文章编号：1000-8152(2002)02-0183-05

一类瓶颈多选择整数规划问题及其求解*

郭 伟 席裕庚
(上海交通大学自动化研究所·上海,200030)

摘要：提出了一类瓶颈多选择整数规划问题(BMCIP)并给出了一种有效解法.在改进的启发式群局部搜索的基础上,利用分枝剪枝法得到全局最优解.作为仿真算例,将ATM网络中虚通道路由规划(VPR)问题转化为此类问题进行求解,并与传统的搜索算法进行了比较.结果表明这种算法是快捷而有效的.

关键词：多选择整数规划；瓶颈问题；局部搜索；分枝剪枝
文献标识码：A

A Kind of Bottleneck Multiple Choice Integer Program Problem and Its Solution Method

GUO Wei and XI Yugeng
(Institute of Automation, Shanghai Jiaotong University · Shanghai, 200030, P.R.China)

Abstract: This paper proposes a kind of bottleneck multiple choice integer program (BMCIP) problem and presents an effective solving method. On the basis of improved heuristic cluster local search, branch and cut method is used to get the global optimal solution. As a simulation example, the route program problem of virtual path (VPR) in ATM networks is transformed into this kind of problem. Comparison results between this method and the traditional search methods show that the algorithm is speedy and effective.

Key words: multiple choice integer program; bottleneck problem; local search; branch and cut

1 引言(Introduction)

多选择整数规划(MCIP)是一种变量被划分到不同集合的线性二进制规划.对于任何一个可行解,每一个划分的变量集合中有且只能有一个变量为1、其它的全部为0.其目标是在满足一般线性约束的情况下最小化线性目标函数.它有广泛的应用背景,指派问题、多选择背包问题、广义指派问题等都可看作此问题的特例[1].

在常见的问题如资源分配中,线性规划模型的性能函数往往是最小化(最大化)各线性项的和.这样做的坏处在于极值解最小化单一总目标(总代价),往往是以不成比例地影响单个行为为代价的.这样的解可能被决策者认为是不理想的和难以实现的.而且,虽然输入参数的微小变化只导致最优目标函数值的微小变化,但却可能导致极值点有完全不同的行为.在许多实际问题中,如网络带宽设计,合适的目标函数不是代价的总和,而是单个代价的最大值,包括整数分配问题[2]和广义指派问题[3].

本文提出了一种特殊的、但却是工业调度、网络资源优化等问题中常见的一类minimax(瓶颈)目标函数.问题具有多选择整数规划问题的约束条件.本文把此类问题归于瓶颈多选择整数规划问题(BMCIP),给出了一种精确解法,并给出了传统搜索算法解决同样问题的结果比较.

2 瓶颈多选择整数规划问题(BMCIP)数学描述(The mathematic description of the bottleneck multiple choices integer program problem)

BMCIP性能指标函数

$$\min_{x_{ij}} \max_{k} XRCL_k \tag{1}$$

约束条件

$$\sum_{j=1}^{n_i} x_{ij} = 1,\ 1 \leqslant i \leqslant m, \tag{2}$$

$$x_{ij} \in \{0,1\},\ 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n_j. \tag{3}$$

其中, m 是划分的变量集合(广义上限集合, GUB-

* 基金项目:国家973(G1998030415)资助项目.
收稿日期:2000-06-26; 收修改稿日期:2001-03-02.

sets)的数目，n_i 是第 i 个 GUB 集合中变量的个数，x_{ij} 是第 i 个 GUB 集合中第 j 变量，是一二进制变量.(2),(3)式迫使 $\{x_{ij}\}_{j=1}^{n_i}$ 中有且只能有一个变量为1.

