

## 第十章 线性规划及其在浪费性通勤测算和医疗服务区位优化中的应用

线性规划(Linear Programming 或简称 LP)是一种优化分析技术, 在社会经济分析与规划中的应用很广。线性规划就是在一组约束条件下, 寻找目标函数的最大值或最小值, 由于目标函数和约束条件都是线性函数, 故称线性规划。很显然, 有关线性规划的问题远不是一个章节能尽述的, 在许多大学的地理系、规划系或其它相关的科系, 一般都有一门或几门课程来讲授线性规划。本章重点讲解线性规划的建模, 以及如何利用 SAS 与 ArcGIS 这两个软件包来求解有关的线性规划问题。

本章共分 5 节。第 10.1 节复习一下线性规划的基本模型和单纯形解法。第 10.2 节以城市中浪费性通勤的测算为例, 讲解一般线性规划 (不涉及整数线性规划) 的具体应用。选用通勤这个主题来讲解线性规划方法的应用, 具有重要意义。通勤是城市研究的一个重要主题, 是联系城市居民点和就业地的桥梁, 与城市结构和土地利用等理论问题关系密切。同时, 通勤又反映居民上下班的方便程度, 优化通勤与环境保护、设计建造高效能城市等有关公共政策的制定也有重要的参考价值。查阅目前关于通勤研究的文献, 已不单单局限于浪费性通勤, 而是涉及到通勤与城市土地利用之间的关系、城市内部通勤差异的解释、通勤对“空间错位 (spatial mismatch)”和就业便捷性的意义等诸多方面。然而, 回顾我们对通勤问题的研究兴趣, 很大程度上要归功于哈密尔顿 (Hamilton, 1982) 早年提出的浪费性通勤问题。第 10.2 节以美国俄亥俄州哥伦布都市区为例, 讲解浪费性通勤的测算方法, 同时演示如何用 SAS 来求解一般线性规划问题。

第 10.3 节介绍整数线性规划。整数线性规划不同于一般线性规划的是, 有些决策变量要求是整数。区位优化 (location-allocation) 中的一些著名问题, 如 p-中位问题 (p-median problem)、最少服务点问题 (location set covering problem 或简称 LSCP) 和最大服务面问题 (maximum covering location problem 或简称 MCLP) 等, 都是整数线性规划的例子。无论在社会公众服务部门, 还是在私营企业部门, 区位优化模型都有十分广泛的用途。第 10.4 节用

俄亥俄州凯霍加县医疗服务规划中的一个案例，讲解如何利用 ArcGIS 进行区位优化分析。

第 10.5 节为本章总结。

## 10.1 线性规划与单纯形法

190

### 10.1.1 线性规划标准型

标准的线性规划问题可以概括为：

求  $\sum_{j=1}^n c_j x_j$  的最大值，其中所有  $n$  个变量  $x_1, x_2, \dots, x_n$  为非负（也就是  $x_j \geq 0$ ，

$j \in \{1, 2, \dots, n\}$ ），而且满足  $m$  个约束条件  $\sum_{j=1}^n a_{ij} x_j \leq b_i$ （其中  $i \in \{1, 2, \dots, m\}$ ）。

上述问题中的  $\sum_{j=1}^n c_j x_j$  为目标函数，任务就是寻找满足约束条件的最优解  $x_j$ （其中  $j \in \{1, 2, \dots, n\}$ ）。

用矩阵形式，线性规划问题可以描述为：

给定  $\mathbf{c} \in \mathbf{R}^n$ ， $\mathbf{b} \in \mathbf{R}^m$ ， $\mathbf{A} \in \mathbf{R}^{m \times n}$ ，在满足  $\mathbf{Ax} \leq \mathbf{b}$  和  $\mathbf{x} \geq 0$  的约束条件下，求  $\mathbf{c}^T \mathbf{x}$  的最大值。

由于线性规划问题完全由  $\mathbf{A}$ ， $\mathbf{b}$ ， $\mathbf{c}$  三个矩阵参数所决定，因此，可以简洁地概括为（ $\mathbf{A}$ ， $\mathbf{b}$ ， $\mathbf{c}$ ）问题。对于不是标准型的线性规划问题，可以通过以下变换规则转变成标准型（Kincaid and Cheney, 1991, 648 页）：

1. 求  $\mathbf{c}^T \mathbf{x}$  的最小值等同于求  $-\mathbf{c}^T \mathbf{x}$  的最大值。

2. 约束条件  $\sum_{j=1}^n a_{ij} x_j \geq b_i$  等同于  $-\sum_{j=1}^n a_{ij} x_j \leq -b_i$ 。

3. 约束条件  $\sum_{j=1}^n a_{ij}x_j = b_i$  等同于  $\sum_{j=1}^n a_{ij}x_j \leq b_i, -\sum_{j=1}^n a_{ij}x_j \leq -b_i$ 。

4. 约束条件  $\sum_{j=1}^n |a_{ij}x_j| \leq b_i$  等同于  $\sum_{j=1}^n a_{ij}x_j \leq b_i, -\sum_{j=1}^n a_{ij}x_j \leq b_i$ 。

5. 如果  $x_j$  为负数，则用两个非负变量来代替，如  $x_j = u_j - v_j$ 。

### 10.1.2 单纯形法 (Simplex Algorithm)

单纯形法 (Dantzig, 1948) 在线性规划问题的求解中广为使用。这里不讨论单纯形法的理论和证明，只是通过一个简单的实例来直接讲述单纯形法的具体步骤。

让我们看下面这个线性规划问题，已写成标准型。如果线性规划问题不是用标准型表达的，需要在用单纯形法之前，根据前面所述的规则先将它变为标准型。

求  $z = 4x_1 + 5x_2$  的最大值

约束条件为：

$$2x_1 + x_2 \leq 12$$

$$-4x_1 + 5x_2 \leq 20$$

$$x_1 + 3x_2 \leq 15$$

$$x_1 \geq 0, x_2 \geq 0$$

191

单纯形法的第一步是将不等式变为等式，具体方法是引入松弛变量  $\mathbf{u} \geq 0$ ，将约束条件中的不等式  $\mathbf{Ax} \leq \mathbf{b}$  转化为等式  $\mathbf{Ax} + \mathbf{u} = \mathbf{b}$ 。

