第19卷第11期 控 制 与 决 策 2004年11月
Vol 19 No. 11 Control and Decision Nov. 2004

文章编号: 1001-0920(2004)11-1213-05

利用模糊次梯度算法求解拉格朗日松弛对偶问题

周 威, 金以慧
(清华大学 自动化系, 北京 100084)

摘 要: 针对利用次梯度算法处理拉格朗日松弛对偶问题时, 计算过程容易出现振荡, 求解效率较低的问题, 首先提出了一种基于模糊理论的次梯度算法, 利用隶属度函数给出迭代过程中所有次梯度的合适权重, 并将它们线性加权得到新的迭代方向; 其次证明了算法的收敛性; 最后通过仿真实验验证了该方法的有效性

关键词: 拉格朗日松弛; 次梯度算法; 模糊理论; 对偶
中图分类号: O232 **文献标识码**: A

Fuzzy subgradient algorithm for solving Lagrangian relaxation dual problem

ZHOU Wei, JIN Yi-hui
(Department of Automation, Tsinghua University, Beijing 100084, China. Correspondent: ZHOU Wei, E-mail: zhouw99@mails tsinghua edu cn)

Abstract: To the problem of zigzaging happened in solving the undifferential Lagrangian dual problems by subgradient algorithm, a subgradient algorithm based on fuzzy theory is presented. In this method, the resulting subgradient direction is attained by combining all history subgradient directions, which are achieved in the iteration process, following a simple membership function. The resulting subgradient direction uses the history information suitably, thereby significantly reduces the solution zigzagging difficulty without much additional computational requirements. The convergence of the algorithm is proved. This method is then applied in the traveling salesman problem, and the results show that this method leads to significant improvement over the traditional subgradient algorithm.

Key words: Lagrangian relaxation; subgradient algorithm; fuzzy theory; dual

1 引 言

在拉格朗日松弛算法(LR)中, 拉格朗日对偶问题的优化非常关键, 其解决方法通常采用次梯度算法 但它不是一种单调递减算法, 在优化过程中容易出现振荡现象, 效率较低 一般认为, 这是由于它的马尔可夫属性造成的[1,2], 即当前次梯度对以前迭代产生的历史次梯度没有记忆效应 对此, 已提出一些利用历史次梯度信息的算法: 文献[2]给出了利用所有历史次梯度信息时的收敛条件; [3]提出了正交次梯度算法, 其利用了上一次迭代的次梯度的信息; [4]则利用了所有的历史次梯度信息, 并采用一个二元二次优化问题选择权重, 但这极大地增加了计算复杂度

本文将模糊的概念引入次梯度的求取中, 提出了一种模糊次梯度算法 该算法利用隶属度函数给出所有历史次梯度的对应系数, 从而“模糊地”利用

收稿日期: 2003-12-16; **修回日期**: 2004-03-11.
基金项目: 国家自然科学基金资助项目(60174046).
作者简介: 周威(1977—), 男, 河南商丘人, 博士生, 从事供应链计划、供应链调度和协调等研究; 金以慧(1936—), 女, 浙江绍兴人, 教授, 从事流程工业的综合自动化、生产过程的建模与控制等研究

了所有历史次梯度信息. 证明了该算法的收敛性, 并给出了利用均衡 TSP 优化问题的仿真结果. 通过与传统次梯度算法的比较, 表明了该算法的有效性.

2 算法描述

对于一个带约束的最小化问题

$$(\mathrm{P}): \min f(x),$$
$$\text{s.t. } g(x) \le 0, x \in D.$$

其中: D 为一个紧集, f 和 g 分别为 D 上的连续凸函数, $g(x)$ 的维度为 $r \times 1$.