X 是一包含所有变量的 n 维($n=\sum_{i=1}^{m}n_i$)选择向量

$$X=[x_{11},x_{12},\cdots,x_{1n_1},x_{21},x_{22},\cdots,x_{2n_2},\cdots,x_{m1},x_{m2},\cdots,x_{mn_m}]. \tag{4}$$

R 是代价权值常矩阵，是一 n 维正实对角阵，

$$R=\mathrm{diag}(\underbrace{r_1,r_1,\cdots,r_1}_{n_1},\underbrace{r_2,r_2,\cdots,r_2}_{n_2},\cdots,\underbrace{r_m,r_m,\cdots,r_m}_{n_m}). \tag{5}$$

C 是 $n\times l$ 维常数矩阵

$$C=\begin{bmatrix} c_{111} & c_{112} & \cdots & c_{11l}\\ \cdots & \cdots & \cdots & \cdots\\ c_{1n_11} & c_{1n_12} & \cdots & c_{1n_1l}\\ \vdots & \vdots & \vdots & \vdots\\ c_{m11} & c_{m12} & \cdots & c_{m1l}\\ \cdots & \cdots & \cdots & \cdots\\ c_{mn_m1} & c_{mn_m2} & \cdots & c_{mn_ml}\end{bmatrix}. \tag{6}$$

其中

$$c_{ijk}\in\{0,1\},1\leqslant i\leqslant m,1\leqslant j\leqslant n_i,1\leqslant k\leqslant l. \tag{7}$$

表示第 i 个 GUB 集合中第 j 个变量所代表的方案或行为是否占用 l 类资源中的第 k 类.这样，l 维向量

$$XRC=[\sum_{i=1}^{m}r_ic_{ib_i1}\ \cdots\ \sum_{i=1}^{m}r_ic_{ib_il}] \tag{8}$$

表示各个 GUB 集合选中变量后各类资源总的负载，下标中的 b_i 表示第 i 个 GUB 集合选中第 b_i 个变量.L_k 定义为

$$L_k=[\underset{1}{0}\ \cdots\ \underset{k-1}{0}\ \underset{k}{1}\ \underset{k+1}{0}\ \cdots\ \underset{l}{0}]^{\mathrm T},1\leqslant k\leqslant l. \tag{9}$$

这样，$XRCL_k$ 表示第 k 类资源的负载总和.

解决上述问题的本质是从每一个多选择域(GUB-set)中选择正确的变量.找到正确的选择需要检查 $\prod_{i=1}^{m}n_i$ 种组合，而这是随 n 指数增长的.MCIP 是 NP-hard，因为它的特例之一，广义指派问题是 NP-hard[1].同样 BMCIP 是 NP-hard，因为它的特例之一，瓶颈广义指派问题是 NP-hard[3].

3 局部搜索算法(Local search method)

局部搜索是求解组合优化问题的一种通用的、有效的方法[4]，是最早提出来的解决组合优化问题中具有计算复杂性的 NP 完全问题的为数不多的有效方法之一.由于在求解一些著名的 NP 完全问题如：n-皇后问题、旅行商问题、图划分问题、SAT 问题取得成功而引起了广泛的注意.

给定一优化问题的目标函数 f 和可行域 R，典型的局部搜索优化算法[4]需要每一个解 $x_i\in\mathbb{R}$ 有一个预先定义的邻域 $N(x_i)\in\mathbb{R}$、从邻域的集合中寻找 x_{i+1}，使得 $f(x_{i+1})<f(x_i)(f(x_{i+1})>f(x_i))$.如果这样一个解存在，它就变成当前解，一直迭代下去.否则 x_i 就是相对于 $N(x_i)$ 的局部极值点.

对于本文问题，标准的局部搜索算法如下：

步骤 3.1 编码.问题共有 $n(n=\sum_{i=1}^{m}n_i)$ 个变量，原搜索空间为 2^n，是变量个数的指数.现以一 m 维向量表示解，即维数等于多选择域的个数.解的每一位对应一个 GUB 集合.该位的取值为该集合中取值为1的变量的索引值，该集合中其它的变量取值为0.这样，解的第 i 位可能从 $\{1,\cdots,n_i\}$ 中取任一值.其中，n_i 为多选择集合 i 中的变量个数.例如：$m=2,n_i=3,i=1,2$.一个满足多选择约束的解是 $x_{12}=x_{23}=1$，其它的4个变量都为0.解就可以表示为[2,3].

