第20卷 第10期 控 制 与 决 策 2005年10月
Vol. 20 No. 10 Control and Decision Oct. 2005

文章编号: 1001-0920(2005)10-1193-04

增加约束条件的线性规划问题递推算法研究

肖建华, 赵明旺
(武汉科技大学 信息科学与工程学院, 武汉 430081)

摘 要: 首先描述线性规划问题中约束条件增加时的递推求解问题, 此问题在线性规划问题中具有广泛的实际背景; 然后提出一个基于凸空间思想的快速求解此类问题的递推算法, 该算法能快速判断其矛盾约束、冗余约束以及新问题的递推最优解; 最后给出了该问题的一个算例, 实验仿真结果表明了该方法的有效性.

关键词: 线性规划; 矛盾约束; 冗余约束; 最优解; 递推算法
中图分类号: O221.1 **文献标识码**: A

Recursive Algorithm to Linear-programming Problems with Increase of Constraints

XIAO Jian-hua, ZHAO Ming-wang
(School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China. Correspondent: ZHAO Ming-wang, E-mail: zhaomw@public.wh.hb.cn)

Abstract: The recursively solving problem of linear programming with the increase of constraints is stated firstly, which has a wide practical background in linear programming. Then a recursive algorithm to judge quickly contradictory redundant constraining conditions and to solve the recursive problem is presented based on convex region. Finally a computational example shows the effectiveness of the presented recursive algorithm.

Key words: Linear programming; Contradictory constraining; Redundant constraining; Optimization solution; Recursive algorithm

1 引 言

不断增加约束时的线性规划(LP)问题的递推求解问题, 具有一定的理论研究意义和广泛的实际背景. 例如, 在系统与控制领域的建模中, L_1 和 L_∞ 辨识问题可转化为LP问题[1~3], 当需要应用在线辨识时, 每增加一组新的观测实验数据, 需要能在线递推求解新的辨识参数, 即相当于增加一个约束条件下递推求解新的LP问题. 类似地, 在曲线拟合、优化与决策等领域, 也存在类似的递推求解LP问题[2,3].

对增加约束条件后的新LP问题, 传统的分析方法通常首先判断原最优解是否满足其新增加的约束. 若满足, 则原最优解就是新LP问题的最优解; 若不满足, 则通过引进松弛变量和人工变量, 用对偶单纯形法重新调用LP算法来求解其新的最优解[4]. 在求解新的LP问题时, 未充分利用原LP问题的最优解信息以及求解过程中产生的中间结果(信息), 使得新问题求解的效率大大降低, 求解时间复杂度偏高. 与此同时, 传统的解决方法不能快速判断新增加的约束是否矛盾, 是否为冗余约束, 从而大大降低了求解问题的效率.

本文从基于凸空间的角度探讨和分析了增加约束条件对LP问题最优解的研究, 并给出了该问题的递推算法. 该算法能快速判断增加的约束是否为矛盾约束、冗余约束, 以及充分利用原问题的求解信息递推求解新问题. 本文的算法不仅易于编程实现, 而且适于手工计算. 仿真表明, 本文算法的效率是较高的.

收稿日期: 2004-10-25; **修回日期**: 2005-03-01.
作者简介: 肖建华(1979-), 男, 重庆人, 硕士生, 从事线性规划、系统辨识的研究; 赵明旺(1964-), 男, 广西富川人, 教授, 博士, 从事系统辨识与自适应控制、智能理论与智能控制的研究.

2 问题描述与讨论

2.1 问题描述

设原LP问题为

$$\max Z_m = CX,$$
$$\text{s.t.}: A_m X \le B_m,\ x_1, x_2, \cdots, x_n \ge 0. \tag{1}$$

其中: $C \in R^n$ 为价值向量; $B_m \in R^m$ 为资源向量; $X \in R^n$ 为决策变量向量; A_m 为约束条件的 $m \times n$ 维系数矩阵. 这里假设原LP问题中的约束是相容的, 且无冗余约束条件, 其最优解为 X_m, 最优目标值为 Z_m.

