文章编号: 1001-0920(2004)08-0881-04

求解网络广告资源优化模型的改进微粒群算法

齐 洁, 汪定伟
(东北大学 信息科学与工程学院, 辽宁 沈阳 110004)

摘 要: 针对网络广告的特点, 提出一个最大化广告效果函数的广告资源分配模型. 在Langheinrich线性模型的基础上, 加入分散广告显示的二次惩罚项, 使所得到的最优解更能发挥网络广告的作用; 结合微粒群(PSO)算法和模型的特点, 设计出能有效处理约束的改进PSO算法. 数值仿真证明了算法的有效性.

关键词: 网络广告; 资源分配; PSO算法; 约束处理
中图分类号: F713.82 **文献标识码**: A

Particle swarm optimization algorithm for a model of optimally scheduling web advertising resources

QI Jie, WANG Ding-wei
(School of Information Science and Engineering, Northeastern University, Shenyang 110004, China. Correspondent: QI Jie, E-mail: briskjie@163.com)

Abstract: A model of scheduling web advertising resources for maximizing advertising effect function is proposed, according to the properties of web advertising. A quadratic punishing item that separates the advertising impressions is added to the Langheinrich linear model, in order to better exert advertising efficacy with the optimized solution. An improved particle swarm optimization (PSO) algorithm is designed to handle the constraint efficiently, considered the properties of PSO algorithm and constraint of models. Simulation result shows the validity of this algorithm.

Key words: web advertising; resources scheduling; PSO algorithm; constraint handling

1 引 言

随着电子商务的发展, 广告作为企业的重要营销手段, 也以新的形式(网络广告)展现在大众面前. 从本质上讲, 网络广告是建立品牌、传播信息和销售等工作在网络这个新兴媒体上的集中体现. 目前, 大多数门户网站的收入均来自网站的广告. 因此, 网络广告商面临如下问题: 如何优化配置有限的广告空间, 以更好地发挥网络媒体的作用, 吸引广告客户.

最早提出广告资源优化模型的是Langheinrich等[1]. Langheinrich 模型是以最大化广告点击次数总和为目标的线性规划问题, 它可以划归为运输问题. 随后, Abe等[2]对网页属性进行了聚类分析, 以减小模型的规模. Tomlin[3]提出了改进的方法, 引入了熵模型均衡结果为零的最优变量.

所有这些研究都是用点击率来衡量广告效果, 然而广告客户希望访问者在不点击广告的情况下能影响尽可能多的消费者[4], 而且, 若增加广告分散度, 就能增加顾客与广告的接触面, 扩大广告的影响. 所以, 本文定义了一个具有分散惩罚项的目标函数. 另外, 上述文献没有明确提出解决问题的算法,

收稿日期: 2003-09-04; **修回日期**: 2003-11-03.
基金项目: 国家自然科学基金资助项目(70171056).
作者简介: 齐洁(1978—), 女, 云南大理人, 博士生, 从事复杂性科学在建模及优化中的应用等研究; 汪定伟(1948—), 男, 江西彭泽人, 教授, 博士生导师, 从事制造系统建模与优化、智能计算与软计算等研究.

为此, 本文提出用改进的微粒群优化算法(PSO)来求解此类问题.

2 问题描述

假设一个网站受理了 n 个广告业务, 记为 $A_j(j = 1, 2, \cdots, n)$. 网站的所有网页按属性分类可分为 m 类, 用 $1, 2, \cdots, i, \cdots, m$ 表示. 例如新浪网的网页可划分为主页、新闻、体育、娱乐、游戏、旅游、文化等. 定义单个网页的日访问量为 k_i; d_{ij} 表示广告 j 在网页 i 下的显示概率, 是决策变量; c_{ij} 表示在特定属性页面 i 下广告 j 的点击概率, 可通过历史数据统计得到; 而 h_j 表示广告客户所要求的广告印次数.

本文提出了一个最大化广告效果函数的优化模型, 以优化配置广告资源. 广告效果取决于广告点击率的大小. 通常情况下, 广告被点击便意味着广告已呈现在用户面前, 用户看到了所感兴趣的信息. 为避免线性规划中出现广告显示率过于集中在某几类页面, 应增加广告与顾客的接触面, 并通过视觉刺激发挥广告树立品牌的作用. 所以, 本文在最大化广告总体点击次数的基础上加入了一个分散惩罚项, 将广告显示率分散到不同的网页.

$$\max\left(\omega_1 \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}k_i d_{ij} - \frac{\omega_2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}k_i \sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}^2\right). \tag{1}$$