步骤 3.2 初始解.随机选择初始解.初始解 x_0 第1位到第 m 位随机赋值，第 i 位从1到 n_i 随机取一整数.

步骤 3.3 邻域设计.给定一个具体问题的可行点 $x\in\mathbb{R}$，通常把某种意义上靠近 x 的点的集合定义为 $N(x)$.在组合优化中 N 的确定通常依赖于问题的结构.在 BMCIP 问题中定义解的邻域为：$N(x)=\{x_{i+1}:x_{i+1}\in\mathbb{R},|x_{i+1}-x_1|=1\}$，$|x_{i+1}-x_i|$ 定义为两个解不同元素的数目.

步骤 3.4 解码.利用编码过程建立的一个解 x 和变量的对应关系得到(1)式中 X.解 x 的评价函数为 $f(x)=\max\limits_{k}XRCL_k$.

步骤 3.5 搜索过程.对于当前解 x_i，随机选取对其第 $j(1\leqslant j\leqslant m)$ 位操作，第 j 位随机从1到 n_j 取一整数，得到 x_i 的一邻域点 x_{i+1}.解码得到 $f(x_i)$ 和 $f(x_{i+1})$.如果 $f(x_{i+1})<f(x_i)$，则 $x_i=x_{i+1}$；否则，x_i 不变，本过程迭代进行.

步骤 3.6 停止规则.如果在指定迭代次数内，x_i 不发生变化，$f(x_i)$ 得不到改进，则停止搜索过程.以当前解作为算法结果.

4 改进的启发式群局部搜索算法(Improved heuristic cluster local search method)

局部搜索简便易行、计算复杂度小，很容易收敛

到一个局部最优解.但得到的结果性能不稳定,而且往往不能令人满意.究其原因,一是搜索过程是单线程的,邻域搜索的本质使得最终结果过于依赖初始点.二来搜索过程是“贪婪”的,对于当前解 x_i 和其邻域点 x_{i+1},如果 $f(x_{i+1}) > f(x_i)$,则轻易抛弃 x_{i+1}.使得搜索过程易陷入局部最优解,一直拘于局部最优解的邻域范围.因此本文提出了一种改进的启发式群局部搜索算法,利用混沌运动特性得到一组分布好的初始点,每一个初始点开始一个独立的搜索过程.搜索过程中不轻易抛弃邻域点,而是在邻域点的基础上,作进一步的试探.同时,搜索过程中采取启发式规则,减少迭代次数,每次迭代中减少计算量.最后在多个搜索过程所得的多个局部最优点中选取最好的一个.

算法步骤如下:

步骤 4.1 编码.同步骤 3.1.

步骤 4.2 初始解.本文采用群局部搜索,多个搜索过程同时进行,自然希望多个初始点具有遍历性、随机性.本文利用混沌运动遍历性、随机性的特性[5]得到初始点.搜索过程 1 到 POP 有各自的参照点

$$u_{j+1} = \mu u_j(1 - u_j),\ 1 \leqslant j < \mathrm{POP}. \qquad (10)$$

第 j 个搜索过程初始解的第 i 位

$$x[i] = \operatorname{mod} f\,(1 + u_j(n_i - 1)). \qquad (11)$$

其中,$\operatorname{mod} f\,(\cdot)$ 为取整操作.

步骤 4.3 邻域设计同步骤 3.3.

步骤 4.4 解码同步骤 3.4.

