第16卷 第7期
2006年7月
计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT
Vol.16 No.7
Jul. 2006

用Excel求解网络优化问题

胡 斌[1,2],黄天强[2],陈生萍[2]
(1.中南大学 信息科学与工程学院,湖南 长沙 410083;
2.吉首大学 信息管理与工程学院,湖南 张家界 427000)

摘 要:规划求解在各个行业都得到广泛的应用,并取得了显著的经济效益。各个领域中的大量问题都可以归结为线性规划问题。通过实例,分析了用Excel提供的"规划求解"功能解决网络优化中的主要问题,论证Excel对于需要大量进行处理数据研究中的实用性。使用Excel的"规划求解"工具可以很方便解决此类问题,为网络决策分析活动制定最优方案。

关键词:Excel;线性规划;网络优化
中图分类号:TP317.3;O221.1 **文献标识码**:A **文章编号**:1673-629X(2006)07-0004-03

Application of Excel in Optimum Networks Analysis

HU Bin[1,2],HUANG Tian-qiang[2],CHEN Sheng-ping[2]
(1.Information Science and Engineering Institute, Central South University, Changsha 410083,China;
2.Information Management and Engineering Institute, Jishou University, Zhangjiajie 427000,China)

Abstract: The methods of optimization have been adopted and have raised economic benefits apparently. A great many problems can be summed up for linear program problem. This paper introduces a simple, easy method to solve optimization problems, which have a wide application in various practical fields. Some certain examples are given, in which Excel is applied to the solution of optimum networks analysis, thus expounding the practicability of this software in researches.

Key words: Excel; linear program problem; optimum networks

0 引 言

在生产实践和社会生活中,有许多现实的网络,如电力网、通讯网、铁路网等。研究这些网络的管理决策问题就是网络规划,它是运筹学中一个重要的分支。网络规划中主要问题有:最大流问题、最小代价流问题、最短路问题和网络计划关键路径问题等[1~6]。用Excel提供的"规划求解"功能可以解决许多问题,可以解方程(组),解决众多线性规划和非线性规划问题。

Microsoft Excel的"规划求解"工具来自于Leon Lasdon和Allan D. Waren共同开发的Generalized Reduced Gradient(GRG2)非线性最优化代码[7,8]。线性和整数规划采用了有界变量单纯形法和分支边界法。"规划求解"是一组命令的组成部分[3,9],这些命令有时也称作假设分析工具。借助"规划求解",可求得工作表上某个单元格(被称为目标单元格)中公式的最优值。"规划求解"将对直接或间接与目标单元格中公式相关联的一组单元格中的数值进行调整,最终在目标单元格公式中求得期望的结果[9]。"规划求解"通过调整所指定的可更改的单元格(可变单元格)中的值,从目标单元格公式中求得所需的结果。在创建模型过程中,可以对"规划求解"模型中的可变单元格数值应用约束条件,而且约束条件可以引用其他影响目标单元格公式的单元格。文中举例说明如何使用这一功能求解运筹学中的网络规划问题。

收稿日期:2005-09-28
基金项目:湖南省自然科学基金资助项目(05JJ40007)
作者简介:胡 斌(1972-),男(土家族),湖南张家界人,讲师,硕士研究生,研究方向为知识发现与计算智能。

1 相关概念与理论

定义1 在有向图$G=(V,A)$上定义如下的权函数:

(1) $L:A\rightarrow \mathbf{R}$为弧上的权函数,弧$(i,j)$对应的权$L(i,j)$记为$l_{ij}$,称为弧$(i,j)$的容量下界(lower bound)。

(2) $U:A\rightarrow \mathbf{R}$为弧上的权函数,弧$(i,j)$对应的权$U(i,j)$记为$U_{ij}$,称为弧$(i,j)$的容量上界,或直接称为容量(capacity)。

(3) $D:V\rightarrow \mathbf{R}$为顶点上的权函数,节点$i$对应的权$d(i)$记为$d_i$,称为顶点$i$的供需量(supply/demand);特别地,称$d_i>0$的顶点为供应点(supply node)或源(source)、起始点或发货点。称$d_i<0$的顶点为需求点(demand node)或汇(sink)、终止点或吸收点,称$d_i=0$的顶点为转运点(transshipment node)或平衡点、中间点。