对于上面例子，引入三个松弛变量 ( $x_3 \geq 0, x_4 \geq 0, x_5 \geq 0$ )，这样原先的线性规划问题可以写成下面的形式：

求  $z = 4x_1 + 5x_2 + 0x_3 + 0x_4 + 0x_5$  的最大值

约束条件为：

$$2x_1 + x_2 + x_3 + 0x_4 + 0x_5 = 12$$

$$-4x_1 + 5x_2 + 0x_3 + x_4 + 0x_5 = 20$$

$$x_1 + 3x_2 + 0x_3 + 0x_4 + x_5 = 15$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0$$

单纯形法通常写成表格的形式。上述例子可改写为下述表格：

$$\begin{array}{cccccc} 4 & 5 & 0 & 0 & 0 & \\ 2 & 1 & 1 & 0 & 0 & 12 \\ -4 & 5 & 0 & 1 & 0 & 20 \\ 1 & 3 & 0 & 0 & 1 & 15 \end{array}$$

其中最顶上一行是目标函数  $\mathbf{c}^T \mathbf{x}$  的系数，接下来的  $m$  行是约束条件。顶行最右上角的空白项就是待求目标函数  $z$  的最大值。这样，单纯形法的通用表格式就是：

$$\begin{array}{ccccc} \mathbf{c}^T & \mathbf{0} & & & \\ \mathbf{A} & \mathbf{I} & \mathbf{b} & & \end{array}$$

一般情况下，如果线性规划问题有解，其解可以通过单纯形法求得；如果没有解（如线性规划问题无边界），则在单纯形法的算法过程中就可以发现这个问题。

如果变量  $x_1$ 、 $x_2$  取值为 0，初始解（ $x_1 = 0, x_2 = 0, x_3 = 12, x_4 = 20, x_5 = 15$ ）显然满足约束条件。定义  $x_j$  值为 0 的变量为非基变量，而余下的、通常不为 0 的变量为基变量。单纯形表格中有  $n$  个非基变量和  $m$  个基变量，分别对应着初始变量和松弛变量的个数。上述例子中， $x_1$ 、 $x_2$  为初始的非基变量（ $n = 2$ ）， $x_3$ 、 $x_4$ 、 $x_5$  为基变量（ $m = 3$ ）。在表中的约束条件中，每个基变量只出现在一行中，而目标函数都是由非基变量来表示的。

在单纯形法的每一步变换中，我们通过将一个非基变量转变为基变量的方法来逐步增加目标函数的值，最终达到求解的目的。由于在线性方程组中行变换不会影响方程组的解，因此，可以通过矩阵变换中的高斯消去法来完成相关的变换。单纯形法的流程如下：

1. 选择目标函数中最大正系数的变量  $x_s$  为新的基变量，即  $c_s = \max\{c_i > 0\}$ 。
2. 在每一行中，用本行中  $b_i$  除以基变量的系数  $a_{ij}$ 。在  $a_{is} > 0$  的行中，选择其中最小的  $b_i / a_{ij}$ ，把这个值定为该基变量的解，也就是， $x_s = b_k / a_{kj} = \min\{b_i / a_{ij}\}$ 。如果所有的  $a_{is}$  都小于或等于 0，则线性规划无解。
3. 以  $a_{ks}$  为轴项，利用高斯消去法在  $s$  列上产生系数为 0 的项，即保持第  $k$  行不变，用其它的行来减它。
4. 如果目标函数（顶行）中的所有系数都小于或等于 0，则当前  $x$  就是线性规划的解；否则，继续进行转换。

针对上述例子，以上四个步骤具体操作如下。

在第一步中，由于目标函数中系数最大的正值是 5，所以  $x_2$  成为新的基变量。

在第二步中， $a_{22}$  被选定为轴项（下表中加有下划线），因为  $20/5$  在  $\{12/1, 20/5, 15/3\}$  中是最小的，基变量的解  $x_2 = 20/5 = 4$ 。

$$\begin{array}{cccccc}
 0.8 & 1 & 0 & 0 & 0 & \\
 2 & 1 & 1 & 0 & 0 & 12 \\
 -0.8 & \underline{1} & 0 & 0.2 & 0 & 4 \\
 0.3333 & 1 & 0 & 0 & 0.3333 & 5
 \end{array}$$

在第三步中，采用高斯消去法产生一个新的表（用其它的行减轴项所在的行）：

$$\begin{array}{cccccc}
 1.6 & 0 & 0 & -0.2 & 0 & \\
 2.8 & 0 & 1 & -0.2 & 0 & 8 \\
 -0.8 & \underline{1} & 0 & 0.2 & 0 & 4 \\
 1.1333 & 0 & 0 & -0.2 & 0.3333 & 1
 \end{array}$$

参照单纯形法流程的第四步，由于顶行中的系数  $c_1 = 1.6 > 0$ ，没有达到所有系数都小于或等于 0 的要求，因此，继续进行变换。

类似的， $x_1$  成为新的基变量，确定  $a_{13}$  为轴项（因为  $1/1.133 < 8/2.8$ ），基变量  $x_1 =$  193

$1/1.1333 = 0.8824$ ，进行高斯消去后，形成新的表：

0	0	0	0.0515	-0.2941	
0	0	0.3571	0.1050	-0.2941	1.9748
0	1.25	0	0.0735	0.2941	5.8824
<u>1</u>	0	0	-0.1765	0.2941	0.8824