步骤 4.5 各搜索过程独自搜索.为克服局部搜索易陷入局部最优解的缺点,各搜索过程采用如下流程:

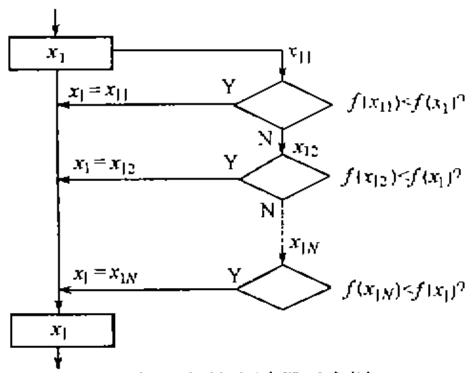

图 1 各搜索过程示意图

Fig. 1 Every search processes' sketch map

其中 x_{11} 是 x_1 的一邻域点,x_{12} 又是 x_{11} 的一邻域点,依次类推.这样,使搜索过程以螺旋式方式进行,跳出局部最优解的概率大大增加.

步骤 4.6 启发式规则.可以随机从邻域中选择解,但把一些启发式信息加入邻域的搜索中,可以有效地改善搜索的效率.针对 BMCIP 问题的特点,只有占用负载最大的资源的变量发生改变,才能改善解的性能.因此本文采用了为经过负载最大的资源的变量所在的 GUB 集合,随机选择新的变量的邻域搜索方式.

设对于当前解 $x = [b_1 \ \cdots \ b_i \ \cdots \ b_m]$,$\max\limits_k XRCL_k = XRCL_p$,$(1 \leqslant k \leqslant l)$,且 C 矩阵中 $c_{ib_ip} = 1$,表示当前解第 i 个 GUB 集合选中的第 b_i 个变量占用当前负载最大的第 p 资源.设其邻解点 x' 改变的是第 i 个元素(第 i 个 GUB 集合中选择的变量发生改变),选中的是第 i 个 GUB 集合中第 b_i' 个变量,$x' = [b_1 \ \cdots \ b_i' \ \cdots \ b_m]$.对于当前解 x,求得满足 $XRCL_p - XRCL_k < r_i$ 的所有的 k,$(1 \leqslant k \leqslant l)$,$r_i$ 为(5)式中第 i 个 GUB 集合对应的代价系数.对于这些 k,如果(6)式 C 矩阵中有任一 $c_{ib_i'k} = 1$,就可判断 $f(x') \geqslant f(x)$;完全是0就可判断 $f(x') < f(x)$.

例如,对于当前解 x 对应的 X,设 XRC 中最大的元素为 $XRCL_p$,而且有且只有

$$XRCL_p - XRCL_2 < r_i,$$
$$XRCL_p - XRCL_3 < r_i,$$
$$XRCL_p - XRCL_p = 0 < r_i.$$

其中,$XRC = [XRCL_1 \ \ XRCL_2 \ \ XRCL_3 \ \ XRCL_4 \ \cdots \ XRCL_p \ \cdots \ XRCL_l]$.那么 C 矩阵中,

$$C = \begin{bmatrix} C_{111} & C_{112} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & C_{11e} \\ \cdots & \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots & \cdots \\ c_{ib_i'1} & c_{ib_i'2} & c_{ib_i'3} & c_{ib_i'4} & \cdots & & c_{ib_i'p} & \cdots & c_{ib_i'l} \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ c_{ib_i1} & c_{ib_i2} & c_{ib_i3} & c_{ib_i5} & \cdots & & c_{ib_ip} = 1 & \cdots & c_{ib_il} \\ \cdots & \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots & \cdots \\ C_{mn_m1} & C_{mn_m2} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & C_{mk_ml} \end{bmatrix}.$$

如果 $c_{ib_i'2}, c_{ib_i'3}, c_{ib_i'p}$ 任一为 1,对于解 x' 对应的 X',必有

$$X'RCL_2 > XRCL_p,$$
$$X'RCL_3 > XRCL_p,$$
$$X'RCL_p > XRCL_p.$$

从而 $\max\limits_k X'RCL_k > XRCL_p (1 \leqslant k \leqslant l)$,即 $f(x') \geqslant f(x)$.即 $c_{ib_i'2}, c_{ib_i'3}, c_{ib_i'p}$ 完全是 0 才有 $f(x') < f(x)$.