定义2 对于流网络$N=(V,A,L,U,D)$,其上的一个流(flow)x是指从N的弧集A到实数集合$\mathbf{R}$的一个函

数 $x:A \rightarrow \mathbf{R}$, 即对每条弧 (i,j) 赋予一个实数(称为弧 (i,j) 上的流)。此时网络可称为流网络(Flow Network,一般仍简称为网络),记为 $N=(V,A,L,U,D)$。对于以 s 为起始点、t 为终止点的运输网络,其中的容量可行且转运点流量守恒的流称为 $s-t$ 可行流,有时为了方便也称为可行流,记作 $N=(s,t,V,A,U)$。

性质1　一个可行流的可行网络满足流量守恒/平衡条件[1,5]。即:$\forall(i) \in V$, 当 i 不为 s 或 t 时,有 $d_i=\sum_{i(i,j)\in A} X_{ij}-\sum_{i(h,i)\in A} X_{hi}=0$。

性质2　一个可行流的可行网络满足容量约束/可行条件[5]。即:$l_{ij} \leqslant X_{ij} \leqslant U_{ij}, \forall(i,j) \in A$。

2　最大流问题

最大流问题(Maximum Flow Problem)就是在 $N=(s,t,V,A,U)$ 中找到流值最大的 $s-t$ 可行流(即最大流)。

算法1　求解最大流问题的算法:

求解 Max v,

约束条件为[5]:

$$\sum_{i(i,j)\in A} X_{ij}-\sum_{i(h,i)\in A} X_{hi}=\begin{cases} v, & \text{当 } i=s \text{ 时} \\ -v, & \text{当 } i=t \text{ 时} \\ 0, & \text{其它} \end{cases}$$

$0 \leqslant X_{ij} \leqslant U_{ij}, \forall(i,j) \in A$

问题1　设有一网络,如图1所示,每条线旁的数字表示容量,如何安排使 $v1$ 到 $v5$ 的流量最大?

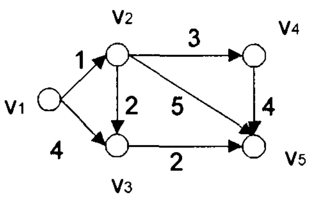

图 1　最大流问题

用 Excel 求解的步骤如下:

1) 初始化各单元格。把网络的原始容量信息输入到工作表中,用 B2:F6 区域表示网络的容量,若没有线路到达,容量为 0;(见表 1)。

表 1　求解最大流原始数据

	A	B	C	D	E	F
1	容量	$v1$	$v2$	$v3$	$v4$	$v5$
2	$v1$	0	1	4	0	0
3	$v2$	0	0	2	3	5
4	$v3$	0	0	0	0	2
5	$v4$	0	0	0	0	4
6	$v5$	0	0	0	0	0

2) 将 B8:F12 区域作为可变单元格,表示相应支路 (i,j) 的真实运输流量。

3) 计算各运输点的供应量。在 G8 单元格中输入"=SUM(B8:F8)",并复制到 G9,G10 和 G11 中,使它们分别变为"=SUM(B9:F9)"、"=SUM(B10:F10)"和"=SUM(B11:F11)"。

4) 计算各运输点的需求量。使 C13,D13,E13 单元格中分别变成"=SUM(C8:C12)"、"=SUM(D8:D12)"和"=SUM(E8:E12)"。

5) 选择菜单"工具"中的"规划求解",出现"规划求解参数"对话框如图 2,逐一填充各栏中的空白:

①在"设置目标单元格"栏后的空白中填入 \$G\$8,并选中"最大"。

②在"可变单元格(B)"栏后的空白中填入 \$B\$8:\$F\$12。

③光标指向"约束"栏,按"添加",出现"添加约束"对话框,依次填入约束关系,每输完一条,按"添加",输入所有约束条件后,按"确定",又退回到图 2 状态,在图 2 中可以选"更改"、"删除"、"全部重设"来编辑约束条件及其他设置。分别按容量约束/可行条件输入约束条件"\$B\$2:\$F\$6>=\$B\$8:\$F\$12"和"\$B\$8:\$F\$12>=0",按流量守恒/平衡条件输入"\$G\$9=\$C\$13"、"\$G\$10=\$D\$13"和"\$G\$13=\$E\$13"。