利用拉格朗日松弛方法, 可得到其对偶函数如下:

$$L(x, u) = f(x) + g^{\mathrm{T}}(x)u,$$
$$q(u) = \inf_{x \in D} L(x, u),$$
$$(\mathrm{D}): \quad \max q(u),$$
$$\text{s.t. } u \ge 0.$$

其中: $L(x, u)$ 为拉格朗日函数; $q(u)$ 为 $f(x)$ 的对偶函数, (D) 为原问题(P) 的对偶问题; u 为 $1 \times r$ 维向量称, 为拉格朗日算子, 当其确定时, $L(x, u)$ 是 x 的连续凸函数.

LR 优化方法通过求解对偶问题(D) 而逐步逼近原问题(P) 的最优解. 由于 $q(u)$ 一般是不可微函数, 在解决对偶问题(D) 时通常采用次梯度算法, 并在第 k 次迭代中采用下式更新算子:

$$u_k = u_{k-1} + t_k g_k. \tag{1}$$

其中: $g_k = g(x_k)$ 为第 k 次迭代时的次梯度, t_k 为第 k 次迭代时的步长.

次梯度算法简单可靠, 是求解不可微优化问题的最常用方法, 但该算法也经常出现振荡, 严重影响了其收敛效率. 为了解决这一问题, 人们分别从步长和次梯度方向两个方面进行了研究[1~5]. 在修改次梯度方向的策略上, 文献[2] 提出用历史次梯度的线性组合代替当前的次梯度, 即用

$$u_k = u_{k-1} + t_k d_k \tag{2}$$

代替式(1), 以克服次梯度算法的马尔可夫属性. 其中

$$d_k = \sum_{j=1}^{k} w_j^k g_j, \tag{3}$$

g_j 为第 j 次迭代时得到的次梯度, w_j^k 表示在第 k 次迭代中历史上第 j 个次梯度的权重系数. 但其中加权系数通常根据经验给出[1,5], 目前仍没有较好的解决方法. 基于文献[6] 提出的一种模糊确定 Bundle 算法中梯度权重的方法, 本文将模糊的概念引入到上述系数的确定中. 认为历史次梯度与当前所得次梯度方向越近, 则其包含的信息越多, 所被赋予的权重也越大. 基于上述思想, 给出定义如下(对于 $j = 1, \cdots, k$):

设第 j 次迭代得到的对偶值为

$$L(u_j, x_j) = \min_{x \in D} L(u_j, x) = g_j^{\mathrm{T}} u_j + c_j. \tag{4}$$

其中: $g_j = g(x_j)$, $c_j = f(x_j)$.

设对偶问题(D) 的最优解

$$L^* = L(u^*, x^*) = \max_{u \ge 0} \min_{x \in D} L(u, x) = g^{*\mathrm{T}} u^* + c^*. \tag{5}$$

其中: $g^* = g(x^*)$, $c^* = f(x^*)$.

则定义权重

$$w_j^k = \overline{w}_j^k \Big/ \sum_{j=1}^{k} \overline{w}_j^k. \tag{6}$$

其中 $\overline{w}_j^k$ 通过下面的隶属度函数定义:

$$\overline{w}_j^k = A(x_j) = \begin{cases} (L(u_k, x_k) + \epsilon - L(u_k, x_j))/\epsilon, \\ \quad 若 L(u_k, x_j) < L(u_k, x_k) + \epsilon 且 \\ \quad L(u_k, x_j) \le L^*; \\ 0, 其他. \end{cases} \tag{7}$$

其中

$$0 \le \epsilon \le (L^* - L(u_k, x_k))/a. \tag{8}$$

这里选择步长 t_k 满足

$$0 \le t_k \le \frac{2(a-1)(L^* - L(u_k, x_k))}{a \| d_k \|^2}, a > 1. \tag{9}$$

可以看出, 算法利用式(7) 中的隶属度函数给出每个历史次梯度合理的权重, 模糊地利用了历史次梯度的信息.