在原LP的问题中, 增加新的约束条件

$$\alpha^{\mathrm{T}}_{m+1} X \le b_{m+1}, \tag{2}$$

因此, 新的LP问题成为

$$\max Z_{m+1} = CX,$$
$$\text{s.t.} \begin{bmatrix} A_m \\ a_{m+1} \end{bmatrix} X \le \begin{bmatrix} B_m \\ b_{m+1} \end{bmatrix},$$
$$x_1, x_2, \cdots, x_n \ge 0. \tag{3}$$

在新问题中, 需判断其相容性, 去掉其中的冗余约束条件, 并基于第 m 步求出的最优解 X_m, 最优目标值 Z_m 及中间结果求解新的最优解 X_{m+1} 和最优目标值 Z_{m+1}.

2.2 问题讨论

根据新增加的约束是否为矛盾约束, 可将其分成以下两大类:

情形 1 矛盾约束. 新问题的可行解域为空, 无最优解, 如图 1 所示.

情形 2 非矛盾约束. 可分为 3 种情况:

情形 2.1 新增加的约束满足原最优解, 但相对新问题而言, 新增加的约束是冗余约束, 如图 2 所示;

情形 2.2 新增加的约束满足原最优解, 但相对新问题而言, 新增加的约束不是冗余约束, 如图 3 所示;

情形 2.3 新问题的最优解发生了变化, 需求解出该新最优解, 找出可能存在的冗余约束, 如图 4 所示.

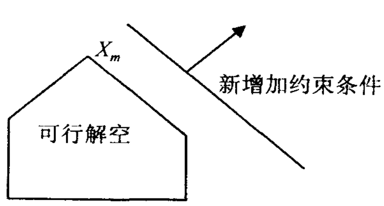

图 1 矛盾约束, 无最优解

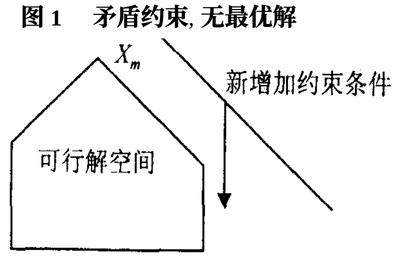

图 2 满足原最优解, 是冗余约束

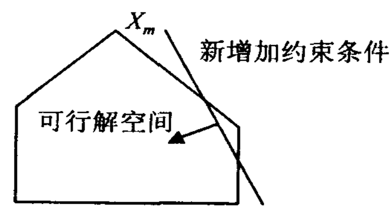

图 3 满足原最优解, 但不是冗余约束

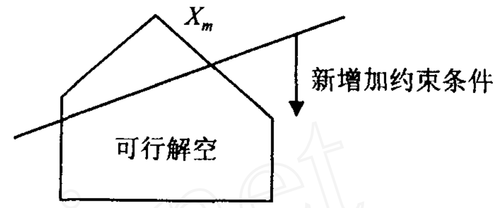

图 4 有新的最优解, 需删除冗余约束

由上述讨论可清楚地看出, 增加一个新的约束条件后, 可能使可行域减少, 也可能使可行域保持不变, 但绝不可能使可行域增大. 当新问题的可行域为空时, 问题无最优解, 新增加的约束条件为矛盾约束. 但当可行域非空时, 问题有最优解, 且新增加的约束不是矛盾约束.

3 算法思想

由数学理论可知, 线性不等式组的解空间是一个凸空间. 在LP中, 问题的可行解域是由问题的线性约束条件界定的凸多面体. 根据文献[5]判断凸空间是否为空的基本思想可知, 如果凸空间 P 非空, 则相应不等式所对应的超平面 $D_1, D_2, \cdots, D_m$ 中至少有一个(不妨设为 D_i) 与 P 的交非空, 事实上 D_i 与 P 的交非空的那部分为 P 的一个支撑面. 从而判断 P 是否为空只需依次判断 P 与 $D_1, D_2, \cdots, D_m$ 的交是否为空. 若 P 与 $D_1, D_2, \cdots, D_m$ 的交都是空, 则凸空间 P 为空, 否则凸空间 P 非空.