④按图 2 中的"选项"按钮,可以调整运算精度等多项选项。

⑤在图 2 中按"求解",即进入求解过程,求解结束,出现"规划求解结果"对话框,选择"保存规划求解结果"后,工作表中可变单元格、目标单元格以及计算约束条件的单元格均发生变化。如不想破坏原始数据,可选择"恢复为原值",同时选中"报告"框中的"运算结果报告",或选"保存方案"以存储运算结果。

运行后得到的最大流方案见表 2,最大流为 3。

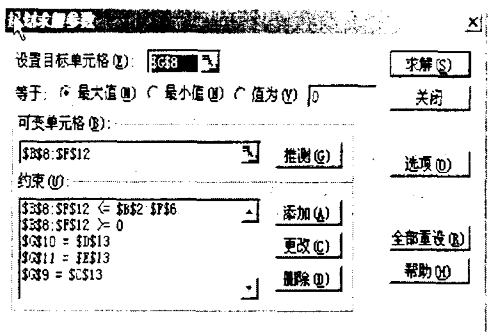

图 2　规划求解参数

表 2　最大流最优结果

	A	B	C	D	E	F	G
7	流量	$v1$	$v2$	$v3$	$v4$	$v5$	供应量
8	$v1$	0	1	2	0	0	3.0
9	$v2$	0	0	0	0.8	0.2	1.0
10	$v3$	0	0	0	0	2.0	2.0
11	$v4$	0	0	0	0	0.8	0.8
12	$v5$	0	0	0	0	0	
13	需求量		1.0	2.0	0.8		

3 最小代价流问题

定义 3 在流网络 $N=(V,A,L,U,D)$ 上增加权函数 $C: A\to\mathbf{R}$ 为弧上的权函数，弧 (i,j) 对应的权 $C(i,j)$ 记为 C_{ij}，称为弧 (i,j) 的单位流量的成本或费用(cost)。此时网络可称为容量 - 费用网络（一般仍可简称为网络），记为 $N=(V,A,C,L,U,D)$。

定义 4 容量 - 费用网络中的流 x 的(总)费用定义为：$C(X)=\sum\limits_{(i,j)\in A}C_{ij}X_{ij}$

最小费用流问题就是在网络中寻找总费用最小的可行流。

算法 2 求解最小费用流问题的算法：

求解：$\mathrm{Min}\ C(X)=\sum\limits_{(i,j)\in A}C_{ij}X_{ij}$

约束条件[5]：

$$\sum_{i(i,j)\in A}X_{ij}-\sum_{i(h,i)\in A}X_{hi}=\begin{cases}v, & 当\ i=s\ 时\\ -v, & 当\ i=t\ 时\\ 0, & 其它\end{cases}$$

