doi:10.3969/j.issn.1001-4616.2018.01.005

一类特殊 0-1 二次规划问题解的必要条件和算法

陈　亮,徐玲玲

(南京师范大学数学科学学院,江苏 南京 210023)

[摘要]　研究一类特殊的 0-1 二次规划问题,其目标函数的系数矩阵为对称矩阵,所有元素均为 0 或 1 并且对角线元素相同,决策变量的和为给定的正整数.首先,给出一个最优解的必要条件. 然后,设计了一个高效的算法,可用于求解大规模的此类问题.

[关键词]　0-1 二次规划,对称 0-1 矩阵,极大值,必要条件

[中图分类号]O221.4　[文献标志码]A　[文章编号]1001-4616(2018)01-0022-04

The Necessary Conditions and the Algorithm for a Special Class of 0-1 Quadratic Programming Problem

Chen Liang,Xu Lingling

(School of Mathematical Sciences,Nanjing Normal University,Nanjing 210023,China)

Abstract:In this paper,we consider a special class of 0-1 quadratic programming problem. The coefficient matrix of its objective function is a 0-1 symmetric matrix with the same element in the diagonal,and the sum of decision variable is a given positive integer. We present a necessary condition for the optimal solution,and design an efficient algorithm that can be used to solve large-scale problem.

Key words:0-1 quadratic programming,0-1 symmetric matrix,maximum value,necessary condition

0-1 二次规划(binary quadratic programming,BQP)问题是一类特殊的二次规划问题,它的目标函数是二次函数,决策变量只能取 0 或者 1.0-1 二次规划问题在很多领域有着广泛应用,如计算机辅助设计、社会心理学、交通管理、金融分析等.

本文主要考虑一类特殊的 0-1 二次规划问题,其目标函数的系数矩阵为对称矩阵,所有元素均为 0 或 1 并且对角线元素相同,决策变量的和为给定的正整数.即:

$$\begin{aligned} &\max\ \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} \\ &\text{s.t.}\ \sum_{i=1}^{n} x_i = m,\quad x_i \in \{0,1\}. \end{aligned} \tag{1}$$

式中,$\boldsymbol{A} \in \{0,1\}^{n\times n}$是一个对角线元素相同的对称 0-1 矩阵,$m$ 是正整数.

这类问题在图论中有着广泛的应用,例如,如何在图中找一个 k 阶子图,使其边数达到最多;求解最大团问题(maximum clique problem,MCP)的一个子问题,所谓团就是在一个子图中,任意两个顶点均有边相连,最大团就是图中含顶点数最多的团.最大团问题是图论中一个经典的组合优化问题,也是一类 NP 完全问题,目前在国际上已有广泛的研究,而国内学者对 MCP 问题的研究则还处于起步阶段.本文的算法可以为判断是否含有 k 阶团提供理论保证和算法实现,算法的思想主要是同时进行 Best in worst out.

下文将给出最优解的必要条件,然后给出数值结果. 若无特殊说明,文中 $\boldsymbol{x}$ 表示列向量,$\boldsymbol{A}$ 表示矩阵.

1　最优解的必要条件

若将 $\boldsymbol{x}$ 中分量为 1 的下标集合记为 S,$\boldsymbol{A}$ 中对应于集合 S 中所有下标的行和列得到的矩阵称为 $\boldsymbol{A}$ 对

收稿日期:2016-11-18.

基金项目:国家自然科学基金(11401314)、江苏省高校自然科学基金(13KJD110007).

通讯联系人:徐玲玲,博士. 研究方向:计算数学. E-mail:xulingling@ njnu.edu.cn

应于 $\boldsymbol{x}$ 的主子矩阵或者称对应于集合 S 的主子矩阵,则不难看出向量 $\boldsymbol{x}$ 对应的目标函数值等于 $\boldsymbol{A}$ 对应于向量 $\boldsymbol{x}$ 的主子矩阵中 1 的个数. 下面我们给出最优解的必要条件.