再继续变换下去， $x_4$  成为新的基变量，产生新的表：

0	0	-3.4	0	-2.9143	
0	0	3.4	1	-2.8	18.8
0	17	-3.4	0	6.8	61.2
5.6667	0	3.4	0	-1.1333	23.8

变换到这里，目标函数（顶行中）的所有系数都小于或等于 0，求解完毕。整理决策变量的解为： $x_1 = 23.8/5.6667 = 4.2$ ， $x_2 = 61.2/17 = 3.6$ ，目标函数的最大值为  $z_{max} = 4 \times 4.2 + 5 \times 3.6 = 34.8$ 。

有许多软件包（包括一些免费的）可用来求解线性规划问题。参见网站 <http://www-unix.mcs.anl.gov/otc/Guide/faq/linear-programming-faq.html>。在下面的第 10.2 节，我们选用 SAS 软件包（其中的 OR 模块）的 LP 程序，讲解如何通过 SAS 来求解线性规划问题。前面章节中曾经介绍过 SAS，这是一个功能强大的软件包，我们选用它来求解线性规划，主要是 SAS 在编码大型矩阵和运算时的方便。SAS 软件包中的 LP 程序可以求解一般型线性规划、整数型线性规划和混合型线性规划等多种问题。

## 10.2 案例研究 10A：测算俄亥俄州哥伦布大都市区的浪费性通勤

### 10.2.1 浪费性通勤问题的提出与测算

浪费性通勤问题，最早是由城市经济学家哈密尔顿（Hamilton, 1982）提出的。假定一个城市内所有居民的住地和工作地是已知的，而居民之间可以自由地调换住所，那么，规划的问题就是：怎样通过调换居民的住所而达到城市总的通勤时间最小？哈密尔顿是经济学家，

那时候当然不知道 GIS 技术的存在，他选用负指数模型来描述居住地和工作地的密度分布，两者都是由市中心向边缘递减。由于就业的分布总是比居民人口的分布更朝市中心（CBD）集中，数学上表现为就业密度分布函数的递减梯度要比居民人口密度分布函数更陡一些。因此，人们上班时的最优通勤方向总是往 CBD 的方向，详见附录 10A。哈密尔顿对美国 14 个城市进行了实证研究，发现有 87% 的浪费性通勤。后来，怀特（White, 1988）用一个简明的线性规划模型来测算这种浪费性通勤。怀特估算的浪费性通勤值很小，极可能是因为她选用了较大的地理单元。后来，又有学者（Small and Song, 1992）基于更小的地理单元，利用怀特的线性规划模型分析了美国洛杉矶市的通勤数据，发现约有 66% 的浪费性通勤。这一结果虽然比哈密尔顿估算的浪费性通勤要少一些，但还是很高。总之，究竟多少是浪费性通勤，学术界至今仍有争论。下面介绍如何用怀特的线性规划模型来测算浪费性通勤，同时帮助我们认识争论存在的根本原因。

给定  $P_i$  为居住地  $i$  ( $= 1, 2, \dots, n$ ) 的上班人数， $E_j$  为就业地  $j$  ( $= 1, 2, \dots, m$ ) 的上班人数 194，最少通勤的计算就是下面的线性规划问题。

求  $\sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}$  的最小值。

约束条件为：

$$\sum_{j=1}^m x_{ij} \leq P_i, \text{ 对每一个居住地 } (i = 1, 2, \dots, n)$$

$$\sum_{i=1}^n x_{ij} \leq E_j, \text{ 对每一个就业地 } (j = 1, 2, \dots, m)$$

$$x_{ij} > 0, \text{ 对每一个 } (i = 1, 2, \dots, n) \text{ 和每一个 } (j = 1, 2, \dots, m)$$

式中  $c_{ij}$  是表示从居住地  $i$  到就业地  $j$  的通勤距离或时间， $x_{ij}$  表示这个方向上的通勤人数。

目标函数是城市通勤的总量，第一个约束条件定义了从每个居住地到各个就业地的通勤

人数不能超过该居住地的上班人数，第二个约束条件定义了从不同的居住地到每个就业地的通勤人数不能超过该就业地总的上班人数。在美国的大都市区内，按就业地统计的总上班人数一般超过按居住地统计的总上班人数，即  $\sum_{i=1}^n P_i \leq \sum_{j=1}^m E_j$ ，也就是说，总体上看，都市区内就业岗位多于住在区内的人口，有一部分人住在都市区外、而到都市区内来上班。

### 10.2.2 ArcGIS 上的数据准备

在光盘中提供的相关数据集有：

1. 图层 urbtazpt 包含 1990 年哥伦布大都市区内城市化地区的 991 个地域单元 (TAZ)；
2. 图层 road 包含研究区的路网。

上述空间数据集来自于美国人口普查局 1990 年的 TIGER 文档，参见第一章第 1.2 节有关 TIGER 文档的介绍。哥伦布大都市区包括七个县 (Franklin, Union, Delaware, Licking, Fairfield, Pickaway 和 Madison)，先将这七个县的交通分析区 (Traffic Analysis Zone 或称 TAZ) 的资料合并，再选取其中城市化地区形成图层 urbtazpt。TAZ 是美国人口普查局在各都市区规划委员会的协作下，专门为交通规划设计的一级地域单元。TIGER 文档对每个 TAZ (象普查街区 census block 一样) 都明确标注是否已城市化。由于我们关心的是城市内部的通勤问题，所以，研究区是由都市区内所有城市化的 TAZ 合并而成的。这里所定义的城市化区，大体相当于中国的城市建成区，如图 10.1 所示。参见有关文献 (Wang, 2001b)，其通勤方面的研究也是用的同一研究区。同样地，路网图层 road 也是通过合并研究区内各县道路网络而成，为了保持路网的连通性，此图层的地域范围要稍大于都市区城市化的范围。在 urbtazpt 图层属性表中，emp 项代表每个 TAZ 作为就业地的上班人数，work 项代表每个 TAZ 作为居住地的上班人数，同时为便于参考，popu 项列出了 TAZ 中总人口数。这些数据来自于 1990 年的 CTPP 数据集 (参见第三章第 3.5 节的有关论述)，从 CTPP 的第一部分中抽取了居住地



