- 5 所述过程从框 1700 开始。在框 1702，使用市场 M 作为循环变量而进入循环过程。在框 1704，检验由过程 MARKET_IS_SOLVED 返回的值。过程 MARKET_IS_SOLVED 的一种实施方式如图 8 所示。如果该过程返回负值，则过程 IS_SOLVED 返回 NO 值。否则，在框 1706，循环继续检验另一个市场 M。一旦所有的市场都已被检验，则检验由
- 10 BUDGETS_ARE_SOLVED 过程返回的值。BUDGETS_ARE_SOLVED 过程的一种实施方式如图 15 所示。在框 1708，如果该过程没有返回正值，则过程 IS_SOLVED 返回 NO 值。否则，在框 1710，所述过程返回 YES 值。所述过程在框 1712 结束。

- 图 18 示出了用于记录每个广告者在每个市场中的最终最佳开支量的
- 15 过程 COMPUTE_SPENDING 的一种实施方式。该算法是一个将每个 SPEND[A,M]值设置为 OPTIMAL_SPEND(A,M)的当前值的循环。

- 该过程从框 1800 开始。在框 1802，使用广告者 A 作为循环变量进入外循环。在框 1804，使用市场 M 作为循环变量进入内循环。在框 1806，将数组 SPEND 中的条目设置为由过程 OPTIMAL_SPEND 返回的当前值。
- 20 这一过程的一种实施方式如图 19 所示。在对广告者的一个值 A 处理了所有的市场 M 后，作为外循环的循环变量的广告者的值加 1。在处理了所有的广告者后，所述过程在框 1812 结束。

- 图 19 示出了使用方程 7 来计算单个市场中的广告者最佳开支的
- 25 OPTIMAL_SPEND 过程的一种实施方式。如果所述值小于 0 或大于流量限，则该算法将其约束到适当的值。

该过程从框 1900 开始。在框 1902，基于所述广告者在一个市场中的 TOTAL_SPEND 来确定变量 OPTIMAL_SPEND 的值。用于这一操作的 TOTAL_SPEND 过程的一种实施方式如图 16 所示。在框 1904，OPTIMAL_SPEND 的结果被设置为 OPTIMAL_SPEND 和 0 中的较大者以

及 OPTIMAL_SPEND 和 TRAFFIC_LIMIT 中的最小者。该过程在框 1906 处结束。

图 20 示出了用于计算广告者的市场值的过程的一种实施方式。该过程从框 2000 开始。所述值 VALUE 是广告者的预算缩放因子 lambda、其
5 每次点击收益以及其点进率之积。每次点击收益及其点进率可以从任何方便的来源处获得。

图 22 的框图因此示出了一个分布式系统 2200，其包括多个客户端计算机 2202、多个广告者网络服务器 2204、帐户管理服务器 2206 和搜索引擎网络服务器 2208，它们全都和网络 2210 相连接。网络 2210 此后通常称
10 为因特网。虽然所公开的系统和方法对于因特网特别有用，但是应当理解，客户端计算机 2202、广告者网络服务器 2204、帐户管理服务器 2206 和搜索引擎网络服务器 2208 可以通过多种不同类型网络之一而连接在一起。这些网络可以包括局域网（LAN）、其它广域网（WAN）以及通过电话线访问的区域性网络，例如商业信息服务。客户端和服务器进程甚至
15 可以包括同时运行在一台计算机上的不同程序。

客户端计算机 2202 可以是传统的个人计算机（PC）、工作站或者任何其它规模的计算机系统。每个客户端 2202 一般都包括一个或多个处理器、存储器、输入/输出设备和网络接口（例如传统的调制解调器）。广告者网络服务器 2204、帐户管理服务器 2206 和搜索引擎网络服务器 2208 可
20 以被类似地配置。但是，广告者网络服务器 2204、帐户管理服务器 2206 和搜索引擎网络服务器 2208 中的每一个都可以包括通过隔离的私有网络相连的多台计算机。事实上，网络 2210 可以包括成百上千的单独的计算机网络。

客户端计算机 2202 可以执行网络浏览器程序 2212，例如
25 NAVIGATOR、EXPLORER 或者 MOSAIC 浏览器程序，以定位存储在广告者服务器 2204 上的网络页面或记录 2214。浏览器程序 2212 允许用户输入所要取回的特定网络页面 2214 的地址。这些地址被称为统一资源定位符或 URL。另外，一旦取回了页面，浏览器程序 2212 就可以在用户“点击”了到其它网络页面的超级链接后提供对其它页面或记录的访问。这些

超级链接位于网络页面 2214 内，并为用户输入另一个页面的 URL 并取回该页面提供了自动化的方式。所述页面可以是数据记录，包括内容纯文本信息，或者更复杂的数字编码的多媒体内容，例如软件程序、图形、音频信号、视频等等。

- 5 在一种实施方式中，客户端计算机 2202 使用由超文本传输协议（HTTP）提供的功能，通过网络 2210 与各个网络信息提供者进行通信，所述提供者包括帐户管理服务器 2206、搜索引擎服务器 2208 和广告者服务器 2204，不过也可以使用其它的通信协议，例如 FTP、SNMP、TELNET 和本领域内公知的多种其它协议。优选地，搜索引擎服务器
- 10 2208、帐户管理服务器 2206 和广告者服务器 2204 位于万维网上。

 如上所讨论的，在各实施方式中至少设想了两类服务器。所设想的第一服务器是包括计算机存储介质 2220 和处理系统 2222 在内的帐户管理服务器 2206。数据库 2224 存储在帐户管理服务器 2206 的存储介质 2220 上。数据库 2224 包含广告者帐户信息，在一种实施方式中包括广告者订

15 购帐户信息。根据这里的说明将会理解到，这里所公开的系统和方法可以以作为可执行指令存储在帐户管理服务器 2206 的计算机存储介质上的软件来实现，所述计算机存储介质例如是存储器或大容量存储设备。运行在客户端计算机 2202 上的传统浏览器程序 2212 可被用来访问存储在帐户管理服务器 2206 上的广告者帐户信息。优选地，通过防火墙来实现对帐户

20 管理服务器 2206 的访问，所述防火墙并未示出，其用于防止帐户管理和搜索结果放置程序以及帐户信息受到外部的篡改。通过对标准通信协议的增强，例如 Secure HTTP（安全 HTTP）或 Secure Sockets Layer（安全套接口层），可以进一步提供安全性。

 所设想的第二服务器类型是搜索引擎网络服务器 2208。搜索引擎程序

25 允许网络用户或搜索者在导航到搜索引擎网络服务器 URL 或者能够通过其浏览器程序 2212 向搜索引擎网络服务器 2208 提交查询的其它网络服务器的网址后，输入关键字查询以在可从万维网获得的数以百万的页面中识别感兴趣的那些页面。在一种实施方式中，搜索引擎网络服务器 2208 生成搜索结果列表，该列表至少部分地包括由帐户管理服务器 2206 所引导

的竞价过程的结果得出并被其格式化的相关条目。搜索引擎网络服务器 2208 生成超文本链接的列表，这些链接所指向的文档中包含了与用户在客户端计算机 2202 处输入的搜索词相关的信息。所述搜索引擎网络服务器以网页的形式将这个列表发送给网络用户，然后显示在运行于客户端计算机 2202 上的浏览器 2212 上。搜索引擎网络服务器的一种实施方式可以通过导航到在以下 URL 处的网页而找到，即 <http://www.overture.com>。

搜索引擎网络服务器 2208 与因特网 2210 相连。在一种实施方式中，搜索引擎网络服务器 2208 包括由搜索编目记录组成的搜索数据库 2230，所述搜索编目记录被用来响应于用户查询而生成搜索结果。另外，搜索引擎网络服务器 2208 还可以与帐户管理服务器 2206 相连。帐户管理服务器 2206 还可以与因特网相连。搜索引擎网络服务器 2208 和帐户管理服务器 2206 用于解决位于客户端计算机 2202 处的用户的不同信息需要。

例如，位于客户端计算机 2202 处的一种用户可以是网络信息提供者，例如广告网站推销者、或者具有位于广告者网络服务器 2204 上的广告者网页 2214 的广告者。这些广告网站推销者或者广告者可能希望访问驻留在帐户管理服务器 2206 上的存储设备 2220 中的帐户信息。广告网站推销者可以通过驻留在帐户管理服务器 2206 上的帐户，参与到与其它广告者之间的竞争性竞价过程中去。一个广告者可以对与该广告者的网站内容相关的任意数量的搜索词出价。在一种实施方式中，可以使用在帐户管理服务器 2206 的处理器 2222 处执行的计算机程序来评估搜索列表中被出价的搜索词和相应的网站之间的相关性，其中所述计算机程序将根据一套预定义的编辑规则来评估搜索词和相应的网站。

当使用广告者所出价的搜索词进行搜索时，出价越高，在搜索引擎 2208 所生成的搜索结果列表页上所处的位置就越有利。在一种实施方式中，广告者所出价的数额包括每次经由搜索结果列表页上的超级链接对该广告者的网站进行访问时，从该广告者的帐户中扣除的钱数。在另一种实施方式中，每次广告者的搜索编目响应于搜索查询被提供或显示给搜索者，就从该广告者的订购帐户扣除预定的数额。搜索者利用计算机输入设备“点击”超级链接，以发起一个取回与广告者的超级链接相关联的信息