$0\leqslant X_{ij}\leqslant U_{ij},\ \forall(i,j)\in A$

问题 2 图 3 给出了一个具有代价的网络，每条有向边旁的数字分别表示容量和代价，如何安排使 s 到 t 的流量为 5 时费用最小？

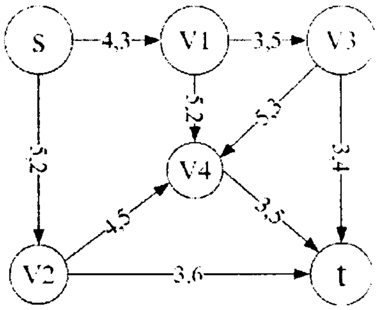

图 3 求解最小代价流问题

用 Excel 求解的步骤如下：

1) 用 B2:G7 区域表示网络各路径的容量 C_{ij}，用 B10:G15 区域表示网络各路径的单位费用 X_{ij}(见表 3)。

表 3 最小代价流问题的原始数据

	A	B	C	D	E	F	G
1	容量	*s*	*v*1	*v*2	*v*3	*v*4	*t*
2	*s*	0	4	5	0	0	0
3	*v*1	0	0	0	3	5	0
4	*v*2	0	0	0	0	4	3
5	*v*3	0	0	0	0	5	3
6	*v*4	0	0	0	0	0	0
7	*t*						
9	费用	*s*	*v*1	*v*2	*v*3	*v*4	*t*
10	*s*	0	3	2	0	0	0
11	*v*1	0	0	0	5	2	
12	*v*2	0	0	0	0	5	6
13	*v*3	0	0	0	0	3	4
14	*v*4	0	0	0	0	0	5
15	*t*	0	0	0	0	0	0

2) 将 B18:G23 区域作为可变单元格，表示相应支路 (i,j) 的真实运输流量。

3) 计算各运输点的供应量。使 H18:H23 单元格中分别变成"= SUM(B18:G18)"、"= SUM(B19:G19)"、"= SUM(B20:H20)"、"= SUM(B21:H21)"和"= SUM(B22:H22)"。

4) 计算各运输点的需求量。使 C24, D24, E24, F24 单元格中分别变成"= SUM(C18:C23)"、"= SUM(D18:D23)"、"= SUM(E18:E23)"、"= SUM(F18:F23)"和"= SUM(G18:G23)"。

5) 计算起始点和各转运点 i 的各供应支路 (i,j) 的运输代价 $\sum\limits_{i(i,j)\in A}C_{ij}X_{ij}$，使 I18:I23 单元格中分别变成："= SUM(B18 * B10 + C18 * C10 + D18 * D10 + E18 * E10 + F18 * F10 + G18 * G10)"、"= SUM(B19 * B11 + C19 * C11 + D19 * D11 + E19 * E11 + F19 * F11 + G19 * G11)"、"= SUM(B20 * B12 + C20 * C12 + D20 * D12 + E20 * E12 + F20 * F12 + G20 * G12)"、"= SUM(B21 * B13 + C21 * C13 + D21 * D13 + E21 * E13 + F21 * F13 + G21 * G13)"和"= SUM(B22 * B14 + C22 * C14 + D22 * D14 + E22 * E14 + F22 * F14 + G22 * G14)"、"= SUM(B23 * B15 + C23 * C15 + D23 * D15 + E23 * E15 + F23 * F15 + G23 * G15)"

6) 计算总运输代价，使 I17 单元格中变成"= SUM(I18:I23)"。

7) 选择菜单"工具"中的"规划求解"，出现"规划求解参数"对话框，逐一填充各栏中的空白：

①在"设置目标单元格"栏后的空白中填入 $I$17，并选中"最小"。

②在"可变单元格(B)"栏后的空白中填入 $B$18:$G$23。

③光标指向"约束"栏，分别按容量约束/可行条件输入"$B$18:$G$23> = 0"和"$B$18:$G$23< = $B$2:$G$7"，按流量守恒/平衡条件输入约束条件"$H$18 = 5"、"$H$19 = $C$24"、"$H$20 = $D$24"、"$H$21 = $E$24"和"$H$22 = $F$24"。

④按"求解"，运行后得到的最小代价流方案见表 4，最小费用为 36。

表 4 最小代价流问题的最优结果

	A	B	C	D	E	F	G	H	I
17	代价	*s*	*v*1	*v*2	*v*3	*v*4	*t*	供应量	36
18	*s*	0	2	3	0	0	0	5	0
19	*v*1	0	0	0	2	0	0	2	10
20	*v*2	0	0	0	0	0	3	3	18
21	*v*3	0	0	0	0	0	2	2	8
22	*v*4	0	0	0	0	0	0	0	0
23	*t*	0	0	0	0	0	0	0	0
24	需求量		2	3	2	0	5		