的上班人数和人口总数，从 CTPP 的第二部分中抽取了就业地的上班人数。

图层 urbtazpt 的属性项 emp 和 work 定义了每个 TAZ 区中按居住地（通勤的起点）和就业地（通勤的终点）而统计的上班人数，对应于上述线性规划模型中的两组变量  $P_i$  ( $i = 1, 2, \dots, n$ ) 和  $E_j$  ( $j = 1, 2, \dots, m$ )。接下来需要通过路网来计算通勤时间  $c_{ij}$ 。第二章曾讨论过有关路网距离和时间的概念和量算方法，正好通过这个案例再熟悉一下。读者也可跳过下面这部分内容，直接进入下一步基于 SAS 的浪费性通勤测算分析（这一小节的运算结果，可见随书所附的光盘）。

下面解释利用 ArcGIS 进行通勤时间计算的具体流程，若有必要，请参考第二章第 2.2 节 196 和第 2.3 节中的相关内容。由于美国城市里绝大多数人驾车上班，这里的通勤时间指的是开车上班时间（不是坐公交、搭便车、骑自行车或步行时间）。本案例假定同样级别的道路具有相同的车流速度。这是一种简化处理方法，忽略了许多复杂因素，参见有关文献（Wang, 2003）的相关讨论。

### 1. 提取居住地（通勤的起点）和就业地（通勤的终点）的点图层

在 ArcMap 中打开图层 urbtazpt，(1)输入选择条件 emp>0，命名输出查询结果为 emptaz (是一个含有 931 个就业地的点图层)；(2)输入选择条件 work >0，命名输出查询结果为 restaz (是一个含有 812 个居住地的点图层)。这一步是为了确定通勤的起点上班人数不为零，同时通勤的终点就业人数也不是零。

### 2. 求离居住地最近的交通结点

路网图层 road 中已含有交通线和交通结点的拓扑关系（若没有，可用 build topology 工具来创建）。首先利用 near 工具，选择 restaz 为 Input Feature（输入图层）、road（其中的 node 图层）为 Near Feature（邻近图层），求得离居住地最近的交通结点。与第二章第 2.3 节中的案例不同的是，这里每个居住地只对应一个最近结点，这样计算方法上比第 2.3 节中介绍的步骤简单。然后通过 Join 工具，将 road 的结点 (node) 图层中有关结点的信息（

特别是 road\_id 属性项) 合并到图层 restaz 的属性表中, 并将合并的结果输出另存为一个新的表 resid.dbf。

### 3. 定义起始结点

需要在 ArcInfo 的工作站 (workstation) 平台下实现。定义一个 INFO 文件作为起始结点, 按照第二章第 2.3.2 节中的第 4 步, 将 dBase 文件 resid.dbf 变为 INFO 文件 resid, 保留 road\_id 项, 并将它重命名为 road-id, 其他项统统舍去, 这个文件中有 812 个起始结点。

### 4. 定义离就业地最近的结点作为终止结点

重复上面的第 2 步 (注意用 near 工具时 Input Feature 定义为就业地图层 emptaz) 和第 3 步, 生成一个 INFO 文件 empid, 这个文件就是终止结点。每个就业地对应着一个最近的路网结点, 共有 931 个终止结点。

### 5. 定义路网的交通阻滞参数

在 road (其中的 line 图层) 的属性表中增加表示速度的属性项 speed。类似于第五章表 5.1 中的交通限速, 这里简单地按属性项 CFCC, 设置相关级别道路的车流速度。

(a)  $CFCC \geq 'A11'$  and  $CFCC \leq 'A18' \rightarrow speed = 55$

(b)  $CFCC \geq 'A21'$  and  $CFCC \leq 'A28' \rightarrow speed = 45$

(c)  $CFCC \geq 'A31'$  and  $CFCC \leq 'A38' \rightarrow speed = 35$

(d) 其它  $\rightarrow speed = 25$

在同一属性表中, 再加入一个表示通勤时间的属性项 time, 按公式  $time = (length/speed) * 60 / 1609$  进行计算。由于路网图层的长度单位是米, 而速度以英里/小时为单位, 利用 1 英里 = 1609 米, 换算得来的时间单位为分钟。

### 6. 在 ArcPlot 中计算通勤时间

也需要在 ArcInfo 的工作站 (workstation) 平台下实现。利用下面的命令可以计算路网交

通时间，结果保存为 INFO 文件 odtime，其中的属性项 network 就是路网结点间的通勤时间。

```
ap                                /* 启动 arcplot 模块

netcover road colsroute          /* 设置路网系统

centers resid                    /* 定义起始结点

stops empid                      /* 定义终止结点

impedance time                   /* 定义交通阻滞参数

nodedistance centers stops odtime 600 network ids

q                                /*退出
```