的取回请求。优选地，对搜索结果列表超级链接的每次访问或“点击”都将被重定向到搜索引擎网络服务器 2208，以将该“点击”与广告者的帐户标识符关联起来。这个不为搜索者所知的重定向动作在使用搜索者所点击的搜索结果列表超级链接来访问广告者的 URL 前，将访问编码到搜索结果页面中的帐户标识信息。在另一种实施方式中，可以是由对广告者 URL 的这种点进操作来导致从广告者的订购帐户中扣除预定的数额。所述帐户标识信息与来自取回请求的信息一起作为取回请求事件被记录在广告者的帐户中。由于通过这种机制获得的信息最终以一种使用本领域内公知的传统服务器系统日志所不可能的方式将帐户标识符与 URL 匹配起来，因此将会保留下准确的帐户债务记录。在某些实施方式中，搜索结果列表页上的广告者的网站描述和超级链接还伴随有关于广告者的编目是付费编目的指示。

客户端计算机 2202 处的第二类用户可以包括在网络上寻找特定信息的搜索者。这些搜索者可以通过其浏览器 2212 来访问驻留于网络服务器 2208 上的搜索引擎网页 2232。搜索引擎网页 2232 包括查询框，搜索者可以在其中键入包括一个或多个关键字的搜索词。或者，搜索者可以通过查询框来查询搜索引擎网络服务器 2208，该查询框被超级链接到搜索引擎网络服务器 2208，并位于存储在远程网络服务器处的网页上。当搜索者结束输入搜索词后，搜索者可以通过点击所提供的超级链接，向搜索引擎网络服务器 2208 发送查询。然后，搜索引擎网络服务器 2208 将生成一个搜索结果列表页，并将该页面发送给客户端计算机 2202 处的搜索者。

搜索者可以点击与搜索结果列表页上的每个编目相关联的超文本链接，来访问相应的网页。所述超文本链接可以访问因特网上任何地方的网页，并且包括到位于广告者网络服务器 2204 上的广告者网页 2214 的付费编目。在一种实施方式中，搜索结果列表中还包括一些非付费编目，这些编目并不是广告者竞价的结果，而是由传统的万维网搜索引擎生成的，例如 INKTOMI、LYCOS 或 YAHOO! 搜索引擎。这些非付费的超文本链接还可以包括由编辑组手工编入数据库 2230 索引的链接。