定理 1　设 $V=\{1,2,\cdots,n\}$, $S=\{k \mid x_k=1\}$, $P=V\backslash S$, $x_i=\begin{cases}1, & i\in S\\ 0, & i\notin S\end{cases}$,那么 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$ 是问题(1)的解的必要条件是 $\boldsymbol{A}$ 对应于 $\boldsymbol{x}$ 的主子矩阵列和最小的列号 i_r,在 $\boldsymbol{A}$ 的 $S\backslash\{i_r\}$ 行 $P\cup\{i_r\}$ 列中,i_r 列列和最大.

证明　不妨设问题(1)的解 $\boldsymbol{x}=(1,1,\cdots,1,0,\cdots,0)^{\mathrm{T}}$ 前 m 个分量为 1,后 $n-m$ 个分量为 0,此时 $S=\{1,2,\cdots,m\}$, $P=\{m+1,m+2,\cdots,n\}$,则 $\boldsymbol{A}$ 对应于 $\boldsymbol{x}$ 的主子矩阵为 $\begin{bmatrix}a_{11} & \cdots & a_{1m}\\ \vdots & \ddots & \vdots\\ a_{m1} & \cdots & a_{mm}\end{bmatrix}$,再设列和最小的列号为 m.

此时 $\boldsymbol{A}$ 的 $S\backslash\{m\}$ 行 $P\cup\{m\}$ 列为 $\begin{bmatrix}a_{1,m} & \cdots & a_{1,n}\\ \vdots & \ddots & \vdots\\ a_{m-1,m} & \cdots & a_{m-1,n}\end{bmatrix}$,若列号为 m 的列列和不是最大的,则存在 t 列列和大于 m 列列和,只需令 $x_m=0$, $x_t=1$,记作 y,则 $\sum\limits_{i=1}^{m-1} a_{im}<\sum\limits_{i=1}^{m-1} a_{it}$.

那么

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=\sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij}=\sum_{i=1}^{m-1}\sum_{j=1}^{m-1} a_{ij}+\sum_{i=1}^{m-1} a_{im}+\sum_{j=1}^{m-1} a_{mj}+a_{mm}=\sum_{i=1}^{m-1}\sum_{j=1}^{m-1} a_{ij}+2\sum_{i=1}^{m-1} a_{im}+a_{mm}<$$
$$\sum_{i=1}^{m-1}\sum_{j=1}^{m-1} a_{ij}+2\sum_{i=1}^{m-1} a_{it}+a_{tt}=\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y},$$

因此 $\boldsymbol{x}$ 不是最优解,矛盾.

现举例加以说明.

例 1

$$\boldsymbol{A}=\begin{bmatrix}1 & 0 & 1 & 1 & 0\\ 0 & 1 & 1 & 0 & 1\\ 1 & 1 & 1 & 0 & 0\\ 1 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 1\end{bmatrix}, m=3.$$

由观察或枚举法易知 $\boldsymbol{x}_0=(0,1,1,0,1)^{\mathrm{T}}$ 是问题(1)的解,$S=\{2,3,5\}$, $P=\{1,4\}$,对应的主子矩阵为

$\begin{array}{cc} \begin{array}{ccc} 2 & 3 & 5\end{array} & \\ \begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 0\\ 1 & 0 & 1\end{bmatrix} & \begin{array}{c}2\\3\\5\end{array}\end{array}$,列和最小的是 2,3 对应于原矩阵就是 3,5.

当 $i_r=3$ 时,$\boldsymbol{A}$ 的 $S\backslash\{3\}$ 行 $P\cup\{3\}$ 列是 $\begin{array}{cc} \begin{array}{ccc} 1 & 3 & 4\end{array} & \\ \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix} & \begin{array}{c}2\\5\end{array}\end{array}$,则列号为 3 的列列和最大. 当 $i_r=5$ 时,$\boldsymbol{A}$ 的 $S\backslash\{5\}$ 行 $P\cup\{5\}$ 列是 $\begin{array}{cc} \begin{array}{ccc} 1 & 4 & 5\end{array} & \\ \begin{bmatrix}0 & 0 & 1\\ 1 & 0 & 0\end{bmatrix} & \begin{array}{c}2\\3\end{array}\end{array}$,则列号为 5 的列列和是最大之一.