整个算法可以描述如下:

Step 1: 初始化: 取 $k = 0$, $u_0 \ge 0$;

Step 2: 在确定的 u_k 下优化对偶问题(D). 如果满足停止准则, 转 Step 5; 否则, $k = k + 1$;

Step 3: 利用式(6) ~ (8) 求取历史中各个次梯度的权重, 利用式(9) 求取步长;

Step 4: 利用式(2) 更新算子 u_k, 转 Step 2;

Step 5: x_k 和 u_k 分别为原问题和对偶问题的目标解; 结束.

设每次迭代时, 一般次梯度算法的计算复杂度为 $O(X)$, 利用模糊次梯度算法的计算量仅增加了 k 次, 其复杂度为 $O(X + k)$. X 由拉格朗日对偶问题的计算复杂度决定, 其值将视具体问题而定, 但通常远大于 k, 故每次迭代时算法的计算复杂度不变, 仍为 $O(X)$. 另外, 一般迭代次数 k 不超过 200, 基于模

糊的次梯度算法所增加的计算量非常小.

3　算法的收敛性

引理 1　对于 $\forall j = 1, 2, \cdots, k$, 如果满足 $L(u_k, x_j) \le L^*$, 则必有下式成立:

$$0 \le L^* - L(u_k, x_j) \le g_j^{\mathrm{T}}(u^* - u_k). \tag{10}$$

证明　由假设条件可知, $L^* - L(u_k, x_j) \ge 0$, 以下证明不等式的后半部分. 由式(4) 可得: $L(u_k, x_k) = \min_{x \in D} L(u_k, x)$, 因此必有

$$L(u_k, x_k) \le L(u_k, x_j). \tag{11}$$

对于

$$L^* = L(u^*, x^*) \le L(u^*, x_j), \tag{12}$$

由式(4) 和(12) 可得

$$L^* - L(u_k, x_j) \le L(u^*, x_j) - L(u_k, x_j) = g_j^{\mathrm{T}} u^* + c_j - (g_j^{\mathrm{T}} u_k + c_j) = g_j^{\mathrm{T}}(u^* - u_k). \quad \square$$

引理 1 说明, 在该算法中, 所选择的次梯度都与当前算子的收敛方向成"锐角".

引理 2　当 $a > 1$ 时, 有下式成立:

$$0 \le \frac{L^* - L(u_k - x_k)}{a} < d_j^{\mathrm{T}}(u^* - u_k). \tag{13}$$

证明　式(13) 前半部分明显成立, 下面主要证明其后半部分: 由 $L(u_k, x_j) < L(u_k, x_k) + \epsilon, \forall j = 1, 2, \cdots, k$ 和式(8), 可得

$$L(u_k, x_j) < L(u_k, x_k) + \epsilon \le L(u_k, x_k) + (L^* - L(u_k, x_k))/a,$$

$$L^* - L(u_k, x_j) > L^* - [L(u_k, x_k) + (L^* - L(u_k, x_k))/a] = (a - 1)(L^* - L(u_k, x_k))/a,$$

即

$$(a - 1)(L^* - L(u_k, x_k))/a < L^* - L(u_k, x_j).$$

由引理 1, 可得

$$\frac{(a - 1)(L^* - L(u_k, x_k))}{a} < L^* - L(u_k, x_j) \le g_j^{\mathrm{T}}(u^* - u_k).$$

由式(3) 和式(6), 则可得

$$\frac{(a - 1)(L^* - L(u_k, x_k))}{a} < \sum_{j=1}^{k} w_j^k g_j^{\mathrm{T}}(u^* - u_k) = d_k^{\mathrm{T}}(u^* - u_k). \quad \square$$

引理 2 说明, 当前通过线性组合得到, 次梯度方向仍然与算子的收敛方向成"锐角".