由上述讨论可知, 新增加一个约束将可能与原凸空间有 4 种关系. 将原最优解 X_m 代入新增加约束条件 L_{m+1} 中, 则可将上述的 4 种情况分成两大类:

1) 原最优解 X_m 满足新增加的约束的情形 2.1 和情形 2.2, 在情形 2.1 下, 通过判断新增加约束 L_{m+1} 所对应的超平面 D_{m+1} 与凸空间 P 的交为空, 从而判断出新增加的约束为冗余约束, 反之则是情形 2.2 的情况, 新增加的约束不是冗余的;

2) 原最优解 X_m 不满足新增加的约束的情形 1 和情形 2.3, 在这一大类问题中, 通过判断新增加约束 L_{m+1} 所对应的超平面 D_{m+1} 与凸空间 P 的交是否

为空, 从而判断出是否为矛盾约束, 若为空, 则新增加的约束为矛盾约束, 即情形 1, 反之则不是矛盾约束, 要求解出新的最优解 X_{m+1}, 并找出原约束条件中相对新问题中的冗余约束, 即情形 2 3.

在情形 2 3 情况下, 不难证明新的 LP 问题的最优解一定在新生成的顶点中取得. 通过比较目标值, 可很快判断出新 LP 问题的最优解.

4　算法描述

4.1　判断矛盾、冗余和求最优解的基本算法

下面给出利用凸空间的思想求取新问题的最优解的 CXG 算法, 描述如下:

Step 1: 将原最优解 X_m 代入新增加的约束 L_{m+1} 中, 若满足, 则原问题的最优解 X_m 等于新问题的最优解 X_{m+1}, 转 Step 2, 判定新增加的约束是否为冗余约束; 若不满足, 则转 Step 3, 判断其新增加的约束是否为矛盾约束.

Step 2: 取新增加的约束不等式 L_{m+1}, 使其对应等式 B_{m+1}. 将 B_{m+1} 代入 $L_1, L_2, \cdots, L_m$ 中联立化简, 采用文献[5] 中的算法判断其是否矛盾. 若矛盾, 则新增加的约束 L_{m+1} 为冗余约束; 否则, 新增加的约束不是冗余约束, 转 Step 4 找出原约束条件中的冗余约束后, 算法结束.

Step 3: 取新增加的约束不等式 L_{m+1}, 使其对应等式 B_{m+1}. 将 B_{m+1} 依次与原约束不等式 $L_1, L_2, \cdots, L_m$ 中的每一个不等式联立化简, 并采用文献[5] 中的算法判断其是否矛盾. 若存在矛盾常数不等式, 即不含有变量的不等式, 则新增加的约束为矛盾约束, 新 LP 问题无最优解, 算法停止; 否则新增加的约束不是矛盾约束, 转 Step 4 判断原约束条件中的冗余约束.

Step 4: 依次在 $L_1, L_2, \cdots, L_m$ 中取一个不等式 L_j, 得到其对应的等式 B_j, 将 B_j 依次与 $L_1, L_2, \cdots, L_m, L_{m+1}$ 中除 L_j 之外的其余不等式联立化简, 采用文献[5] 中的算法判断其是否矛盾. 若矛盾, 则 L_j 为新生成的约束不等式组中的冗余条件; 若不矛盾, 则该不等式 L_j 不是冗余.

Step 5: 求出新的顶点, 并计算各新顶点的目标值, 按目标值从大到小进行排序.

Step 6: 根据问题的目标函数, 求出新问题的最优解 X_{m+1} 和最优目标值 Z_{m+1}, 算法结束.

CXG 算法可用来判断新增加的约束是否为矛盾约束, 方便地求解出新问题的冗余约束及其他的最优解.

4.2　递推算法

为了充分利用上一轮求解出的最优解 X_m 和删除新问题中的冗余约束, 简化新问题的最优解的求解, 给出其递推算法如下:

Step 1: 初始化. 利用单纯形法或其他方法求解出该 LP 问题 $m = 1$ 时的最优解 X_m 和最优目标值 Z_m. 当然, 当 $m = 1$ LP 时问题无冗余约束, 且约束是相容的.