从以上描述中可以发现，所给出的实施方式提供了与广告者所花费的

数额成比例地来显示广告者的编目的方法和装置。每个广告者决定他想在一个搜索词上花多少钱，并且搜索提供者据此显示广告者的编目。

虽然示出并描述了本发明的特定实施方式，但是可以做出多种修改。因此想要在所附的权利要求中覆盖这些变化和修改，其在本发明的真正精神5 和范围内。

因此想要把以上这些详细的描述视为示例性的而非限制性的，应当理解，是想以包括所有等同物在内的所附权利要求来限定本发明的精神和范围。

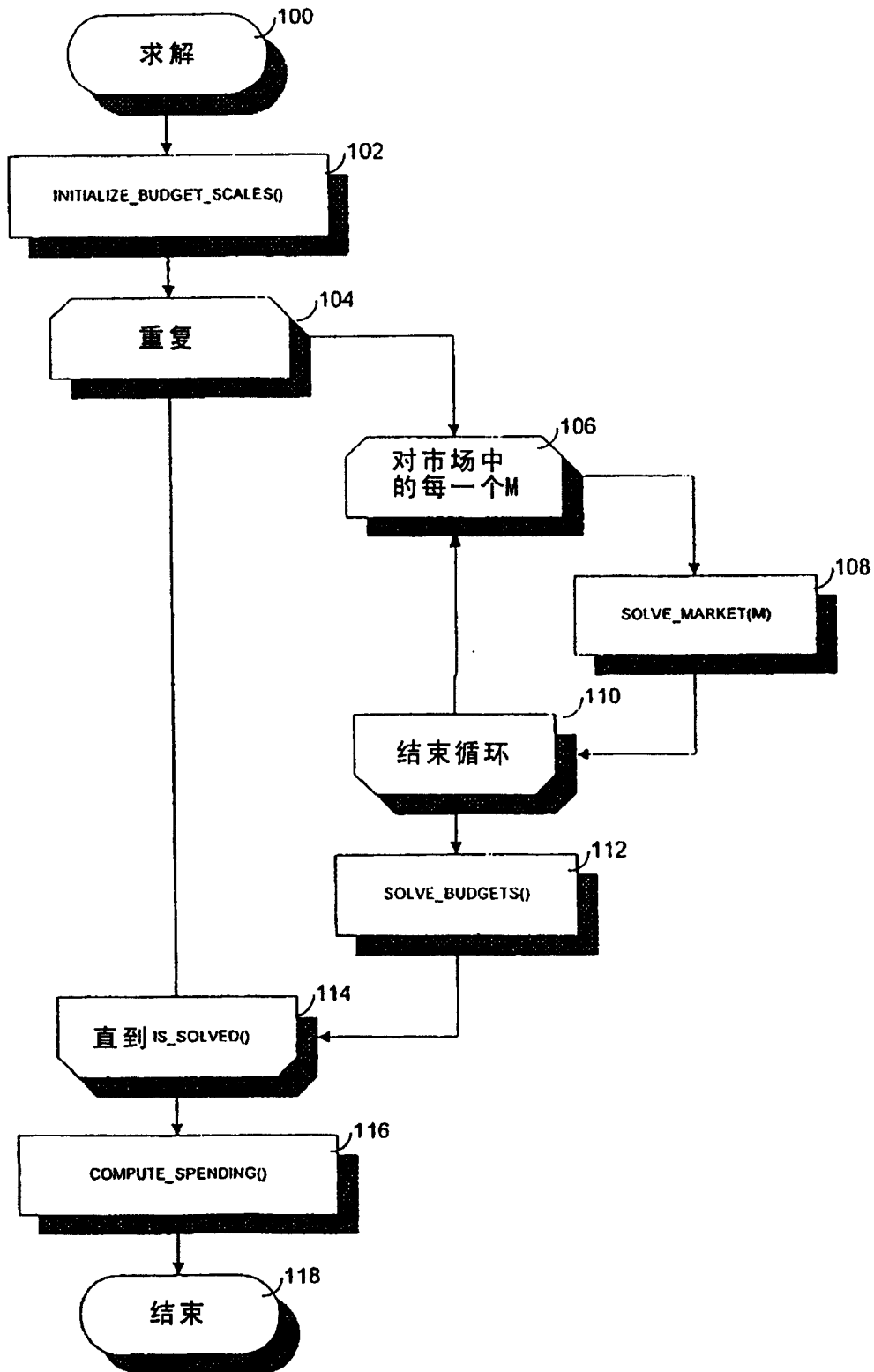


图1

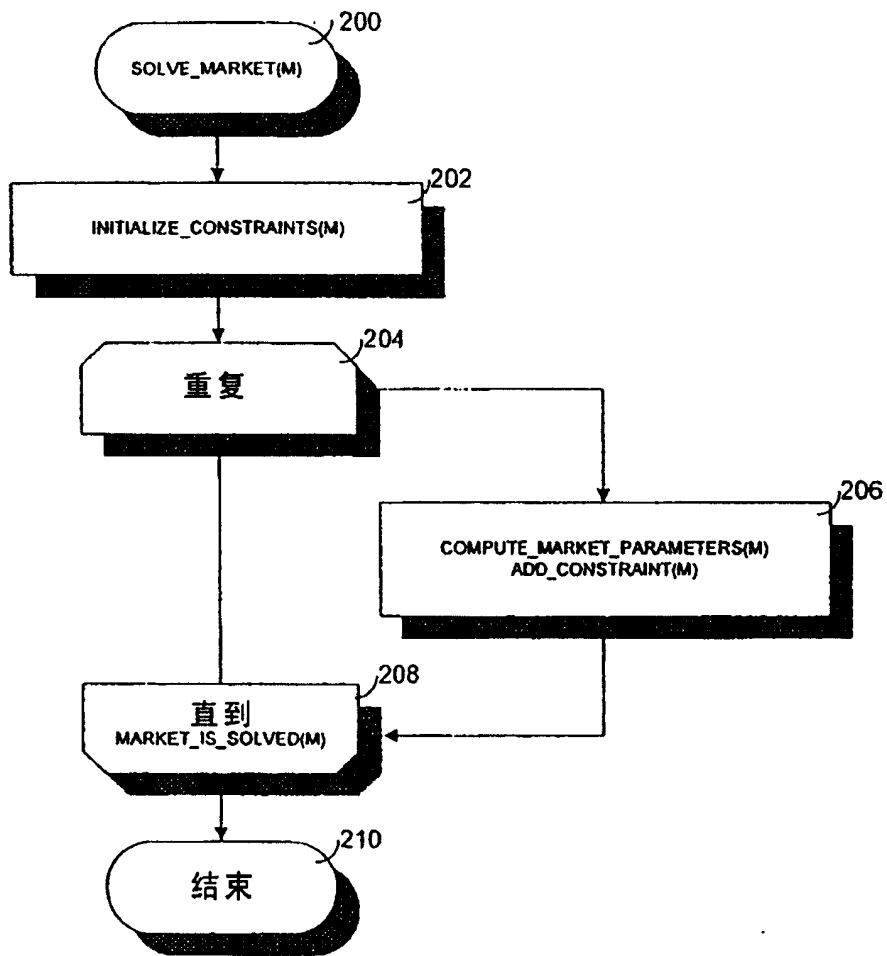


图2

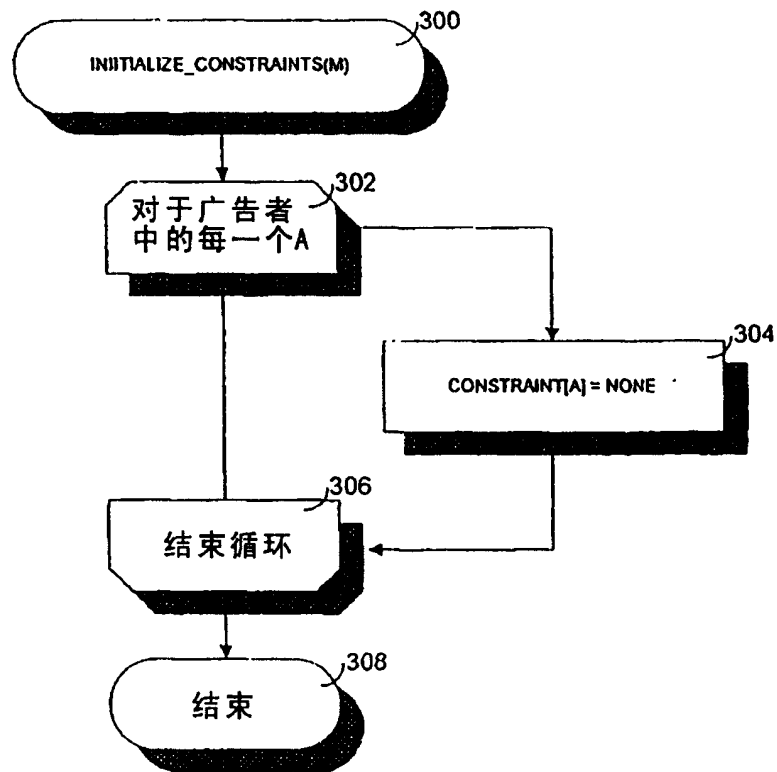


图3

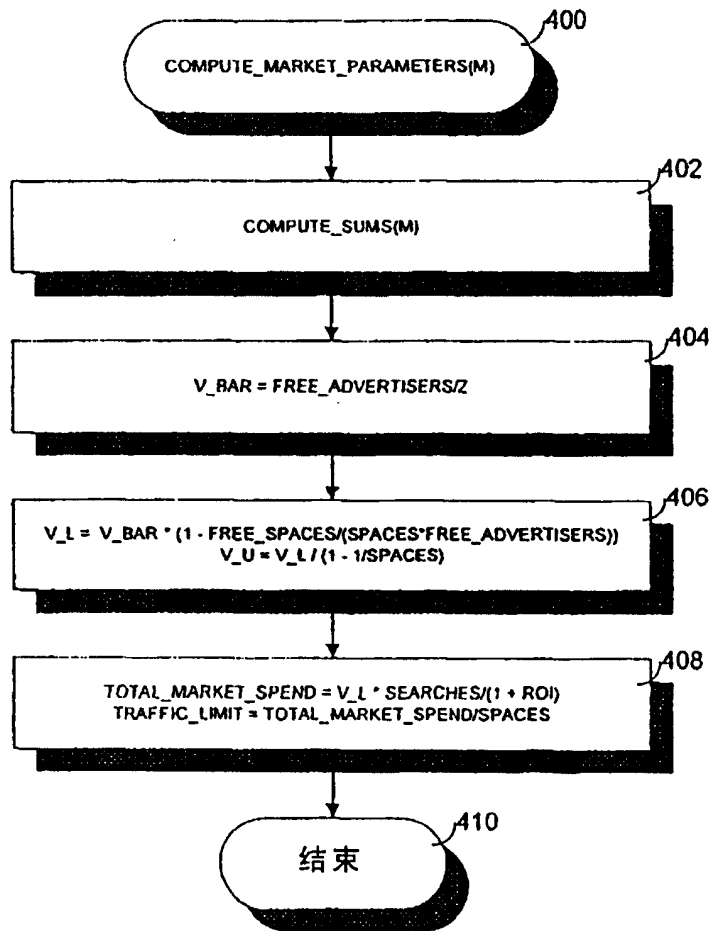


图4