这样,不必每次计算邻解点的评价函数,大大减少了计算量.

步骤 4.7 停止规则.同步骤 3.6.

步骤4.8 搜索结果.从各搜索过程所得结果中选择具有最小评价函数值的一个作为搜索结果.

5 瓶颈多选择整数规划问题解法(The solution method of the BMCIP)

在改进的启发式群局部搜索的基础上,我们进一步利用分枝剪枝法得到全局最优解,从而提出了一种精确解法.算法步骤如下:

步骤5.1~5.8同于步骤4.1~4.8.

步骤5.9 分枝剪枝法.依据(5)式中的代价系数从大到小对各GUB集合排序.依次从各级GUB集合顺序选择变量,设定相应的(1)中的X,X中相对于尚未选的GUB集合变量的所有位都为0.计算相应的性能指标$\max\limits_k XRCL_k(1 \leqslant k \leqslant l)$,如其已大于等于搜索结果的评价函数,则剪去该分枝退回到上一级GUB集合变量的选取.遍历采用深度优先策略.如m级GUB集合都已选定且性能指标优于现有性能指标,则替代最优解及最优性能指标.因为步骤5.1~5.8的搜索结果已非常接近全局最优解,这一部分工作量很小.

6 ATM网VP路由规划问题算例(Simulation example of VP route programming problem in ATM networks)

6.1 ATM网VP路由规划问题描述(The description of VP route programming problem in ATM networks)

作为仿真算例,将ATM(异步传输模式)网络中虚通道(VP)路由规划(VPR)问题转化为BMCIP问题进行求解.为了方便对业务流的控制和网络资源管理,ATM网络引入了VP(virtual path)的概念,若干VP在同一物理链路中进行复用,而若干虚信道(VC)又在同一VP中进行复用.可以通过建立VP网络重新组织网络的资源提高网络的效率.其中为呼叫建立连接服务的路由规划技术,简称VPR问题,是一个NP完全问题.

使用无向图$G=(V,E)$表示网络的拓扑结构,V是节点的集合,E是边的集合.VP被定义为具有起点s_i,终点t_i,从s_i到t_i的路由P_i和带宽r_i的一条通路.定义$L(e)$为链路的负载,因为,ATM网络的链路带宽通常是有限的,因此减少链路的负载就成为VP优化所要考虑的主要问题.假设VP的终端节点对和带宽已知,VPR问题可以定义为[6]:

定义1 VP的路由是优化的,如果负载最大的链路上负载最小,即

$$\min(\max_{e \in E} L(e)).$$

这样VPR问题就成为已知m个VP的终端对$(s_1,t_1),\cdots,(s_m,t_m)$及每一终端对应的带宽要求$r_i$,寻找路由的集合$P=\{P_1,\cdots,P_m\}$,$P_i$连接$s_i$和$t_i$,并且满足定义1的问题.

路由集合可以使用第k Dijkstra最短路由算法求出[7].求出满足时延约束的所有路由组成路由的集合.设Q_i为第i条VP的路由集合,$Q_i=\{P_i^1\cdots P_i^j\cdots P_i^{n_i}\}$,$P_i^j$为第$i$条VP的第$j$条路由,$n_i$为第$i$条VP满足时延约束的路由的数目.路由的集合求出后,VPR问题就成为m个有各自流量(带宽)要求r_i的VP对,从各自的满足时延约束的路由集合n_i条路由中选择且只能选择一条,使得流量最大的链路上的负载最小化的组合优化问题.X如(4)式定义,X中每一位对应一条路由,选中为1,不选中为0.各VP对流量(带宽)要求r_i构成如(5)式的矩阵R.每一条路由P_i^j经过共l条链路中的某一条k,则C矩阵中相应该路由的行中相应该链路的列c_{ijk}为1,否则为0.性能指标如(1)式,VPR问题转化为BMCIP问题.