定理 1　利用该次梯度算法, 所得次梯度必然逐步收敛到最优次梯度方向. 即

$$\| u^* - u_{k+1} \| < \| u^* - u_k \|, \forall k. \tag{14}$$

证明

$$\| u^* - u_{k+1} \|^2 = \| u^* - u_k - t_k d_k \|^2 \le \| u^* - u_k \|^2 - 2 t_k d_k^{\mathrm{T}}(u^* - u_k) + t_k^2 \| d_k \|^2.$$

由引理 2 得

$$\| u^* - u_{k+1} \|^2 < \| u^* - u_k \|^2 - t_k \frac{2(a - 1)(L^* - L(u_k, x_k))}{a} + t_k^2 \| d_k \|^2 = \| u^2 - u_k \|^2 - t_k \left[\frac{2(a - 1)(L^* - L(u_k, x_k))}{a} - t_k \| d_k \|^2 \right].$$

由式(9), 得

$$\| u^* - u_{k+1} \|^2 < \| u^* - u_k \|^2. \quad \square$$

4　仿真结果

将本文算法应用于均衡的旅行商问题(TSP) 中整数规划的解决. 其描述如下:

$$\min \sum_{i=1}^{m} \sum_{j=1, j \ne i}^{m} c_{ij} x_{ij},$$

$$\text{s.t.} \sum_{j=1, j \ne i}^{m} x_{ij} + \sum_{j=1, j \ne i}^{m} x_{ji} = 2, i = 1, \cdots, m.$$

其中: i 和 j 表示节点; m 表示节点的个数; X 表示所有的一阶树的集合; c_{ij} 表示节点 i 与 j 之间的费用; 当节点 i 和 j 之间有连接时 $x_{ij} = 1$, 否则, $x_{ij} = 0$.

将问题约束松弛, 整理可得到对偶函数为

$$q(u) = -2 \sum_{i=1}^{m} u_i + \min \left\{ \sum_{i=1}^{m} \sum_{j=1, j \ne i}^{m} (c_{ij} + u_i + u_j) x_{ij} : x \in X \right\}$$

这里利用 XPRESS 软件, 分别处理了 33, 42 和 53 节点 3 种情况, 其中的数据是 1～ 10 之间均匀分布的随机数. 算法的停止准则为: gap $\le 0.1\%$, 其中 gap $= (L^* - L)/L^* \times 100\%$.

仿真中将一般次梯度算法(S) 与本文的模糊次梯度算法(FS) 进行了比较, 其中次梯度算法中利用

$$t_k = \gamma_k (L^* - L(u_k, x_k)) / \| g_k \|^2 \tag{15}$$

进行了步长的选择. 选择的参数为: $\gamma_k = 1, a = 2$.

在实际问题中, 直接获取真实最优解 L^* 是不现实的, 一般利用一个估计值替代该值, 并从理论上分析了估计值的偏差对计算结果的影响[1], 文献[7] 提出了一些无需预先了解最优值的次梯度算法. 本文的重点是对一般次梯度算法与提出的模糊次梯度算法进行比较. 为了便于计算和比较, 在仿真过程中, 首先利用 XPRESS 软件对问题进行计算, 并直接采用所得最好结果作为最优值 L^*.

所有计算都通过 PC 机完成, CPU 主频为 850

MHz, 内存 128 M. 最后的计算结果分别如表 1～表 3 所示. 表中的迭代次数反映了算法的振荡程度, 固定迭代次数得到的值反映了算法的收敛速度. 实验也考察了一般次梯度算法和模糊次梯度算法的计算时间, 但二者几乎没有差别, 因此没有列出.