## 7. 通勤时间修正

这一步就是要把从居住地到路网结点、从就业地到路网结点的这两段直线距离转换成通勤时间，再添加到上述的路网结点之间的通勤时间上去。首先，把表 resid.dbf（含居住地到最近结点的直线距离）连接到表 odtime，然后，将表 empid.dbf（含就业地到最近结点的直线距离）也连接到表 odtime。在表 odtime 中，增加一个新属性项 comtime，并按如下公式计算其值， $\text{odtime:comtime} = \text{odtime:network} + ((\text{empid.near\_dist} + \text{resid.near\_dist}) / 25) * 60/1609$ 。这里假设两段直线距离上的行车速度为每小时 25 英里。这个 INFO 文件中有  $812 \times 931 = 755,972$  条记录。

## 8. 准备用于浪费性通勤分析的文件

将 urbtazpt 图层属性表中的所有记录输出为文本格式文件 urbtaz.txt，中间以空格符作为定界符，首行不需要变量名，仅保存 taz、work、emp 这三个变量的值。同理，将 INFO 文件 odtime 也输出为以空格作为定界符的文本文件 odtime.txt，首行不需要字段名列表，只含有 resid.taz、empid.taz、comtime 这三个变量的值，这两个文件将在后面的浪费性通勤分析中使用。光盘有一个 AML 程序 (rdtime.aml) 可用来执行上述从

第 1 步到第 8 步的相关数据处理操作。

### 10.2.3 利用 SAS 进行浪费性通勤测算

SAS 采用两种格式来定义线性规划问题，即紧凑式和松散式。松散式有利于编码大型矩阵，在这里将讲解这种方法。松散式输入时，通常用 COEF、COL、TYPE 和 ROW 等关键词的定义来定义线性规划中的各个系数，也可直接借用 SAS 的内定义变量 \_COEF\_、\_COL\_、\_TYPE\_ 和 \_ROW\_。比如第 10.1 节中讲到的例子，用 SAS 来解，代码如下。

```
Data;

Input _row_ $1-6 _coef_ 8-9 _type_ $11-13 _col_ $15-19;

Cards;

Object . max . /*定义目标函数类型 */

Object 4 . x1 /*目标函数的第 1 个变量系数*/

Object 5 . x2 /*目标函数的第 2 个变量系数*/

Const1 12 le _RHS_ /*第 1 个约束条件的等式右项*/

Const1 2 . x1 /*第 1 个约束条件的第 1 个变量系数*/

Const1 1 . x2 /*第 1 个约束条件的第 2 个变量系数*/

Const2 20 le _RHS_ /*第 2 个约束条件的等式右项*/

Const2 -4 . x1 /*第 2 个约束条件的第 1 个变量系数*/

Const2 5 . x2 /*第 2 个约束条件的第 2 个变量系数*/

Const3 15 le _RHS_ /*第 3 个约束条件的等式右项*/

Const3 1 . x1 /*第 3 个约束条件的第 1 个变量系数*/

Const3 3 . x2 /*第 3 个约束条件的第 2 个变量系数*/

;

Proc lp sparsedata; /*运行 LP 分析 */
```

Run;

在上述 SAS 程序代码中，主要工作是定义目标函数和约束条件。通过下面 4 个变量来定义一个目标函数或一个约束条件，而且需要多条记录来完成。

1. `_row_`: 标识是目标函数还是约束条件;
2. `_coef_`: 定义变量系数（但在第一条记录中，对应于目标函数，其值缺失，用“.”表示；在每个约束条件的第一条记录中，其值为等式的右项值）;
3. `_type_`: 在目标函数中的第一行取值为“MIN”或“MAX”，在每个约束条件中的第一行取值为“LE”、“LT”、“EQ”、“GE”或“GT”，其它地方都是缺失值，用“.”表示。
4. `_col_`: 变量名（但在目标函数中的第一行时为缺失值，在约束条件中的第一行取值为“\_RHS\_”）。

从上面的小程序中可以看到，如果是目标函数，第一条记录总是用于定义目标函数的类型（MIN 或 MAX）；如果是约束条件，其相应的第一条记录总是用于定义等式的右项值，然后才逐个定义变量系数。因此，定义目标函数或每个约束条件，通常要比变量数多一条记录。

计算浪费性通勤的 SAS 编码要稍稍复杂些，主要是用到一些 DO 循环语句。为了方便，光盘中提供了第 10.2 节中所生成的两个结果文件：

1. `urbtaz.txt` 包含三列数据，即：TAZ 标识码，居住地上班人数，及就业地上班人数，共有 991 个 TAZ 区的记录数；
2. `odtime.txt` 包含 3 列数据，即：居住地的 TAZ 编码，就业地的 TAZ 编码，及两者之间的通勤时间。

附录 10B 中（光盘中含）的 `LP.SAS` 程序用于求解线性规划问题，在程序的关键步骤中加有注释。当程序读入就业地、居住地和二者通勤时间数据文件后，程序首先根据文件内容

定义就业地上班人数这个约束条件，然后定义居住地上班人数这个约束条件，最后定义目标函数。运行程序可以产生一个最优解，分析结果（最优的通勤分配）存放在外部文件 `min_com.txt` 中，其中包含三列数据，即起始 TAZ 区标识码，终止 TAZ 区标识码，起止点间的通勤时间。