2　迭代算法

根据证明的思路,我们设计了一个求解最优解的算法:将迭代过程中产生的点中的某个分量 1 进行替换,使得替换后得到的点对应的目标函数值增加,进而得到一个不断更新最优值的算法,具体如下:

算法:

输入　初始集合 $S=\{i_1,i_2,\cdots,i_m\}$ 及矩阵 $\boldsymbol{A}$.

输出　近似最优值 f 和近似解集合 S.

Step 1:找出 $\boldsymbol{A}$ 对应于 S 的主子矩阵所有列和最小的列 i_r,标记符 $tag=0$;

Step 2:逐个判断 i_r 是否为 $\boldsymbol{A}$ 的 $S\backslash\{i_r\}$ 行 $P\cup\{i_r\}$ 列中列和最大,如果不是,$tag=1$.选出一个列和更大的 i_t(可以选取列和最大的 i_t,如果个数多于一个,则任取其中一个),令 $S=(S\backslash\{i_r\})\cup\{i_t\}$.

Step 3:如果 $tag=0$,则每个 i_r 均满足必要条件,转 Step4;否则返回 Step1.

Step 4:输出 S 以及 $\boldsymbol{A}$ 对应于 S 的主子矩阵的所有元素之和.

下面应用例 1 的数据来说明算法的迭代过程.

例 2　选取初始值 $S=\{3,4,5\}$.

(1)$\boldsymbol{A}$ 对应于 S 的主子矩阵 $\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$,列号为 3,4,5 列列和均为最小,$tag=0$;

(2)当 $i_r=3$ 时,$\boldsymbol{A}$ 的 4,5 行 1,2,3 列 $\begin{bmatrix}1&0&0\\0&1&0\end{bmatrix}$,1,2 列列和均比第 3 列大,选取第 1 列替换第 3 列,此时 $S=\{1,4,5\}$,$tag=1$;

(3)当 $i_r=4$ 时,$\boldsymbol{A}$ 的 1,5 行 2,3,4 列 $\begin{bmatrix}0&1&1\\1&0&0\end{bmatrix}$,三列列和相同,$S$ 不变;

(4)当 $i_r=5$ 时,$\boldsymbol{A}$ 的 1,4 行 2,3,5 列 $\begin{bmatrix}0&1&0\\0&0&0\end{bmatrix}$,3 列列和大于 5 列,3 列替换 5 列,此时,$S=\{1,3,4\}$,$tag=1$;

(5)$tag=1$,故重新检查 $\boldsymbol{A}$ 对应于 S 的主子矩阵 $\begin{bmatrix}1&1&1\\1&1&0\\1&0&1\end{bmatrix}$,列号为 3、4 列列和均为最小,$tag=0$;

(6)当 $i_r=3$ 时,$\boldsymbol{A}$ 的 1,4 行 2,3,5 列 $\begin{bmatrix}0&1&0\\0&0&0\end{bmatrix}$,3 列列和最大,$S$ 不变;

(7)当 $i_r=4$ 时,$\boldsymbol{A}$ 的 1,3 行 2,4,5 列 $\begin{bmatrix}0&1&0\\1&0&0\end{bmatrix}$,2,4 列列和最大且相等,$S$ 不变;

(8)$tag=0$,循环结束,输出 $S=\{1,3,4\}$,结果为 7.

注 1　每次迭代后函数值一定会上升,又由于函数值有上界,因此,循环一定是有限步终止.