从结果中可以看出, 模糊次梯度算法对振荡的抑制作用明显好于一般次梯度算法. 当迭代次数超过 20 次后, 模糊次梯度算法的收敛速度明显快于一般次梯度算法. 但在迭代次数为 10 时, 模糊次梯度算法的收敛速度并没有绝对优势, 有时甚至不如一

表 1　节点数为 33 时的情况

问　题	1		2		3		4		5	
最优解	150.61		128.382		159.867		142.017		149.012	
算法	S	FS	S	FS	S	FS	S	FS	S	FS
迭代次数	86	52	312	115	272	87	71	55	182	88
10	146.483	146.709	124.39	123.368*	155.972	156.146	138.468	138.574	142.831	143.524
20	148.153	148.78	125.918	126.277	157.758	158.371	140.269	140.895	145.170	145.978
30	148.877	149.168	125.858	126.768	158.038	158.754	140.828	141.182	146.741	147.469
40	148.927	149.673	126.133	127.169	158.254	158.846	141.169	141.455	147.207	147.856
50	149.797	150.110	127.098	127.386	158.442	159.122	141.503	141.884	147.348	148.024
75	150.375	—	126.971	127.953	158.601	159.599	—	—	147.957	148.620
100	—	—	127.428	128.155	159.059	—	—	—	148.372	—
125	—	—	127.662	—	159.201	—	—	—	148.652	—
150	—	—	127.791	—	159.434	—	—	—	148.683	—
175	—	—	127.751	—	159.397	—	—	—	148.817	—

表 2　节点数为 42 时的情况

问　题	1		2		3		4		5	
最优解	175.387		177.43		183.063		174.908		176.34	
算法	S	FS	S	FS	S	FS	S	FS	S	FS
迭代次数	129	52	126	66	94	52	60	41	72	48
10	170.431	170.995	171.928	172.981	176.442	177.452	169.300	168.910*	171.432	171.843
20	173.405	173.861	174.513	174.770	181.51	181.666	172.918	172.975	173.416	174.351
30	173.745	174.563	175.953	176.073	182.208	182.445	173.914	174.169	174.548	175.219
40	174.151	174.946	176.385	176.632	182.449	182.657	174.441	174.635	175.5	175.924
50	174.611	175.127	176.571	176.92	182.531	182.826	174.614	—	175.855	—
75	174.898	—	176.945	—	182.717	—	—	—	—	—
100	174.954	—	177.113	—	—	—	—	—	—	—
125	175.211	—	177.231	—	—	—	—	—	—	—
150	—	—	—	—	—	—	—	—	—	—
175	—	—	—	—	—	—	—	—	—	—

表 3　节点数为 53 时的情况

问　题	1		2		3		4		5	
最优解	219.314		222.459		207.226		233.896		216.189	
算法	S	FS	S	FS	S	FS	S	FS	S	FS
迭代次数	53	43	164	82	94	64	190	94	48	48
10	212.592	213.531	213.587	212.300*	200.211	201.547	225.981	227.169	213.535	213.619
20	217.534	217.554	217.868	218.665	204.226	204.689	230.738	230.975	215.484	215.551
30	218.204	218.679	219.767	220.575	205.394	206.030	232.12	232.411	215.829	215.924
40	218.718	219.009	220.679	221.48	206.284	206.662	232.793	233.222	215.896	—
50	219.021	—	221.18	221.853	206.847	206.917	233.174	233.275	—	—
75	—	—	221.777	222.192	206.992	—	233.231	233.526	—	—
100	—	—	221.865	—	—	—	233.292	—	—	—
125	—	—	222.012	—	—	—	233.403	—	—	—
150	—	—	222.121	—	—	—	233.258	—	—	—
175	—	—	—	—	—	—	233.598	—	—	—

般次梯度算法(表中 * 号处), 这是因为迭代步长的影响. 比较式(9)和式(15), 不难发现, 模糊次梯度在迭代步长 t_k 中引入了变量 a, 这可能造成步长的减小, 从而影响了个别情况下算法的短期收敛速度. 但这种情况并不多见(15 组数据中只出现 3 次), 综合起来, 本文提出的模糊次梯度算法在计算时间几乎不变的情况下, 收敛效果要明显优于一般次梯度算法.