其中: 目标函数中第 2 项的最大值(即去掉负号后的最小值) 是在各个 d_{ij} 相等时取得的, 所以起到了将广告 i 分散于不同属性页的作用; 两个权值 ω_1 和 $\omega_2(\omega_1 + \omega_2 = 1)$ 规定了第 1 项和第 2 项在目标函数中将起多大作用. 同时满足如下约束:

$$\text{s.t.}\quad \sum_{i=1}^{m} k_i d_{ij} = h_j,\ j = 1, \cdots, n; \tag{2}$$

$$\sum_{j=1}^{n} d_{ij} = 1,\ i = 1, \cdots, m; \tag{3}$$

其中 $d_{ij} \geq 0$.

约束方程的含义如下: 1) 广告在各种属性页面的印次总数达到广告客户的要求; 2) 所有广告在各个属性页面的显示概率之和等于 1; 3) 显示概率非负. 当约束方程满足 $\sum_{j=1}^{n} h_j = \sum_{i=1}^{m} k_i$ 时, 问题存在可行解. 此模型等价于非线性运输问题.

3 用 PSO 算法求解广告资源优化模型

微粒群算法[5~7]是由 Kennedy 和 Eberhart 等人于 1995 年提出的一种演化算法, 核心要素是一群微粒在搜索空间中的位置(表示决策变量) 和运动速度. 设第 i 个微粒的位置为 $X_i = (x_{i1}, x_{i2}, \cdots, x_{in})$, 它经历的最好位置为 p_{best}, 微粒群体所经历过的最好位置为 g_{best}. 下一代微粒速度 $V_i = (v_{i1}, v_{i2}, \cdots, v_{in})$ 的更新和位置更新公式分别为

$$v_{id} = \omega v_{id} + c_1\text{rand}(\,)(p_{\text{best}} - x_{id}) + c_2\text{Rand}(\,)(g_{\text{best}} - x_{id}), \tag{4}$$

$$x_{id} = x_{id} + v_{id}. \tag{5}$$

其中: ω 是惯性权重, c_1 和 c_2 分别为加速常数, rand( ) 和 Rand( ) 为两个独立产生的在[0, 1] 之间变化的随机函数. PSO 适合求解可行域连通的连续函数最优值, 但在本文模型中的约束为线性方程组, 可行域狭小, 因而难以设计出一个合适的惩罚函数. 如果惩罚较大, 则惩罚后的目标函数将不连续; 惩罚较小, 得到的最优解可能在可行集外. 并且, 算法的大部分时间耗费在从非可行域跳到可行域的运算中, 运算效率较低. 因此, 本文不用惩罚函数, 而根据约束集和 PSO 算法的特点, 设计每一步迭代微粒都可行的 PSO 算法.

3.1 初始化过程

用矩阵表达决策变量 $d_{ij}(i = 1, \cdots, m$ 且 $j = 1, \cdots, n)$ 的集合, 则

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{bmatrix}.$$

在 PSO 算法中, 微粒坐标用 $D = \{d_{ij}\}$ 表示, 描述微粒在 mn 维空间中的运动位置, 而微粒的运动速度可由矩阵 $V = \{v_{ij}\}$ 表示. 基于问题的约束条件(2) 和(3), 可用下述初始化过程产生满足所有约束的初始微粒[8,9]:

(1) 初始化 1

Input

Array $h[j]$; Array $a[i] = 1$ and $k[i]$. for $j = 1, \cdots, n$; for $i = 1, \cdots, m$

Begin

设定集合 $Ag = \{1, 2, \cdots, mn\}$

Repeat

从集合中随机选一个数 r

计算相应的行和列: $i = \text{Int}((r - 1)/n + 1)$, $j = (r - 1) \bmod n + 1$

令 $d[i][j] = \min(a[i], h[j]/k[i])$

修改数据: $a[i] = a[i] - d[i][j]$, $h[j] = h[j] - d[i][j] * k[i]$

集合 $Ag = Ag \setminus \{r\}$

Until 集合 Ag 为空

End

初始化过程 1 产生一个满足约束条件的微粒 $D = \{d_{ij}\}$, 并生成一个至多有 $m + n - 1$ 个非零元素的矩阵, 表示凸可行集的顶点, 对应约束集中的基可行解.

为了求解最优化问题(1), 需要设计可保持微粒可行性条件的速度. 由问题可行集和 PSO 算法中微粒速度及位置更新公式的特点, 可导出以下定理:

定理 1　如果可行微粒的初始速度 V 满足条件

$$\sum_{i=1}^{m} k_i v_{ij} = 0, j = 1, \cdots, n;$$

$$\sum_{j=1}^{n} v_{ij} = 0, i = 1, \cdots, m. \tag{6}$$

则按更新公式(5) 迭代, 得到的微粒满足约束条件(2).