注 2　如果矩阵 $\boldsymbol{A}$ 通过交换行和列能够化成准对角阵,那么适合对每一个对角块进行该算法,要求每个对角块的阶数均不低于 m,否则在解集跨越多个对角块时,会使解最终收敛到某一对角块中,而使得解有所遗漏,如:

$$\boldsymbol{A}_1=\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix},\boldsymbol{A}_2=\begin{bmatrix}1&1&1&0\\1&1&0&1\\1&0&1&1\\0&1&1&1\end{bmatrix},\boldsymbol{A}=\begin{bmatrix}\boldsymbol{A}_1&0\\0&\boldsymbol{A}_2\end{bmatrix},\quad m=3.$$

显然,最优解是 $S=\{1,2,3\}$,但是通过该算法,只要初值中只有一个数落在 $S=\{1,2,3\}$ 中,最优解就会收敛到 $\{4,5,6,7\}$ 的子集中去.

3　数值结果

我们通过一些数值例子来说明算法的可行性和有效性.取

$$\boldsymbol{A}=(a_{ij})_{1000\times1000},其中\ a_{ij}=\begin{cases}1, & i=j,\\0, & i+j\ 是\ 3\ 的倍数且\ i\neq j,\\1, & i+j\ 不是\ 3\ 的倍数且\ i\neq j.\end{cases}$$

矩阵通过同时交换行和列不能化成准对角矩阵.

在 MATLAB 环境下编程实现求解过程,计算结果见表 1.

表 1　MATLAB 运行结果

Table 1　Operation result by MATLAB

m	初值	迭代次数	函数值	时间/s	是否达到最优值
200	$\{1,2,\cdots,200\}$	66	20 200	5.89	否
	$\{1,6,11,16,21,26,\cdots,996\}$	67	40 000	6.76	是
	列和从大到小排名前 200 的列标	1	20 200	1.41	否
500	$\{1,2,\cdots,500\}$	167	125 500	77.41	是
	$\{1,3,5,7,\cdots,999\}$	29	12 500	18.21	否
	列和从大到小排名前 500 的列标	1	125 500	4.94	是
800	$\{1,2,\cdots,800\}$	134	240 800	221.54	是
	$\{1,2,\cdots,1\,000\}\setminus\{1,6,11,16,\cdots,996\}$	134	240 800	229.12	是
	列和从大到小排名前 800 的列标	0	240 800	1.16	是

由计算结果我们猜测:

(1)能够达到最值解的概率与 m、n 的比值有关,比值越接近 1,最值解对初值的选取依赖程度越低. 相反,比值越接近 0,最值解对初值的依赖程度越高.

(2)迭代步数与计算时间也与初值有关,选取列和从大到小排名前 m 的列标迭代速度更快,但当 m、n 比值较小时,这样选取不易达到最优值.

对这两个观察得到的结果,目前我们无法提供理论证明.

4　结论

本文考虑了一类特殊的 0-1 二次规划问题,我们提出了一个最优解的必要条件,并设计了一个高效的算法,可以适用于大规模的数值问题,给出最优解.该算法的优点是:算法简单、高效,比较大小的次数是 $o(n^3)$,不涉及其余复杂算法,选取较好的初值,对于变量维数较高的情况有着较快的速度.缺点是极值解与初值相关,并且只能保证解是极值解,不能保证最终的解是最优解.但是在只需要近似求其最值解的情况下,可以不断随机改变初值,反复使用该算法,找出使目标函数达到最大的解,这个解是最优解的近似甚至就是最优解.我们将在以后的工作中对算法的改进作进一步研究.

[参考文献]

[1]　陈伟. 0-1 二次规划的全局最优性条件及算法[D]. 上海:上海大学,2005.

[2]　张爱君,秦新强,龚春琼. 求解 0-1 二次规划问题的迭代禁忌搜索算法[J]. 计算机工程,2012,38(1):140-142.

[3]　周光明,王奇生,邓康. 带线性约束 0-1 二次规划罚参数的改进[J]. 南华大学学报(理工版),2004,18(1):67-69.

[4]　王青松,范铁生. 低度图的最大团求解算法[J]. 计算机工程,2010,36(6):39-41.

[5]　周旭东,王丽爱,陈峻. 启发式算法求解最大团问题研究[J]. 计算机工程与设计,2007,28(18):4 329-4 332.

[责任编辑:陈　庆]