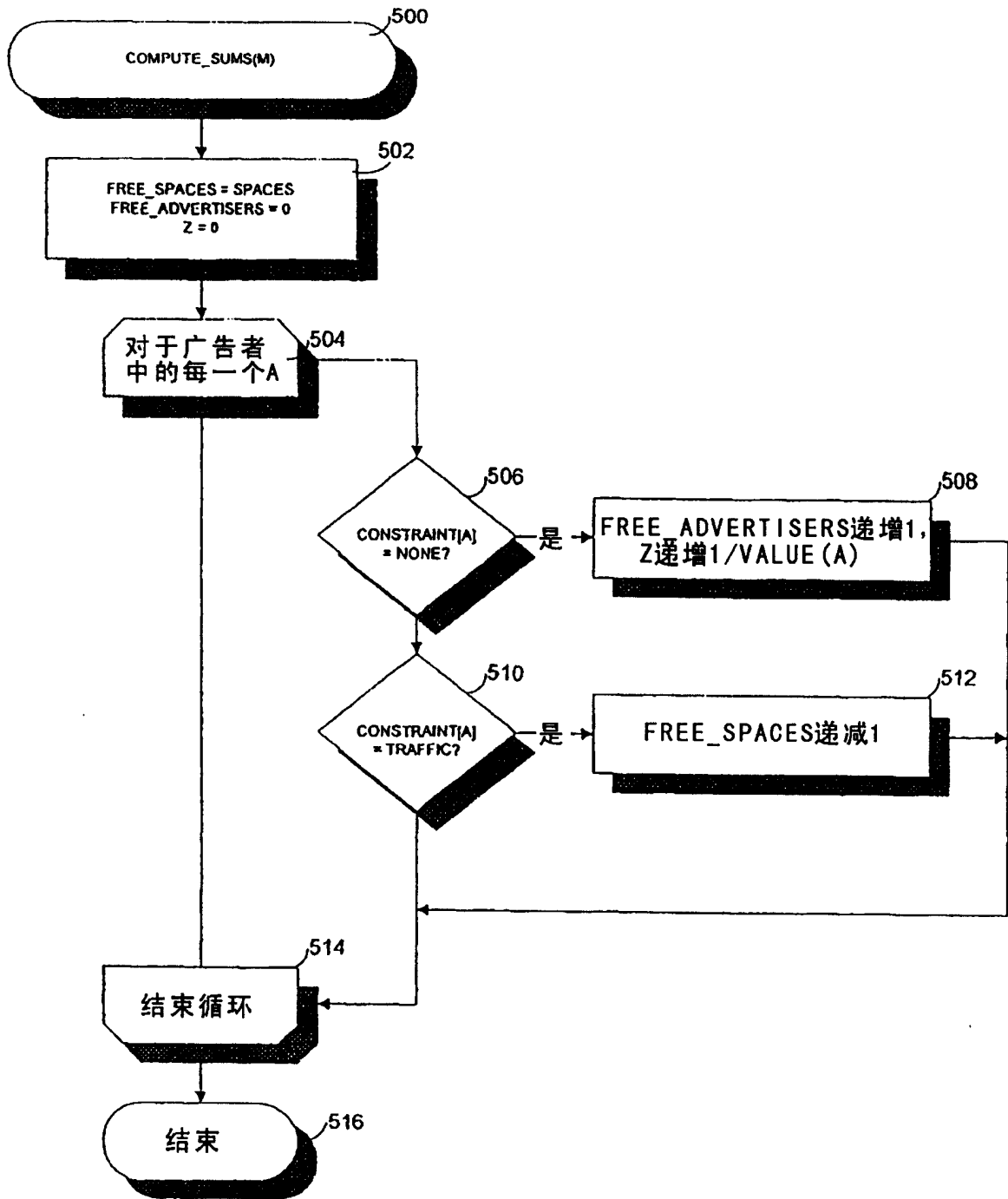


图5

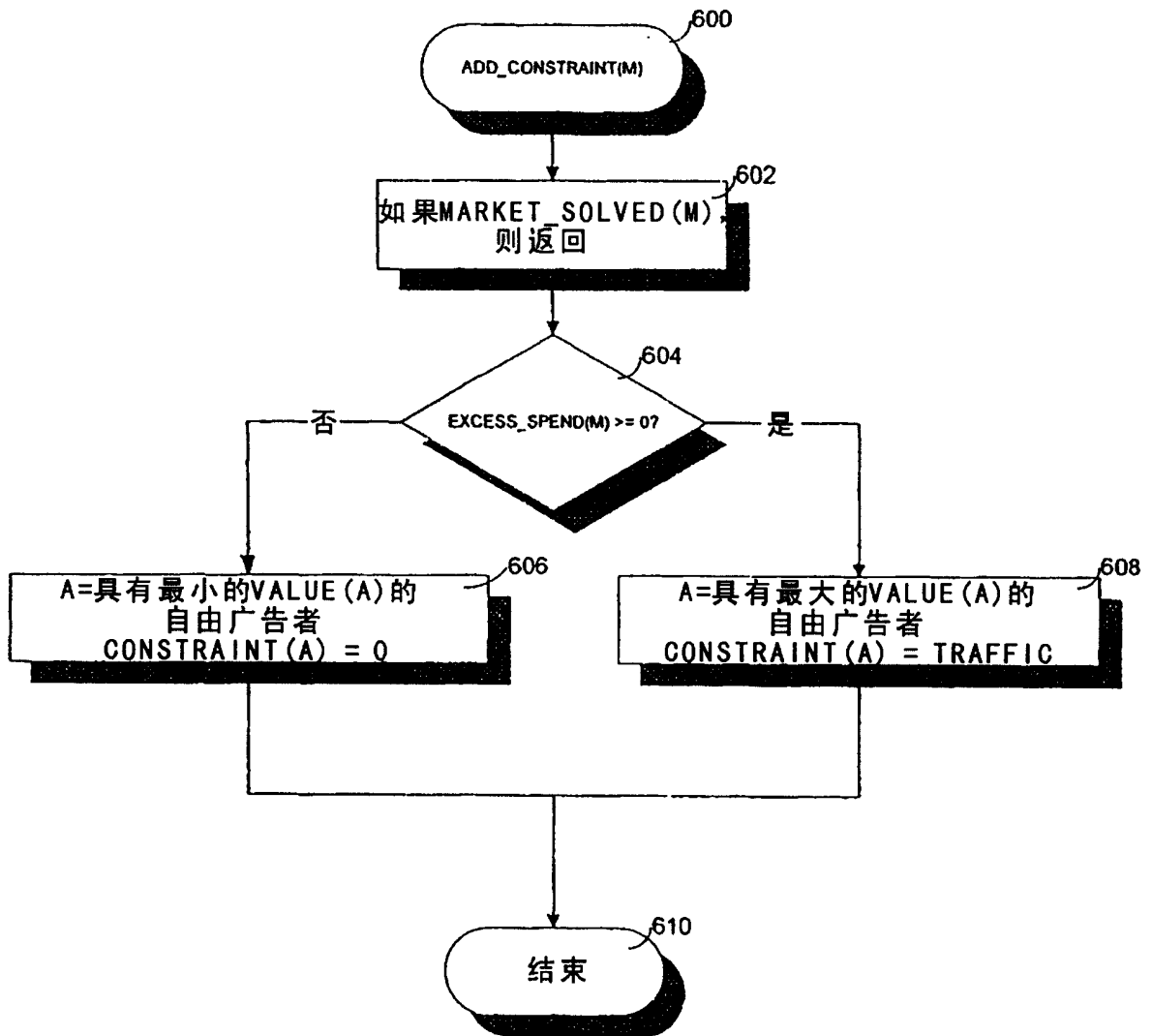


图6

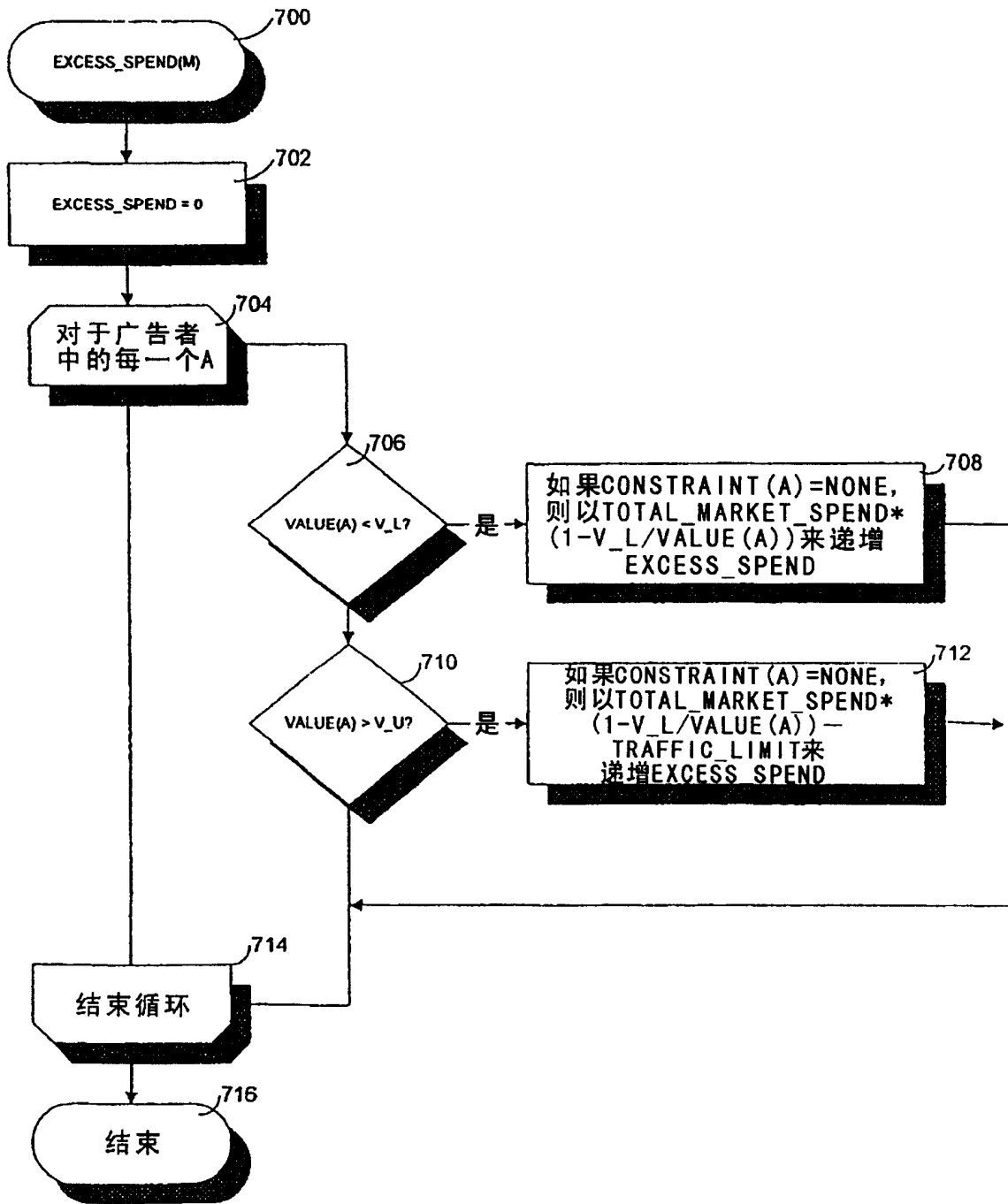


图7

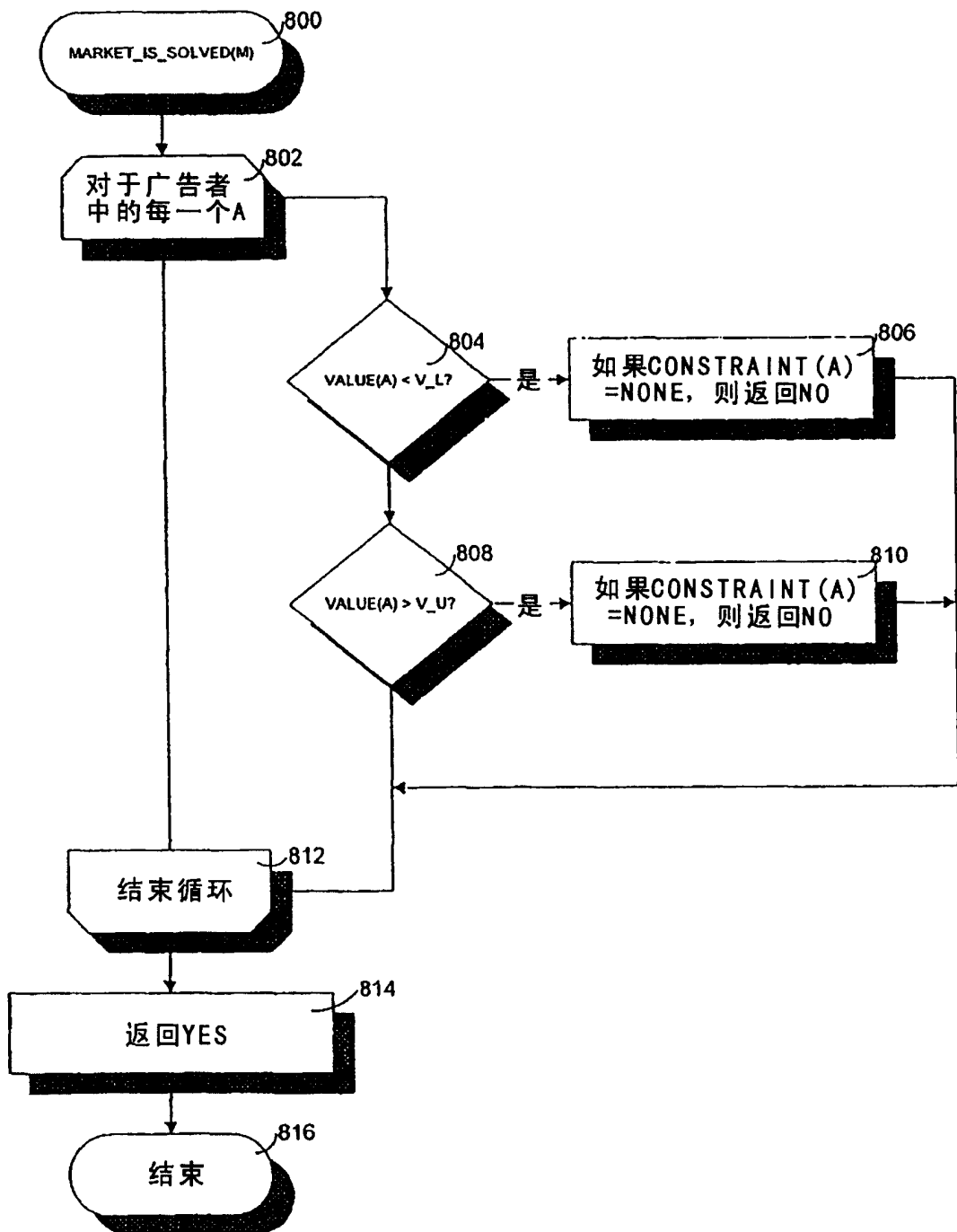


图8

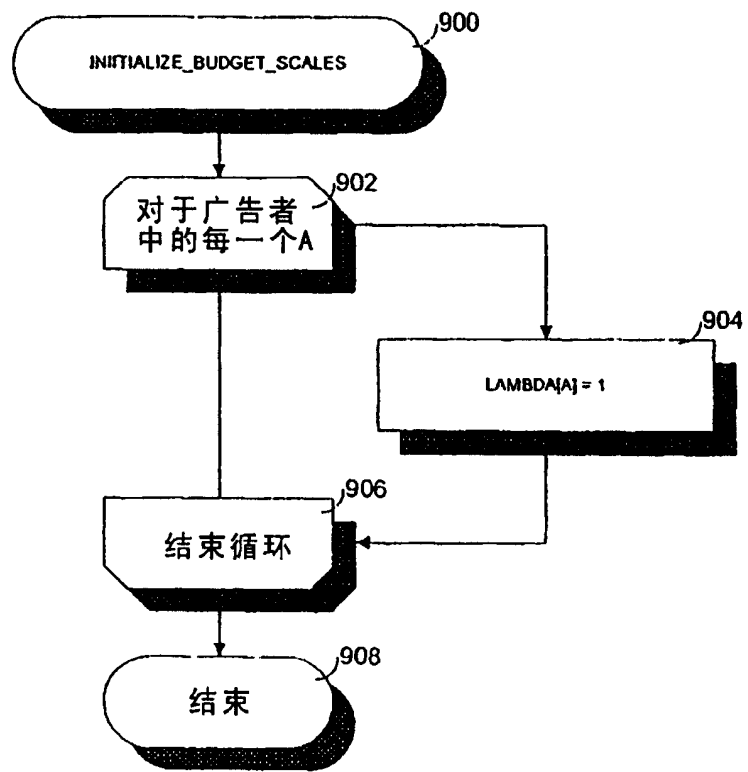