证明　设可行微粒 $D = \{d_{ij}\}(i = 1, \cdots, m; j = 1, \cdots, n)$, 按式(5) 更新后得到微粒 $(d_{ij} + v_{ij})$, 代入式(2), 并且由式(6), 可得

$$\sum_{i=1}^{m} k_i (d_{ij} + v_{ij}) = \sum_{i=1}^{m} k_i d_{ij} = h_j, j = 1, \cdots, n;$$

$$\sum_{j=1}^{n} d_{ij} + v_{ij} = \sum_{i=1}^{m} d_{ij} = 1, i = 1, \cdots, m.$$

所以更新后的微粒 $(d_{ij} + v_{ij})$ 满足约束条件(2). □

定理 2　如果式(4) 中, 每个微粒各分量的速度更新都采用相等的 rand( ), 以及相等的 Rand( ), 且初始速度满足式(6), 则按式(4) 和(5) 更新的微粒总满足约束条件(2).

证明略.

因此, 为了在每一步迭代中都能获得满足约束条件(2) 的微粒群, 不但要产生满足约束的可行微粒, 还需产生满足式(6) 的初始速度. 根据定理 2, 可由初始化 1 产生的可行微粒得到满足式(6) 的初始速度.

(2) 初始化 2

Begin

运用初始化 1, 得到一群新的可行微粒(规模设为 Num)

Repeat

随机从微粒种群中取出两个微粒 $D[p]$ 和 $D[q]$

令 $V[k] = (D[p] - D[q])/(g * \text{rand}( ) + 0.2)$, 其中: $p, q \in \{1, \cdots, \text{Num}\}$, $g \in R$

$k = k + 1$

Until $k >$ Num

End

3.2　算法步骤

Step1: 运用初始化 1 初始化一群微粒 d_{ij}; 并运用初始化 2 得到初始速度. 注意, 在初始化 2 中所得到的一群新的微粒与算法的初始微粒不同.

Step2: 计算每个微粒的适值.

Step3: 对于每个微粒, 将其适值与其所经历过的最好位置 p_{best} 进行比较, 并检验这个微粒的可行性条件(3). 如果这个微粒的适值较好且满足 $d_{ij} \geq 0$, 则将其作为当前的最好位置 p_{best}(微粒的适值和位置同时设定).

Step4: 对于每个微粒, 将其适值与全局所经历的最好位置 g_{best} 进行比较, 并检验微粒的可行性条件(3), 如果这个微粒适值较好且满足 $d_{ij} \geq 0$, 则将其设为 g_{best}(微粒的适值和位置同时设定).

Step5: 根据式(4) 变化微粒的速度, 为满足约束(2), 令每个微粒各个分量的 rand( ) 和 Rand( ) 值相等; 然后设定一个小的正数 $\epsilon = 10^{-6}$. 如果一个微粒的速度小于 ϵ, 并在 10 步的迭代中仍保持小于 ϵ, 则按初始化 2 重新初始化速度和位置, 并令 p_{best} 为当前位置; 然后根据式(5) 变化微粒的位置.

Step6: 如未达到结束条件(通常为足够好的适值或达到一个预设最大代数 $G_{\max}$), 则返回 Step2.

上述算法与基本的 PSO 算法的区别在于:

1) 对于约束的处理, 首先产生可行的初始微粒和可行速度, 保证每次迭代的微粒群满足约束条件(2); 为了同时满足约束条件(3), 在 Step3 和 Step4 比较适值时, 加入了可行条件(3) 的判断, 只有当微粒同时满足适值较好和 $d_{ij} \geq 0$ 时, 才将此微粒存入 p_{best} 或 g_{best}, 这样便避免了微粒向适值较好的非可行解方向运动.

2) 加入动态调整机制. 通常当 PSO 算法运行到一定的阶段便会收敛到稳定状态, 此时所有微粒将聚集到一块, 微粒的调整速度越来越小, 逐渐趋于零. 所以, 此时应重新初始化速度和位置, 令微粒恢复活力, 发挥寻优的作用.