Step 2: 通过新观测数据, 形成一个新约束条件.

Step 3: 调用第 4.1 节提出的 CXG 算法, 判断新增加的约束. 如果新增加的约束是矛盾约束, 则算法停止; 如果新增加是冗余约束, 则转 Step 2; 如果该问题有新的最优解, 则转 Step 4.

Step 4: 利用 Step 3 判断是否为矛盾约束的同时, 求解出新问题的最优解 X_{m+1} 和最优目标值 Z_{m+1}.

Step 5: 调用 CXG 算法中判断是否为冗余约束的算法, 判断新问题中的冗余约束. 如果约束是冗余的, 则删除该约束. 删除所有的冗余约束后, 转 Step 6.

Step 6: 记录新问题的最优解 X_{m+1} 和最优目标值 Z_{m+1}, 并转 Step 2, 令 $m = m + 1$.

该递推算法在每增加一个约束条件时, 先判断是否为矛盾约束. 如果是矛盾约束, 则算法停止; 如果是冗余约束, 则不改变可行域的形状和大小, 不参与解的更新; 如果新增加的约束改变了 LP 问题的最优解, 则需求解出新问题的最优解, 并将其中的冗余约束删除.

5　算法示例

考虑如下 LP 问题:

$$\mathrm{Max}\ Z = 2x + 3y.$$
$$\mathrm{s.t.}\ L_1:\ -x + y \leq 2;$$
$$L_2:\ -x - y \leq -1,\ x, y \geq 0;$$
$$L_3:\ x \leq 2$$

该问题的最优解 $X_3 = (2, 4)$, 最优目标值 $Z_3 = 16$. 若新增加的约束为 L_4: $x + 3y \leq 3$, 将原最优解 $X_3 = (2, 4)$ 代入新增加的约束 $x + 3y \leq 3$. 经计算可知, 原最优解不满足新增加的约束条件.

由表 1 可知, 新增加的约束条件不是矛盾约束, 并知道 $(2, 1/3)$, $(0, 1)$ 是新生成的 2 顶点. 分别将其代入目标函数得目标 5 和 3, 故新问题的最优解

表 1　判断新增加的约束不是矛盾约束

		第 1 层	第 2 层
原约束条件	L_1: $-x + y \leq 2$	$F_{1,1}$: $4y \leq 5$	$F_{1,2}$: $4/3 \leq 5$
	L_2: $-x - y \leq -1$	$F_{2,1}$: $2y \leq 2$	$F_{2,2}$: $2/3 \leq 2$
	L_3: $x \leq 2$	$F_{3,1}$: $-3y \leq -1$	$F_{3,2}$: $3y = 1$
新增加的约束	L_4: $x + 3y \leq 3$	$B_{4,1}$: $x + 3y = 3$	L_4: $x + 3y \leq 3$ 不是矛盾约束

表 2 判断原约束条件中的冗余约束

	原约束条件	第 1 层之一	第 1 层之二	
第 1 层之一	L_1: $-x+y\le 2$ L_2: $-x-y\le -1$ L_3: $x\le 2$ L_4: $x+3y\le 3$	$B_{1,1}$: $-x+y=2$ $F_{2,1}$: $-2x\le 1$ $F_{3,1}$: $x\le 2$ $F_{4,1}$: $4x\le -3$	$B_{2,2}$: $-2x=1$ $F_{3,2}$: $-1/2\le 2$ $F_{4,2}$: $-2\le -3$(矛盾)	原约束条件 L_1: $-x+y\le 2$ 是冗余约束
第 1 层之二	L_1: $-x+y\le 2$ L_2: $-x-y\le -1$ L_3: $x\le 2$ L_4: $x+3y\le 3$	$F_{1,1}$: $-2x\le 1$ $B_{2,1}$: $-x-y=-1$ $F_{3,1}$: $x\le 2$ $F_{4,1}$: $2x\le 0$	$F_{1,2}$: $-4\le 1$ $B_{3,2}$: $x=2$ $F_{4,2}$: $-4\le 0$	原约束条件 L_2: $-x-y\le -1$ 不是冗余约束
第 1 层之三	L_1: $-x+y\le 2$ L_2: $-x-y\le -1$ L_3: $x\le 2$ L_4: $x+3y\le 3$	$F_{1,1}$: $y\le 4$ $F_{2,1}$: $-y\le 1$ $B_{3,1}$: $x=2$ $F_{4,1}$: $3y\le 1$	$F_{1,2}$: $-1\le 4$ $B_{2,2}$: $y=-1$ $F_{4,2}$: $-3\le 1$	原约束条件 L_3: $x\le 2$ 不是冗余约束