上述程序计算出的最小通勤时间总计为 1,939,404.01 分，也就是平均每个就业工人的通勤时间为 3.90 分(总时间除以通勤人数 497,588)。据 1990 年统计资料，实际上研究区的平均通勤时间为 20.40 分（Wang, 2001b, 173 页）。对比这两个数据，可以发现惊人的浪费性通勤，高达 80.89%。进一步检查最优解的结果文件 `min_com.txt`，可以发现许多条记录的起点 TAZ 区标识码与止点 TAZ 区标识码是相同的，这说明最优解方案中，许多通勤路径是从一个 TAZ 出发又回到同一个 TAZ（上班就在同一区内，或称“区内通勤”）。从研究 1990 年 CTPP 第三部分关于实际通勤的资料来看，该研究区中有 346 个 TAZ 区确实存在通勤起止点是同一区，算得的平均区内通勤时间达 16.3 分。在俄亥俄州的另一城市克里夫兰，算得的平均区内通勤时间为 11.3 分，表明区内通勤时间不宜忽略（Wang, 2003, 258 页）。案例中区内通勤时间的估算，多小于 1 分，很难反映真实情况，显示了 GIS 以点带面（假定区内所有的人都集中在其几何中心）的局限性。上述讨论说明，测算浪费性通勤的结果或多或少，往往是由数据中地理单元的大小、通勤时间估算上的误差引起的，理论和方法上并无多大争议。

## 10.3 整数规划与区位优化问题

### 10.3.1 整数规划通用形式和解

线性规划问题中的一些决策变量只能取整数，那么这类问题就成了整数规划，又分为纯整数线性规划和混合型整数线性规划

决策变量全部都是整数的线性规划称为整数线性规划（Integer Linear Programming 或简称 ILP），其公式表示如下。

求  $\sum_{j=1}^n c_j x_j$  的最大值,

约束条件为:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \text{ 其中 } i \in \{1, 2, \dots, m\},$$

整型变量  $x_j \geq 0$ , 其中  $j \in \{1, 2, \dots, n\}$ .

如果线性规划决策变量中的部分变量是整数, 其他为常规的实数, 则称为混合型整数线性规划 (Mixed Integer Linear Programming 或简称 MILP), 公式表示如下。

求  $\sum_{j=1}^n c_j x_j + \sum_{k=1}^p d_k y_k$  的最大值

约束条件为:

$$\sum_{j=1}^n a_{ij} x_j + \sum_{k=1}^p g_{ik} y_k \leq b_i, \text{ 其中 } i \in \{1, 2, \dots, m\},$$

整数变量  $x_j \geq 0$ , 其中  $j \in \{1, 2, \dots, n\}$ ,

实数变量  $y_k \geq 0$ , 其中  $k \in \{1, 2, \dots, p\}$ .

也许有人认为, 求解 ILP 或 MILP 问题很简单, 只要利用原来的方法求解一般的线性规划问题, 然后对解得的决策变量取整数, 就可以得到 ILP 或 MILP 的解。事实上, 很多时候这种取整的做法并不能得到最优解 (Coppins, 1981, 399 页)。尤其是在决策变量限于 0 和 1 的时候, 取整方法根本无法解决这种 0-1 线性规划问题 (0-1 binary programming problem)。0-1 线性规划在科研和管理上用途广泛, 区位优化 (location-allocation) 问题就是典型的 0-1 线性规划问题。实际上, 整数规划问题 (ILP 和 MILP) 需要一些特别的求解方法, 如割平面法和分枝定界法, 后者较为普及。下面概略地介绍一下分枝定界法。

1. 假如目标函数类型是求最大值。先求取一个可能解  $f_L$  作为目标函数的底(下界), 同时

作为可能解的子集。

2. 在可能解的子集中（在分枝时可能会产生多个可能解），将其中一个分解成两个或多个解子集。
3. 计算每个解子集对应的目标函数值，取其中对应着最大值的解子集  $f_U$  将作为顶（上界）。
4. 符合以下条件时，删除分枝的解子集。如果  $f_U < f_L$ ，或者分枝没有可能解，或者是在这个分枝中已求得最优解，此时要用这个上界  $f_U$  来代替原来的下界  $f_L$  的值。
5. 再没有分枝时程序退出，否则，转到第 2 步。

### 10.3.2 区位优化问题

我们在这里选取三个经典的区位优化问题来讲解 ILP 的建模。

在许多候选的服务设施中，如何选择一定数目的服务设施，以达到需求点到服务点的总距离或时间最小？这就是 p-中位问题（p-median problem）（ReVelle and Swain, 1970），其公式如下。

$$\text{求 } Z = \sum \sum a_i d_{ij} x_{ij} \text{ 最小值}$$

约束条件为：

$$x_{ij} \leq x_{jj} \quad \forall i, j, i \neq j \quad (\text{每个需求点的分配受限于服务设施是否被选})$$

$$\sum_{j=1}^m x_{ij} = 1 \quad \forall i \quad (\text{每个需求点必须配给某个服务设施})$$

$$\sum_{j=1}^m x_{jj} = p \quad \forall j \quad (\text{选用的服务设施总数刚好是 } p)$$

$$x_{ij} = 0, 1 \quad \forall i, j \quad (\text{标志哪个需求点是否配给哪个服务设施})$$



这里，

$i$  是需求点编码 ( $i = 1, 2, \dots, n$ )

$j$  是候选服务点编码 ( $j = 1, 2, \dots, m$ )

$p$  是要选取的服务点数目

$a_i$  是需求点  $i$  的总需求

$d_{ij}$  是需求点  $i$  到服务点  $j$  之间的距离或时间

$x_{ij} = 1$  表示第  $j$  点的服务设施覆盖第  $i$  个需求点，否则为 0。

第二个模型为最少服务点问题 (location set covering problem 或简称 LSCP)，即给定需求区域，求解覆盖整个区域所需的最少服务点数目 (Toregas and ReVelle, 1972)。模型公式如下。

$$\text{求 } Z = \sum_{j=1}^m x_j \text{ 的最小值}$$

约束条件为：

$$\sum_{j=1}^{N_i} x_j \geq 1 \quad \forall i \quad (\text{至少有一个营业的服务设施在需求点的可达半径之内})$$

$$x_j = 0, 1 \quad \forall j \quad (\text{一个候选的服务设施要么营业，要么关闭})$$

这里

$N_i$  为在需求点  $i$  可达半径之内的服务设施总数

$x_j = 1$  表示候选服务设施  $j$  是营业的，否则为 0。

$i, j, m$  和  $n$  与上面中心区位问题的定义相同。

第三个模型为最大服务面问题 (maximum covering location problem 或简称 MCLP)，即给定  $p$  个服务设施，如何分配服务设施，以使一定距离 (时间) 阈值范围内被服务的需求总量最大 (Church and ReVelle, 1974)。模型公式如下。