图9

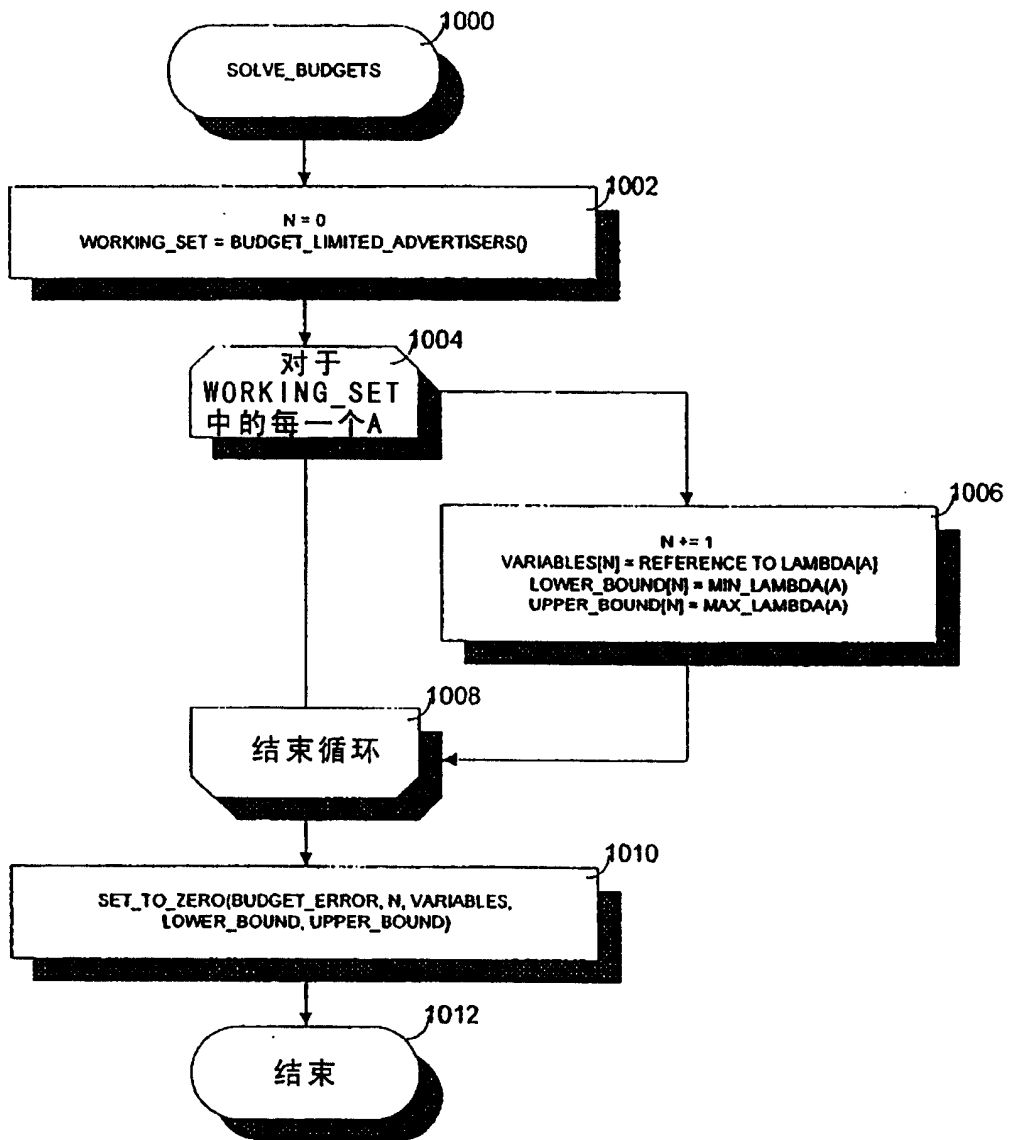


图10

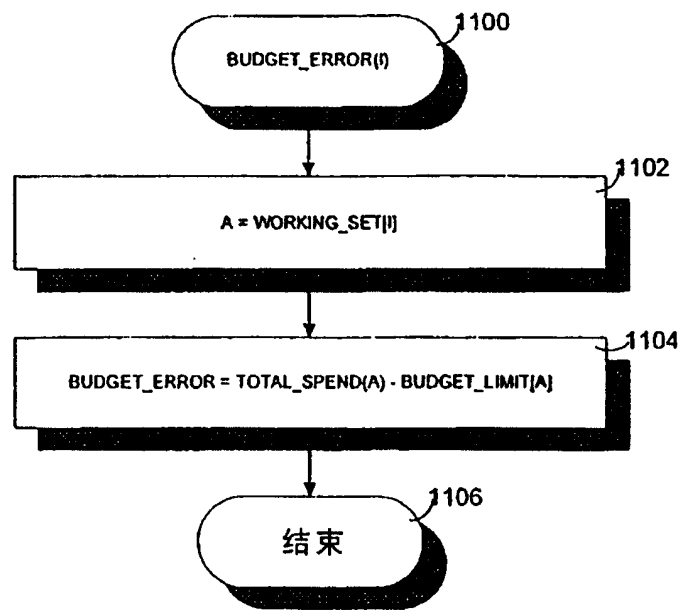


图 11

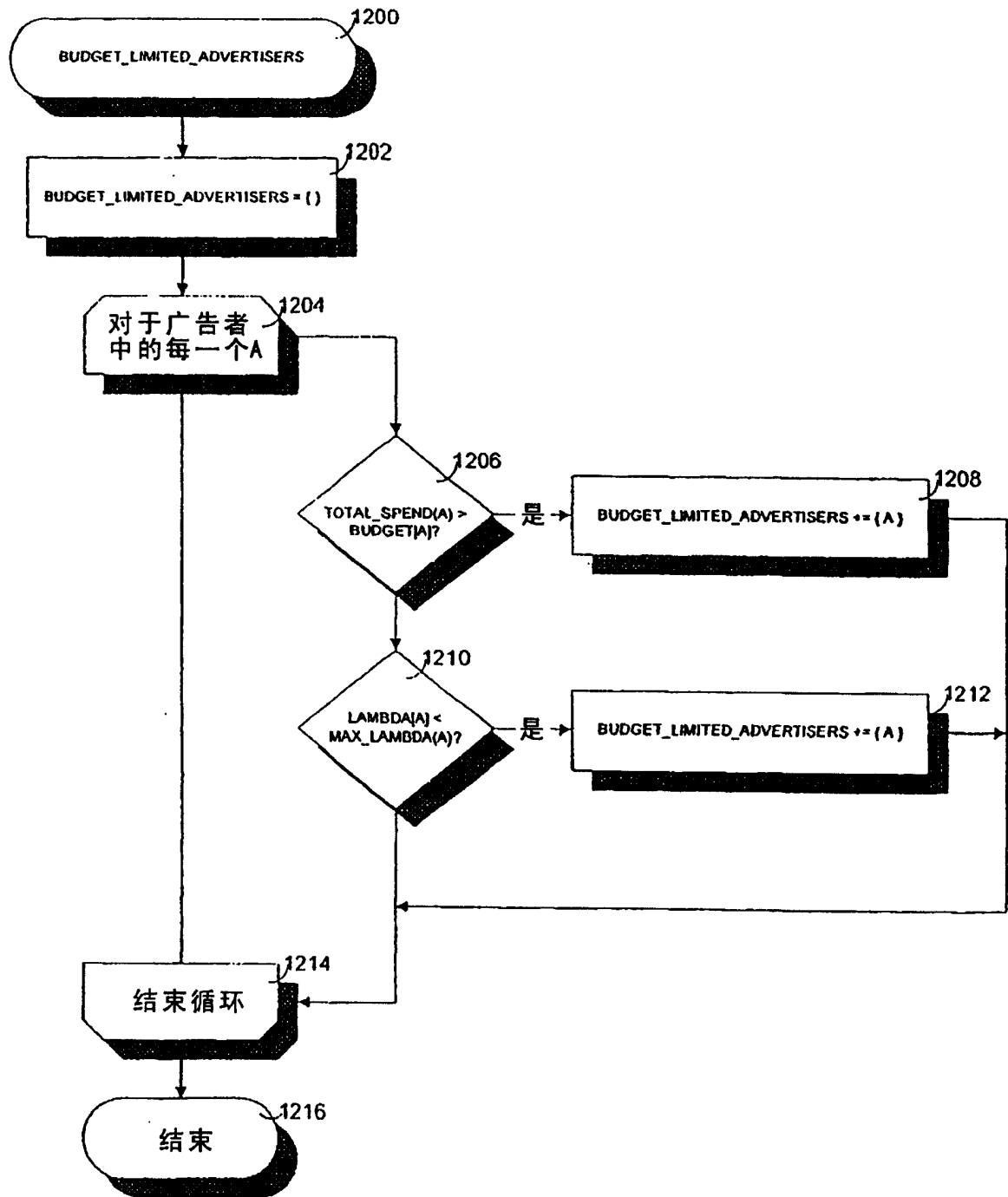


图12

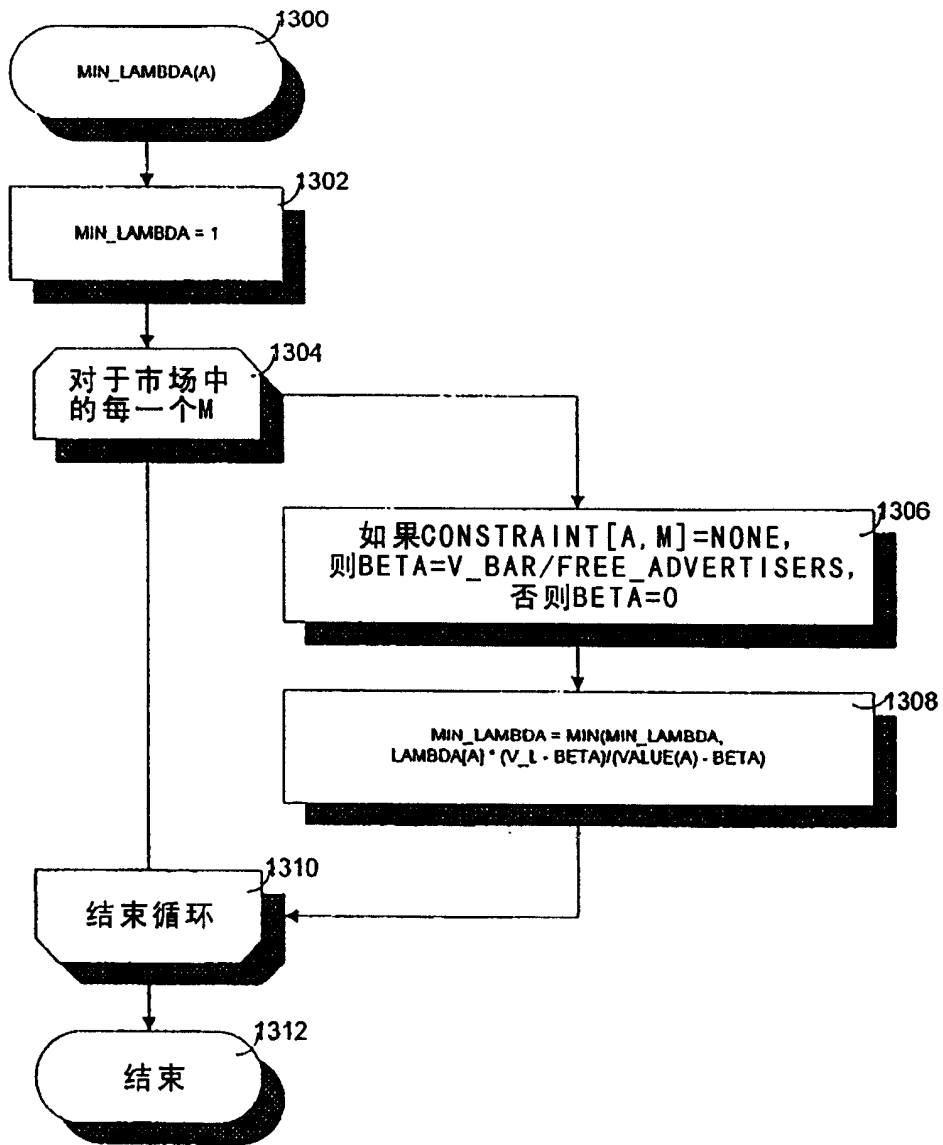


图13