$X_{m+1}=(2,1/3)$, $Z_{m+1}=5$.

由表 2 可判断出原约束条件中 L_1: $-x+y\le 2$ 相对于新问题而言是冗余约束, 可以删除. 若再增加新的约束条件 L_5: $x-y\le -2$, 仿表 1 同样的方法判断可知该约束为矛盾约束, 算法停止.

6 结 语

关于不断增加约束条件的 LP 问题的研究, 对许多实际问题具有重要意义. 如何快速地判断新增加的约束是否为矛盾约束, 以及不矛盾的情况下如何快速判断约束条件中的哪些约束是冗余的, 均有利于对新的 LP 问题求解的简化. 本文提出的基于凸空间思想的算法有效地解决了该类问题, 并可利用算法的递推性将其应用于在线辨识等领域. 实验仿真表明, 该算法具有可行性和高效性.

参考文献(References)

[1] 黄圣乐. 用线性规划单纯形法在计算机上进行控制环节辨识[J]. *同济大学学报*, 1989, 17(2): 259-264.
(Huang S L. Identification of Controlled Processes with Linear Programming Simplex Method on Computer [J]. *Tongji University J*, 1989, 17(2): 259-264.)

[2] 王应明. 参数辨识的线性规划法及其快速实现[J]. *计算技术与自动化*, 1992, 11(2): 31-36.
(Wang Y M. Parameter Identification and its Fast Realization Based on Linear Programming [J]. *Computer Technology and Automation*, 1992, 11(2): 31-36.)

[3] 程小辉. *不确定性系统区间鲁棒辨识方法的研究*[D]. 北京: 北京工业大学, 2001: 40-46.
(Cheng X H. *Research of Interval Robust Indentification of System with Uncertainty* [D]. Beijing: Beijing University of Technology, 2001: 40-46.)

[4] 钱颂迪, 胡运权, 甘应爱, 等. *运筹学*[M]. 北京: 清华大学出版社, 1990: 62-64.
(Qian S D, Hu Y Q, Gan Y A, et al. *Operational Research* [M]. Beijing: Tsinghua University Press, 1990: 62-64.)

[5] 宋恩民, 黄文奇. 判断具有多线性约束条件的凸空间是否为空的交边算法[J]. *计算机学报*, 1996, 19(9): 704-708.
(Song E M. An Intersecting-side Algorithm to Determine Whether Convex Regions Bounded by Multiple Constraints are Empty [J]. *Chinese J Computer*, 1996, 19(9): 704-708.)

(上接第 1192 页)

参考文献(References)

[1] Zhang W, Branicky M S, Philips S M. Stability of Networked Control Systems[J]. *IEEE Control Systems Magazine*, 2001, 21(2): 85-99.

[2] Walsh G C, Beldiman O, Bushnell L G. Asymptotic Behavior of Nonlinear Networked Control Systems[J]. *IEEE Trans on Automatic Control*, 2001, 46(7): 1093-1097.

[3] Halevi Y, Ray A. Integrated Communication and Control Systems [J]. *J of Dynamic Systems Measurement and Control*, 1988, 110(4): 367-381.

[4] Luen Woei Liou, Asok Ray. A Stochastic Regulator for Integrated Communication and Control Systems[J]. *J of Dynamic Systems Measurement and Control*, 1991, 113(4): 604-617.

[5] Nilsson J, Bernhardsson B, Wittenmark B. Stochastic Analysis and Control of Real-time Systems with Random Time Delay[J]. *Automatica*, 1998, 34(1): 57-64.