求  $Z = \sum_{i=1}^n a_i y_i$  的最小值

约束条件:

$$\sum_{j=1}^{N_i} x_j + y_i \geq 1 \quad \forall i \quad (\text{需求点必须在至少一个营业的服务设施可达半径内, 否}$$

则其需求不能被满足)

$$\sum_{j=1}^m x_j = p \quad (\text{营业的服务设施数目为 } p)$$

$$x_j = 0, 1 \quad \forall j \quad (\text{候选的服务设施要么营业, 要么关闭})$$

$$y_i = 0, 1 \quad \forall i \quad (\text{需求点要么得到了服务, 要么没得到服务})$$

这里

$y_i = 1$  表示需求点  $i$  没得到服务,  $= 0$  则表示得到服务 (注意目标函数

已转换为最小值, 所以  $y_i$  值的解释与传统意义正好相反)

$i, j, m, n, a_i$  和  $p$  的定义同上面的  $p$ -中位( $p$ -median)问题

$N_i$  和  $x_j$  定义同上面的 LSCP 问题

上述三类问题都可以利用 SAS 软件包求解, 只是需要花一些时间来编程。如果利用 ArcInfo 工作站版中的内嵌模块 (目前 ArcGIS 桌面版软件不支持), 那么求解这些问题就比较方便。对于  $p$ -中位 ( $p$ -median) 问题, 可以用 ArcInfo 软件中的 MINDISTANCE 模块求解。简单的中心区位问题中, 目标函数是求交通总量最小, 在选择服务设施时缺少最大距离的限制, 某个服务设施到某个需求点的距离有可能很远, 在实践中并不一定符合要求。我们增加一个约束条件, 要求需求点必须在服务设施的某个距离或时间以内, 这样修正后的中心区位问题就变成了最大服务半径内的  $p$ -中位问题 ( $p$ -median problem with a maximum distance constraint), 参见有关文献 (Khumawala, 1973; Hillsman and Rushton, 1975)。在 ArcInfo 软

件中对应的模块为 MINDISTANCE (constrained)。目前 ArcInfo 软件中还没有内嵌的模块可以用来处理最少服务点问题（简称 LSCP）。求解最大服务面问题（简称 MCLP）的 ArcInfo 模块为 MAXCOVER。类似地，在 MCLP 模型中可以增加约束条件，这里定义一个比原来距离阈值大的第二阈值，使原先那些没有获得服务的需求点落在第二阈值范围内，这样修订后的模型叫最大服务半径内的最大服务面问题（MCLP with mandatory closeness constraints），参见 Church 和 ReVelle（1974），可以利用 ArcInfo 的模块 MAXCOVER (constrained) 求解这类问题。表 10.1 总结了上述三种模型及对应的 ArcInfo 模块。有关利用 ArcInfo 软件分析区位优化问题的详细内容，请查阅在线帮助系统中从“Working with Linear Features”到“Location-Allocation”等主题的相关内容。

目前 ArcGIS 软件中有关区位优化方面的功能模块还比较有限，模块中也不能定义额外的约束条件。在许多应用研究中如果遇到一些更复杂的整数规划问题，就需要用 SAS 或其它专业软件来解决。请参阅有关文献（Curtin et al., 2005）关于美国达拉斯警察巡逻区最优分配的研究成果。

#### 10.4 案例研究 10B: 美国俄亥俄州凯霍加县医疗服务的区位优化

203

下面通过一个假想的案例，来讲解如何在 ArcInfo 中运用区位优化模型。美国俄亥俄州某县（就是克里夫兰市所在的凯霍加县）卫生局计划选择五个临时诊所来为儿童和老年人接种流感疫苗，这五个临时诊所要从现有的 21 个医院中挑选。规划的目标是服务对象（儿童和老年人）从居住地到临时诊所的总距离或时间最小，这就是一个  $p$ -中位（ $p$ -median）问题。下面讲解中心区位问题在 ArcInfo 平台下的求解方法，其他区位优化模型的求解与其相似。

案例分析的相关资料如下：

1. 多边形图层 cuyatrt 为该县的人口普查资料
2. shape 格式文件 tgr390351ka 包含全县的路网

3. 以逗号作为定界符的文本文件 `cuya_hosp.csv` 包含现有 21 个候选医院的地址

图层 `cuyatrt` 属性表中的属性项 `age_under5` 代表年龄小于 5 岁的儿童，属性项 `age_65_up` 代表年龄大于 65 岁的老人，这些资料来自于 2000 年的人口普查数据库。医院文件 `cuya_hosp.csv` 中的地址和名称来自于俄亥俄州医院协会的网站（<http://www.ohanet.org/research/dataresources>）。

这个规划问题是从 21 个医院中挑选五个作为临时诊所，最有效地服务儿童和老年人，具体目标函数是需求到服务的距离或时间总和最小。`ArcInfo` 软件中的 `MINDISTANCE` 模块可用来求解这个规划问题。实际应用中，需求区位和服务候选区位可用 GIS 中的点、面、路网结点等来表示。如果以点或面来表示需求与服务设施区位时，其间的交通阻滞 `ArcInfo` 简化为欧氏距离；如果有交通道路系统的数据，`ArcInfo` 可用路网的结点来近似需求与服务设施的区位，计算它们之间的交通阻滞可以由路网距离或时间来衡量。前者称为基于面（点）的分析，后者称为基于路网的分析。基于点的分析方法和基于面的分析方法类似，这里不再重复。下面将分别讲解基于面和基于路网的分析方法。