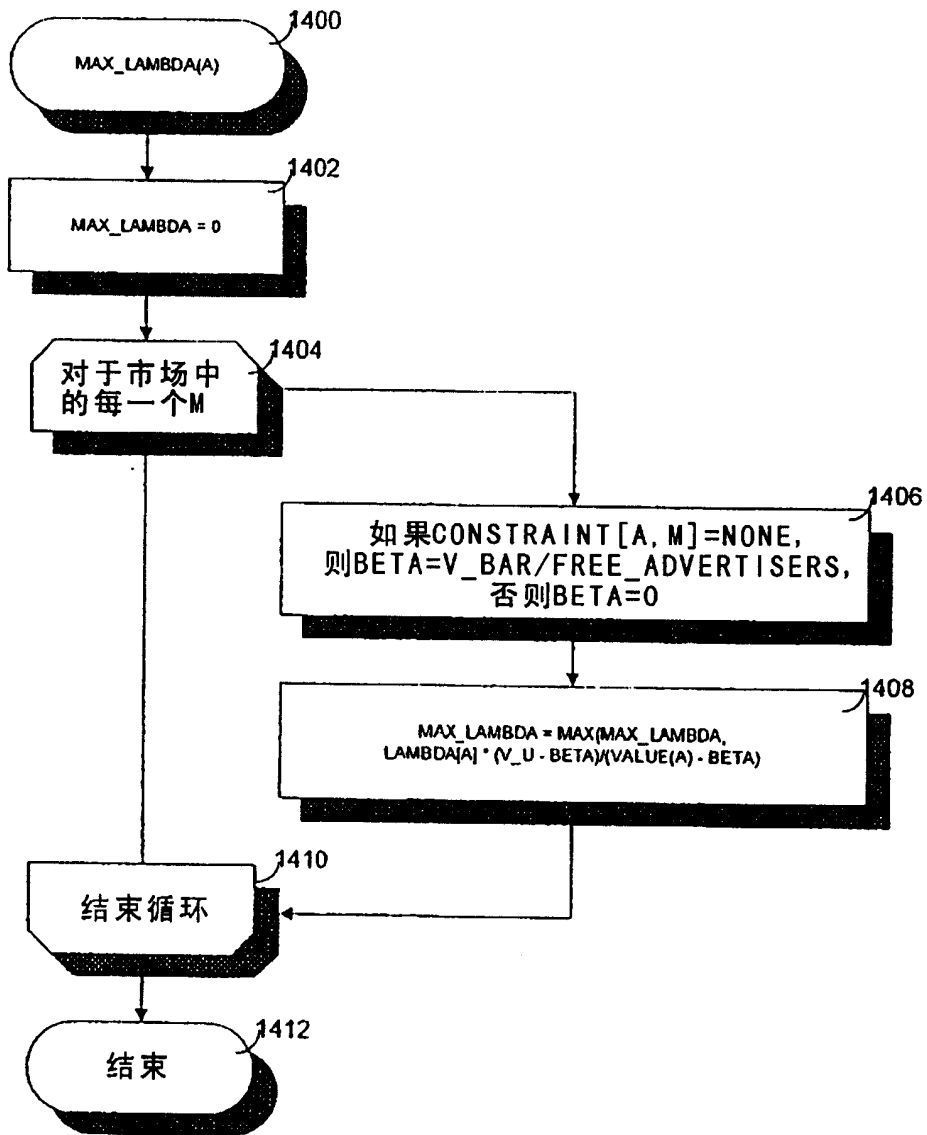


图14

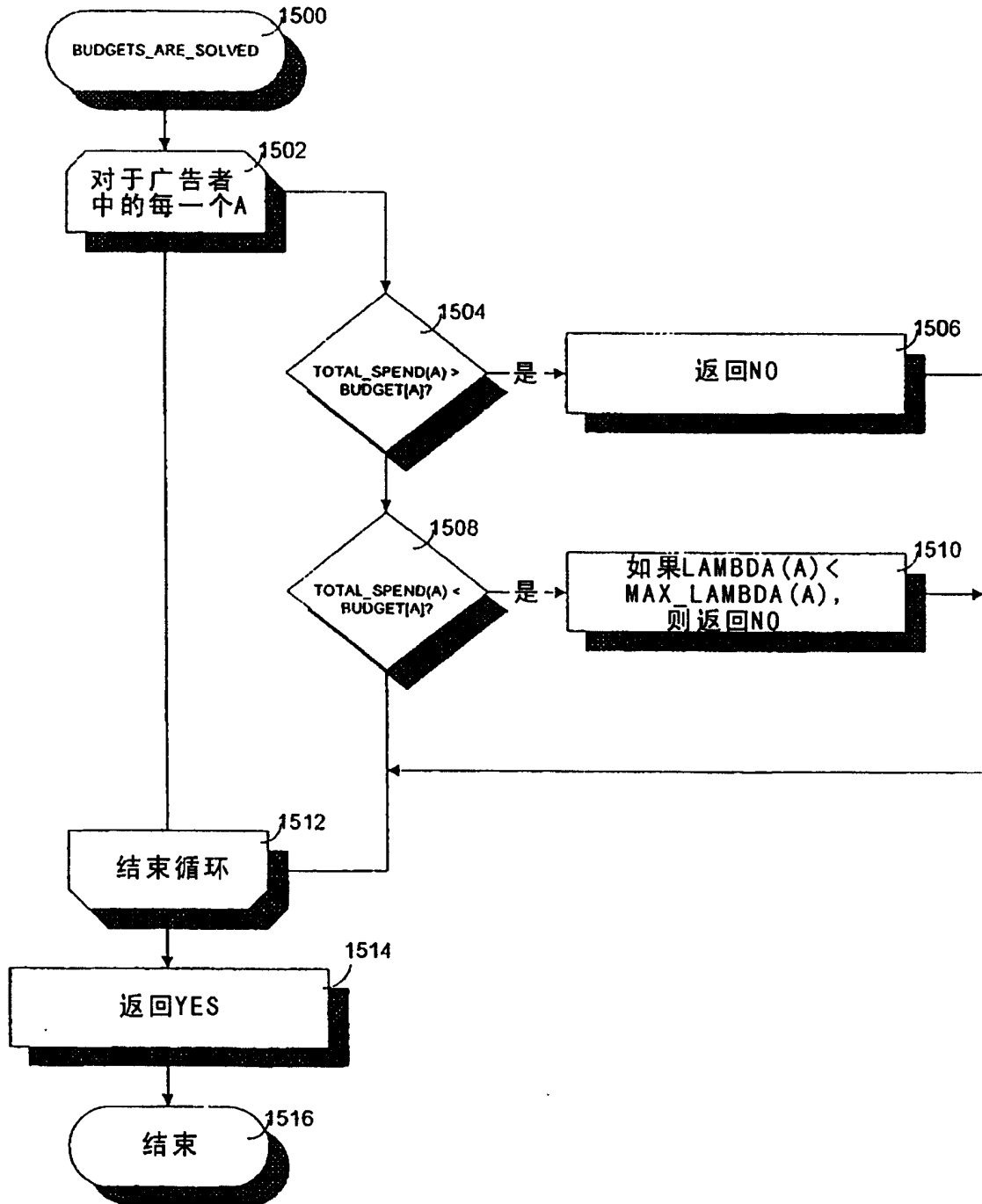


图15

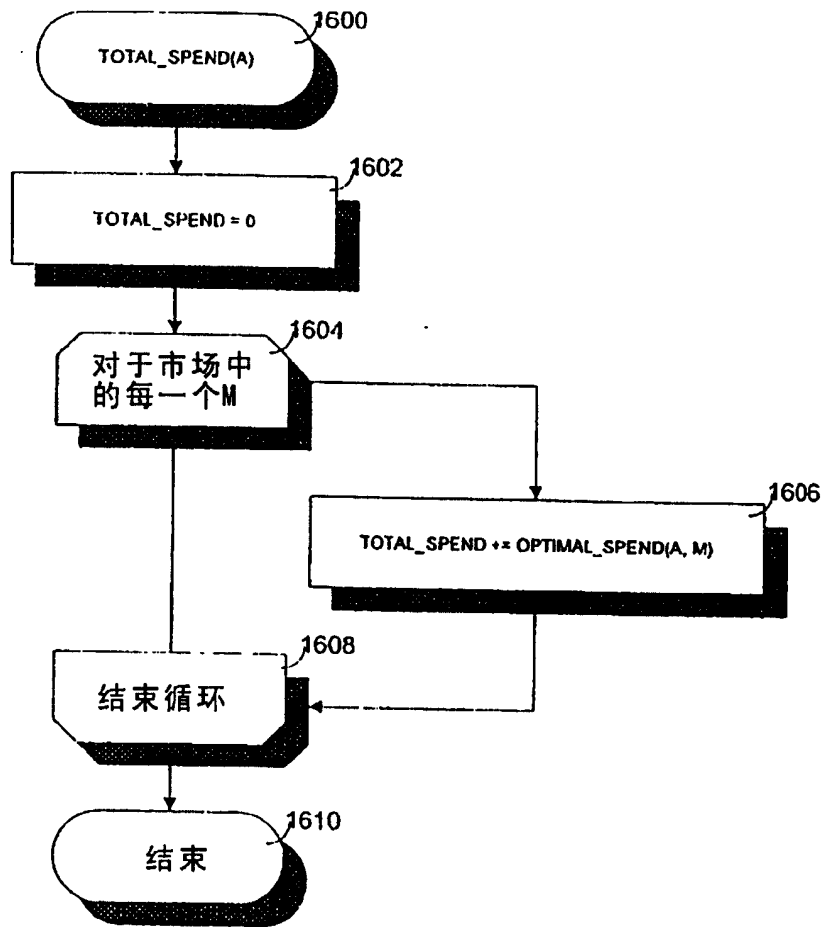


图16

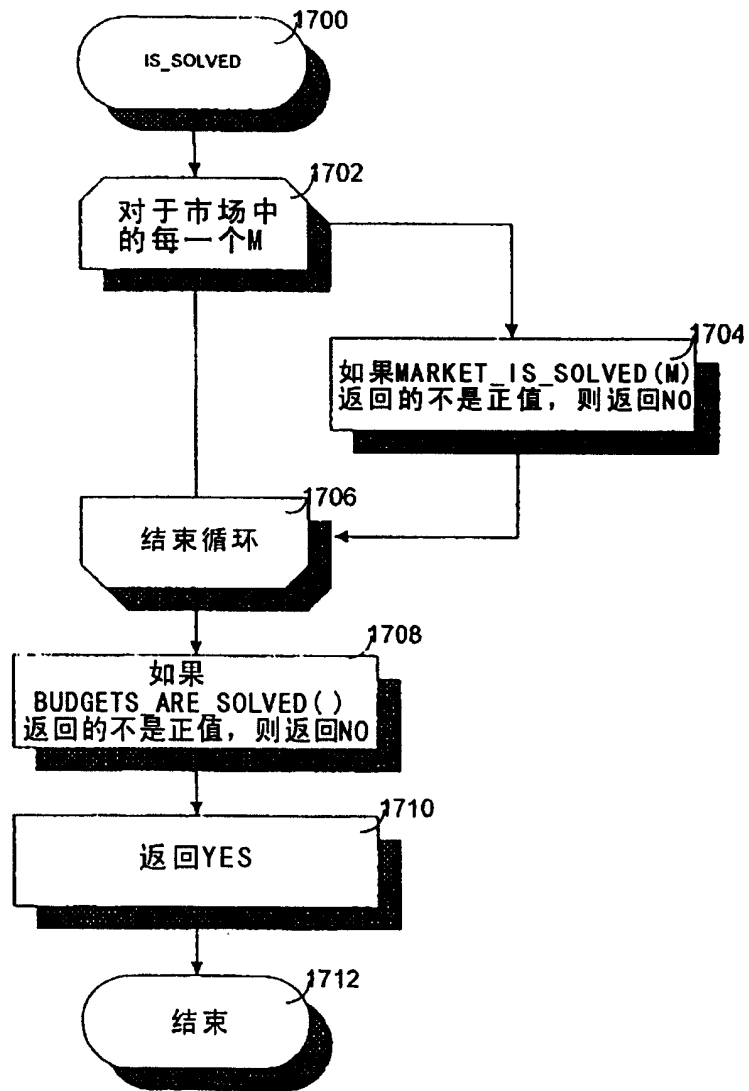


图17

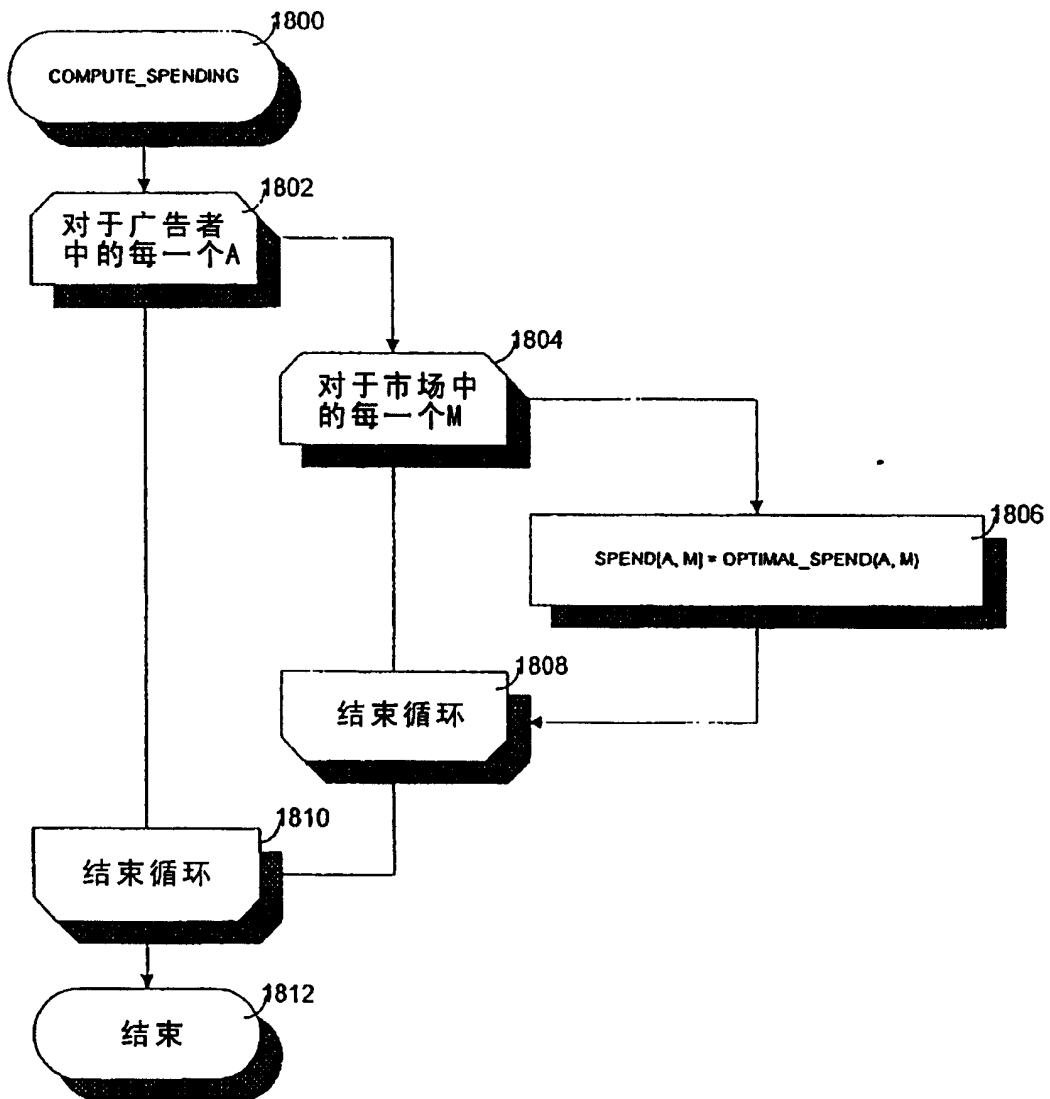