（下转第 9 页）

```
private static Connection con=null;
private static String DBdriver="oracle.jdbc.driver.OracleDriver";
private static String DBurl="jdbc:oracle:thin:@10.129.10.71:1521:ORCL";
private static String DBuser="kduser";
private static String DBpassword="kduser";
public ssDatabaseAccess(){}
public int initDB(String sdriver,String surl,String suser,String spassword){……}//初始化数据库
public int Close(){……}//关闭数据库
public String[][] execQuery(String sSQL){……}//将查询的结果按照二维数组返回
public int execUpdate(String sSQL){……}//执行没有返回值的SQL语句
}
```

然后利用集成了Web Service的开发环境生成一个Web Service的应用。

接下来利用Delphi来实现客户端,使用Delphi是因为它集成了Web Service。将已经建立起来的Web Service导入到Delphi中,就可以得到一些相应的类和文件,直接就可以拿来使用。

这种方式的好处在于,在客户端可以不需要任何附加的东西。但是在服务器端需要有Web服务器作为Web Service的载体。

2.3 OCI

以OCI作为Oracle客户端开发的应用程序接口的文章已经很多,这里不再赘述,可以参看文献[3~5]。

以这种方式开发的Oracle客户端开发可以完成所有Oracle客户端的全部功能,具有非常好的性能,并且其客户端体积也大大减小。但是,这种方式使得开发的过程相当复杂,且代码维护困难。

3 结束语

通过以上的分析,可以得到如表1所示的结论。

	代码维护性	编程复杂性	可移植性	服务器端要求	代码效率
JNI	好	简单	更换数据库之后只需要将数据库驱动改变	没有要求	与Java本身的效率有关
OCI	差	复杂	只适用于Oracle	没有要求	高,直接通过网络访问数据库
Web service	好	简单	更换数据库之后只需要将数据库驱动改变	要求有Web服务器	与Java本身的效率有关

从表中可见,当程序需要较高的效率的时候,使用OCI比较好;当程序没有效率要求的时候,用其它两种方式比较简单,也提高了开发效率。

参考文献:

[1] Sun Microsystems. JNI specification[S]. 2003.

[2] 吕 曦,王化文. Web Service的架构与协议[J]. 计算机应用,2002,22(12):62-65.

[3] 曾志聪,姚国祥. 基于OCI技术的Oracle数据库连接[J]. 微机发展,2004,14(8):11-13.

[4] 李乾富,黄书强. VC6.0访问Oracle LOB的方法[J]. 微型机与应用,2003(2):7-9.

[5] 汪林林,马 锐. 用OCI开发ORACLE数据库的方法[J]. 计算机应用,2003,23(12):46-57.

(上接第6页)

4 算法分析

用Excel的规划求解工具线性规划问题,简单易行,很容易掌握。其规律及技巧可归纳为:在实际的求解过程中,只需确定目标函数单元格及"可变单元格"区域位置两处单元格位置,然后正确地输入约束条件和确定所求的目标是最大还是最小即可求得正确结果。

5 结束语

利用Excel提供的规划求解法可以解运筹学中的许多问题,譬如线性规划、运输问题、指派问题、网络最优化问题、目标规划、机器分配问题、人事安排等。使用Excel的"规划求解"工具可以很方便地解决此类问题,为人们的决策活动制定最优方案。

参考文献:

[1] Balakrishnan V K. Network optimization[M]. London: Chapman Hall, 1995.

[2] 高 尚. 用Excel求解网络规划问题[J]. 计算机与信息技术,2000(12):73-76.

[3] 平 澄. 用Excel解方程和得出数学模型的最优化解[J]. 电脑开发与应用,2002(9):23-25.

[4] Ahuja R K. Network Flows: Theory, Algorithms, and Applications[M]. Englewood Cliffs, New Jersey: Prentice Hall. 1993.

[5] 谢金星,邢文训. 网络优化[M]. 北京:清华大学出版社. 2003.

[6] 唐焕文. 实用最优化方法[M]. 大连:大连理工大学出版社. 2002.

[7] Fylstra D, Lasdon L. Design and Use of the Microsoft Excel Solver[J]. Interfaces, 1998,28(5):29-55.

[8] Waren A D. The status of nonlinear programming software: an update[J]. Operations Research Archives, 1987,35:489-503.

[9] 顾运筠. Excel规划求解的两类应用[J]. 计算机应用与软件,2005,22(1):137-139.