6.2 仿真结果(Simulation results)

采用本文提出的算法在图2所示结构的网络[8]上进行了仿真.设VP路由的时延约束为5跳,要求规划的VP终端对和流量要求(带宽需求)如表1所示.算法的性能指标为最小化负载最大的链路上的流量.结果比较如表2所示,可以看出本文提出的改进的启发式群局部搜索算法在最优性能、最坏性能和平均性能上均优于已有算法,本文精确算法能够求出转化后问题的全局最优解,因此是一种有效的算法.

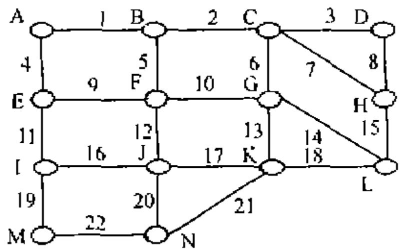

图2 仿真模型图

Fig. 2 Simulation model map

表1 各VP终端对流量要求

Table 1 The flow requirements of each VP pair

VP终端对	AN	AL	DM	BM	DI	CM	DN	AH	HI	MH
流量要求	32	40	25	33	48	34	21	27	15	31

表 2　各算法性能比较

Table 2　Performance comparison of the algorithms

算法	最好性能	最坏性能	平均性能
文献[6]算法	90	169	151
文献[9]算法	88	107	96
模拟退火搜索算法	90	97	94
文献[10]算法	81	94	88
遗传算法	82	88	84
改进的启发式群局部搜索算法	82	82	82
本文精确算法	80		

7　结论(Conclusion)

本文提出了一类瓶颈多选择整数规划问题(BMCIP)并给出了一种有效解法.在改进的启发式群局部搜索的基础上,利用分枝剪枝法得到全局最优解.作为仿真算例,将 ATM 网络中虚通道由规划(VPR)问题转化为此类问题进行求解,并与传统的搜索算法进行了比较.本文没有考虑形如 $AX+b\geqslant 0$ 的约束,对于此类约束可用拉格朗日法或在搜索过程中采用罚函数法.需待进一步地研究.

参考文献(References)

[1] James C Bean. A Langrangian algorithm for the multiple choice integer program [J]. Operations Research, 1984,32(5):1185 – 1193

[2] Zeev Zeitlin. Integer allocation problems of min-max type with quasi-convex separable function [J]. Operations Research, 1981,25(2):207 – 211

[3] Joseph B M. An algorithm for the bottleneck generalized assignment problem [J]. Computers Operations Research, 1993,20(4):355 – 362

[4] Gu Jun. Local search for satisfiability (SAT) problems [J]. IEEE Trans. Systems, Man, and Cybernetics, 1993,23(4):1108 – 1129

[5] Li Bing and Jiang Weisun. Chaos optimization method and its application [J]. Control Theory and Applications. 1997, 14(4):613 – 615 (in Chinese)

[6] Imrich Chlamtac, Andras and Zhang Tao. Optimizing the system of virtual paths [J]. IEEE/ACM Trans. on Networking, 1994,2(6):581 – 587

[7] Lawer E. Combinational Optimization: Networks and Matroids [M]. New York: Holt, Rinehart and Winston, 1976:100 – 105

[8] Yang Yang and Xi Yugeng. Route optimization method in ATM networks based on evolution programming [J]. Communications Technology, 1998,92(1):31 – 34 (in Chinese)

[9] Qu Runtao and Xi Yugeng. Local search method for VP route programming in ATM networks [J]. Communications Technology, 1999,106(3):25 – 28 (in Chinese)

[10] Guo Wei and Xi Yugeng. Tabu search method for VP routes program problem in ATM networks [J]. Journal of China Institute of Communications, 2000,21(12):42 – 46 (in Chinese)

本文作者简介

郭　伟　1973 年生.上海交通大学自动化系博士研究生.主要研究方向为通信网络的路由优化及运筹学 VRP 问题及其相关问题.

Email: guowei568@263.net

席裕庚　1946 年生.教授.上海交通大学博士生导师.主要研究方向:预测控制、多机器人协调系统、生产调度,网络资源优化.