图18

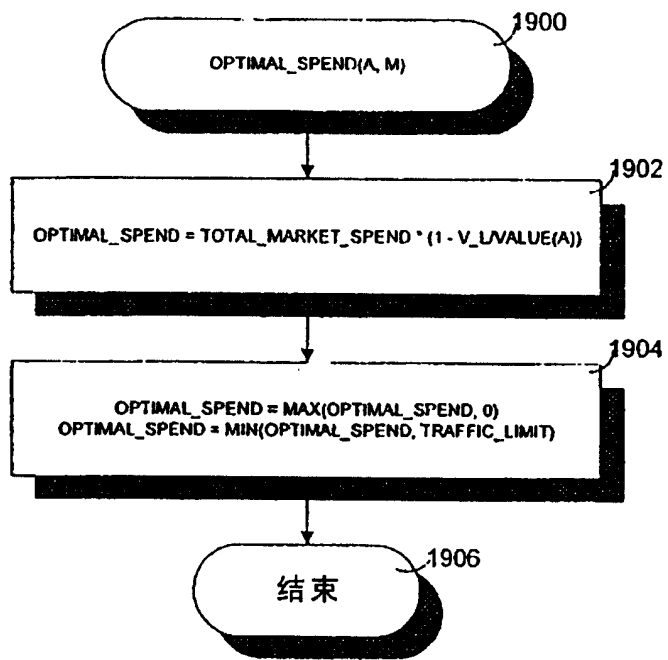


图19

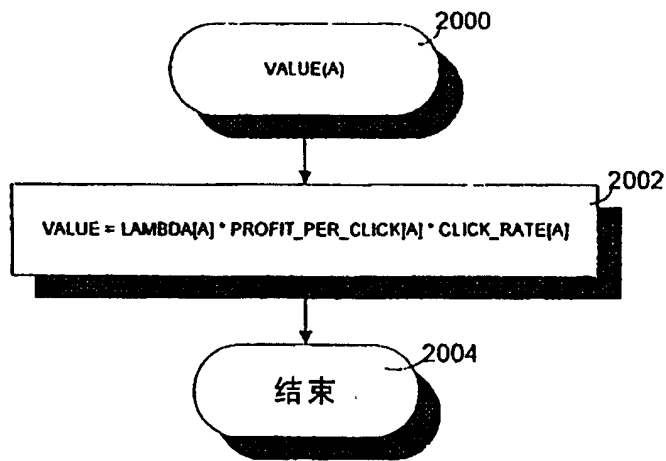


图20

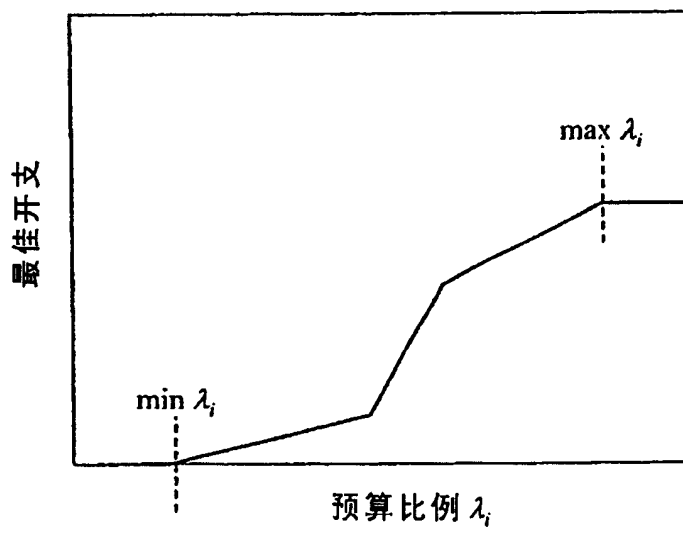


图21

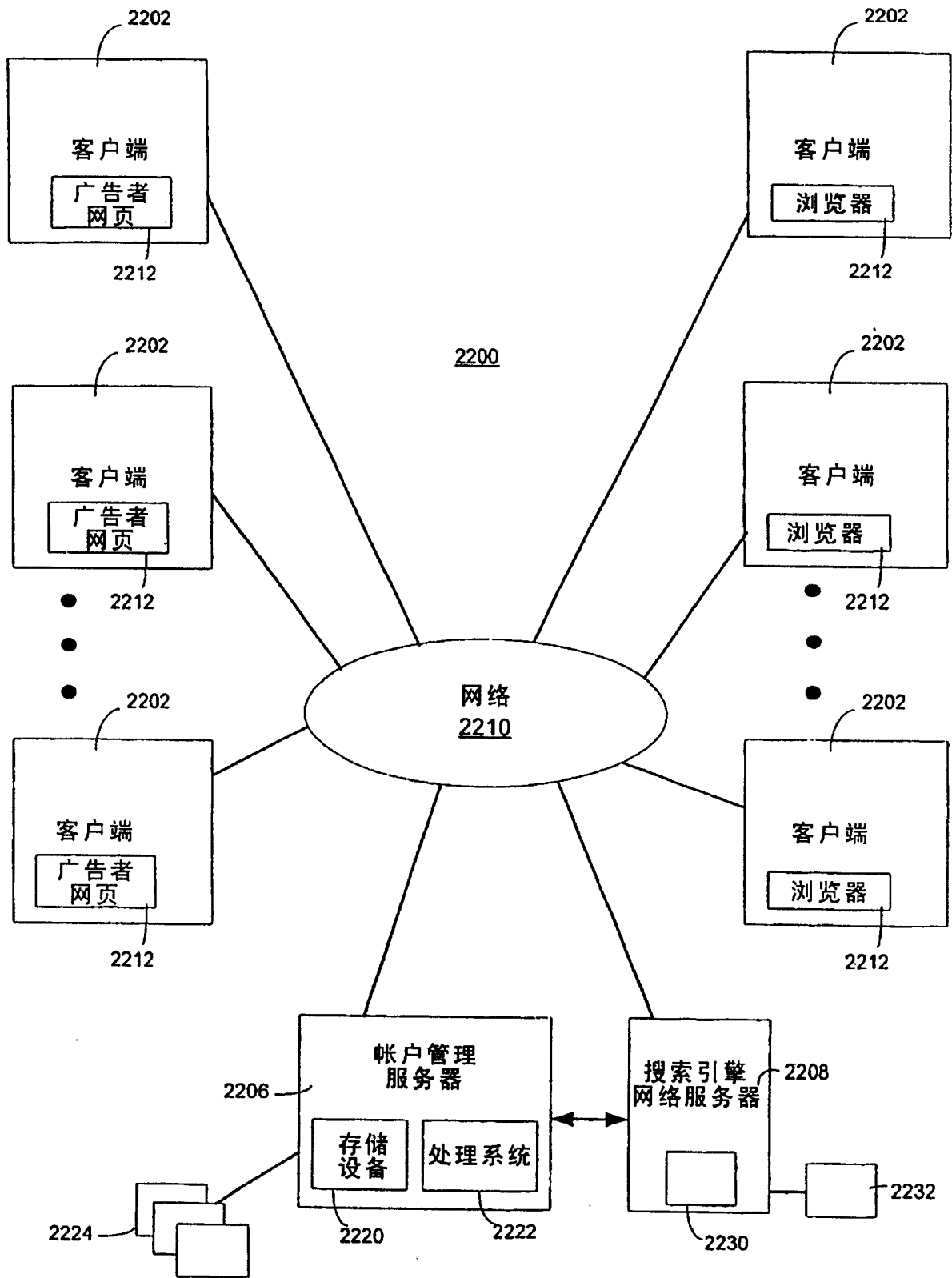


图22