4　计算实例

算法用 JDK1.3 实现, 在 P4 1.8G, 256M 计算机上对多个问题进行了成功的仿真, 证明了算法的有效性. 本文仅以规模为 10 个属性页, 20 个广告的问题为例. 随机产生 20 组数据, 每组数据运算 50 次, 得到其最优结果和达优率如表 1 和表 2 所示. 其中参数 ω_1 和 ω_2 分别取为 0.8 和 0.2. PSO 算法参数设置为: $\omega = 0.5$, $c_1 = 0.7$, $c_2 = 0.7$. 表中的数值说明

本文提出的 PSO 算法是有效的. 仿真结果表明, 由于变量较多(200 个), 可行空间的维数也较大, 所以种群规模取值较大时, 算法效果比较好.

表 1 优化结果(针对其中一组数据的均值)

微粒个数	迭代次数		
	1 000	5 000	10 000
20	822.3	833	860.45
100	852.12	860.61	861.08

表 2 达优率 (%)

微粒个数	迭代次数		
	1 000	5 000	10 000
20	32.4	49	91.2
100	81.8	91.40	93.01

5 结 论

本文提出的广告资源分配模型, 通过最大化广告效果函数来优化广告资源在一个网站上的分配, 以更大地发挥网络广告的作用, 增强广告效果, 为网站吸引更多的广告客户. 求解模型采用了改进 PSO 算法, 此算法具有寻优能力较强、易于实现等优点, 是解决此类问题的有效途径. 由于算法对约束条件进行了有效处理, 使得算法具有广泛的应用范围, 适于求解各种非线性目标函数的运输问题.

参考文献(References):

[1] Langheinrich M, Nakamura A, Abe T K N, et al. Unintrusive customization techniques for web advertising[J]. *Computer Networks*, 1999, 31(11-16): 1259-1272.

[2] Abe N, Nakamura A. Learning to optimally schedule internet banner advertisements[A]. *Proc of the 16th Int Conf on Machine Learning*[C]. Bled, Slovenia, 1999. 12-22.

[3] Tomlin J. An entropy approach to unintrusive targeted advertising on the web[J]. *Computer Networks*, 2000, 33(1-6): 767-774.

[4] Zeff R, Aronson B. Internet 广告实战策略[M]. 北京华中兴业发展有限公司译. 第 2 版. 北京: 人民邮电出版社, 2001.

[5] Kennedy J, Eberhart R. Particle swarm optimization[A]. *Proc IEEE Int Conf on Neural Networks*[C]. Perth, 1995. 1942-1948.

[6] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[A]. *Proc 6th Int Symposium on Micro Machine and Human Science*[C]. Nagoya, 1995. 39-43.

[7] 谢晓峰, 张文俊, 杨之康. 微粒群算法综述[J]. 控制与决策, 2003, 18(2): 129-134.
(Xie X F, Zhang W J, Yang Z K. Overview of particle swarm optimization[J]. *Control and Decision*, 2003, 18(2): 129-134.)

[8] Michalewicz Z, Vignaux G A, Hobbs M. A non-standard genetic algorithm for the nonlinear transportation problems[J]. *ORSA J on Computing*, 1991, 3(4): 307-316.

[9] 玄光男, 程润伟. 遗传算法与工程设计[M]. 汪定伟, 唐家福, 黄敏译. 北京: 科学出版社, 2000. 186-189.

(上接第 876 页)

参考文献(References):

[1] 李云钢, 常温森. 磁浮列车悬浮系统的串级控制[J]. 自动化学报, 1999, 25(2): 247-251.
(Li Y G, Chang W S. Cascade control of an EMS maglev vehicle's levitation control system[J]. *Acta Automatica Sinica*, 1999, 25(2): 247-251.)

[2] Wolff E A, Skogestad S. Temperature cascade control of distillation columns[J]. *Industrial Engineering & Chemical Research*, 1996, 35(2): 475-484.

[3] Kemin Zhou, Doyle J C, Glover K. *Essentials of Robust Control*[M]. Upper Saddle River: Prentice Hall, 1998.

[4] Lee Y H, Oh S G, Park S W. Enhanced control with a general cascade control structure[J]. *Industrial Engineering & Chemical Research*, 2002, 41(11): 2679-2688.

[5] Huang H P, Chien I L, Lee Y C. Simple method for tuning cascade control systems[J]. *Chemical Engineering Communications*, 1998, 165(1): 89-121.

[6] 陈元杰, 汪洋, 苏宏业, 等. 串级控制系统的 PID 参数自动整定算法[J]. 控制与决策, 1996, 11(5): 580-584.
(Chen Y J, Wang Y, Su H Y, et al. Auto-tuning PID method for cascade control systems[J]. *Control and Decision*, 1996, 11(5): 580-584.)

[7] Morari M, Zafiriou E. *Robust Process Control*[M]. Englewood Cliffs: Prentice Hall, 1989.