5　结　　论

本文提出的模糊次梯度算法利用一个隶属度函数给出了历史次梯度的权重系数, 从而简单有效地利用了历史次梯度的信息. 仿真实验表明, 与传统的次梯度算法相比, 该算法在基本不增加计算时间的情况下, 可以更有效地抑制振荡现象, 其收敛效果明显优于一般的次梯度算法.

参考文献(References):

[1] Naum Z Shor. *Nondifferentiable Optimization and Polynomial Problems* [M]. Boston: Boston Kluwer, 1998.

[2] Kiwiel K C. An aggregate subgradient method for nonsmooth convex minimization [J]. *Mathematical Programming*, 1983, 27(3): 320-341.

[3] Camerini P M, Fratta L, Maffioli F. On improving relaxation methods by modified gradient techniques [J]. *Mathematical Programming Study*, 1975, 3: 26-34.

[4] Kim S, Ahn H. Convergence of a generalized subgradient method for nondifferentiable convex optimization [J]. *Mathematical Programming*, 1991, 50(1): 75-80.

[5] Francesca F. A modified subgradient algorithm for Lagrangean relaxation [J]. *Computers and Operations Research*, 2000, 28(1): 33-52.

[6] Zhao X, Luh P B. New bundle methods for solving Lagrangian relaxation dual problems [J]. *J of Optimzation Theory and Applications*, 2002, 113(2): 373-397.

[7] Kiwiel K C. The efficiency of subgradient projection methods for convex optimization, part I: General level methods [J]. *SIAM J Control and Optimization*, 1996, 34(2): 660-676.

(上接第 1212 页)

参考文献(References):

[1] Pawlak Z. *Rough Sets: Theoretical Aspects of Reasoning about Data* [M]. Amsterdam: Kluwer Academic Publishers, 1991.

[2] Bazan J. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision system [A]. *Rough Sets in Knowledge Discovery* [C]. Herdelberg: Phyisca-Verlag, 1998. 321-365.

[3] Narendra P, Fukunaga K. A branch and bound algorithm for feature subset selection [J]. *IEEE Trans on Computer*, 1977, 26 (9): 917-922.

[4] Liu H, Motoda H. *Feature Selection for Knowledge Discovery and Data Mining* [M]. Boston: Kluwer Acadcemic Press, 1998.

[5] 王国胤. Rough 集理论与知识获取[M]. 西安: 西安交通大学出版社, 2001.

[6] Lin M. *Software System for Intelligent Data Processing and Discovering Based on the Fuzzy-Rough Sets Theory* [M]. San Diego: San Diego University, 1995.

[7] 唐建国, 谭明术. 粗糙集理论中的求核与约简[J]. 控制与决策, 2003, 18(4): 449-452.
(Tang J G, Tan M S. On finding core and reduction in rough set theory [J]. *Control and Decision*, 2003, 18(4): 449-452.)

[8] 苗多谦, 王钰. 粗糙集理论中概念与运算的信息表示[J]. 软件学报, 1999, 10(2): 113-116.
(Miao D Q, Wang J. An Information representation of the concepts and operations in rough set theory [J]. *J of Software*, 1999, 10(2): 113-116.)

[9] 王国胤, 于洪, 杨大春. 基于条件信息熵的决策表约简[J]. 计算机学报, 2002, 25(7): 759-766.
(Wang G Y, Yu H, Yang D C. Decision table reduction based on conditional information entropy [J]. *Chinese J of Computers*, 2002, 25(7): 759-766.)

[10] Aha D, Bankert R. A comparative evaluation of sequential feature selection algorithms [A]. *5th Int Workshop on Artificial Intelligence and Statistics* [C]. New York: Springer, 1991.

[11] Cover T M. *Elements of Information Theory* [M]. New York: Wiley, 1991.

[12] Brenner N, Strong S P, Koberle R, et al. Synergy in a neural code [J]. *Neural Computation*, 2000, 13(7): 1531-1552.

[13] Yaglom A M, Yaglom I M. *Probability and Information* [M]. Hardbound: D Reidel Publishing Company, 1983.