#### 10.4.1 第一部分：基于面的分析方法

为方便起见，在 `ArcInfo` 区位优化分析模块中，假设需求（居民点）和供给（如候选医院）这两方面的信息都在同一个图层。在基于面的分析方法中，需求地与候选服务点的属性变量都保存在多边形图层的 `PAT` 文件中。在本例中，由于医院与其最近邻人口普查区（以几何中心代表）的距离很小，可以忽略不计，我们可以用这一人口普查区的重心近似代表医院的区位。这样，需求（人口普查区）和供给（候选医院）这二者的信息就同时放到多边形图层 `cuyatrt` 中。具体步骤如下。

##### 1. 医院地理编码

在 `ArcCatalog` 中启动“地理编码”功能，操作方法为：选择 `Address Locators > Create New`

Address Locator > 选 US Streets with Zone (File) > 命名这个新的地址匹配器 (address locator) 为 Cuyahoga; 在 Primary table, Reference data 栏目下, 选择文件 tgr390351ka, 其他项选用缺省值即可。

在 ArcMap 模块中进行“地址匹配”, 操作方法为: 选 Tools > Geocoding > Geocode Address。选凯霍加 为地址匹配器, 选 cuya\_hosp.csv 为地址表, 执行结果存放在 shape 文件 hosp0, 然后通过地图投影变换模块 (projection)将这个 shape 文件投影到图层 cuyatrt 的坐标系中 (参见第一章第 1.2 节中的第 2 步), 新的图层文件名为 hosp。

## 2. 定义需求值

打开图层 cuyatrt 的属性表, 加入一个新的属性项 patient, 根据公式  $\text{patient} = \text{age\_under5} + \text{age\_65\_up}$  计算它的值, 这个属性项定义了每个人口普查区的总需求。

## 3. 将医院信息转到其最近的人口普查区

这个操作在 ArcGIS 上分四步完成:

(1) 使用 feature to point 工具, 将多边形图层 cuyatrt 转为点图层 cuyatrtpt。

(2) 使用 near 工具, 选择 hosp 为 input feature, cuyatrtpt 为 near feature, 找到离每个医院最近的人口普查区。

(3) 把 cuyatrtpt 属性表连接到 hosp 属性表上去。连接以后的 hosp 属性表中, 加上了离其最近人口普查区的标识码 (例如 STFID)。将这个新 hosp 属性表输出另存为一个新表 hosptrt。

(4) 根据关联码 STFID 将表 hosptrt 连接到人口普查区图层 cuyatrt。

注意, 在利用 near 工具处理得到的结果中, 有两个医院 (Rainbow Babies and Children's Hospital 和 University Hospitals of Cleveland) 彼此非常接近, 二者邻近于同一个人口普查区。这意味着这两个医院只能合并为一个候选对象, 合并以后的候选区位从原来的 21 个减为 20 个。

#### 4. 定义区位优化的候选对象

在 `cuyatrt` 图层中加入一个新的属性项 `clinic`，根据选择条件 `hosp:id>0`（只有靠近医院的人口普查区满足这一条件）选出代表医院区位的人口普查区，将其赋值为 `clinic = 1`。属性项 `clinic` 值为 0 时标志该区不是候选区位，为 1 时标志是候选区位，为 2 时标志是已经选定的服务设施点。

#### 5. 执行基于点的区位优化分析

在 `ArcInfo` 工作站版的软件环境中，进行区位优化操作。首先，将存放上述文件的目录指定为当前工作目录，在 `ArcInfo` 命令行中键入 `ap` 命令来起动 `ArcPlot` 模块，在 `ArcPlot` 命令提示符方式下执行如下命令：

```
locatecandidates cuyatrt poly patient clinic
locatecriteria mindistance
locateallocate outalloc1 outcent1 outglob1 5
```

命令 `locatecandidates` 设置区位优化分析所需要的环境参数，包括（1）定义分析图层名为 `cuyatrt`，（2）参数 `poly` 定义当前分析的图层类型为多边形，若是点图层则使用参数 `point`，（3）指定属性表中的 `patient` 项为需求参数，（4）指定属性项 `clinic` 为候选服务区位。命令 `locatecriteria` 定义使用区位优化分析模块 `MINDISTANCE` 来求解 `p-median` 问题。命令 `locateallocate` 执行区位优化分析，其中，（1）第一个参数指定输出结果为 `INFO` 文件 `outalloc1`，它也是一个记录列表文件，包含需求人口普查区(编号)、需求量、最近的候选区位(编号)及距离、次近的候选区位(编号)及距离等内容；（2）第二个参数指定另一个输出结果文件为 `outcent1`，它是一个记录列表文件，包含有被选定的候选区位编号和相关信息；（3）第三个参数指定输出结果中的第三个 `INFO` 文件 `outglob1`，其中汇总有简要的分析结果；（4）第四个参数指定要选 5 个最优服务点（临时诊所）。

图 10.2 总结了基于面的区位优化分析中用到的输入文件和输出的结果文件。

## 6. 分析结果显示

(1) 通过关联码 `cuyatrt#`，将分析结果中的表 `outalloc1` 和表 `outcent1` 连接到图层 `cuyatrt` 的属性表中。`outalloc1` 表中的 `site1` 项表示各人口普查区是分配给哪一个临时诊所来服务的。(2) 用 `dissolve` 工具，根据属性项 `site1` 对图层 `cuyatrt` 中的人口普查区分组合并，可以生成 5 个临时诊所的服务区分布图，如图 10.3 所示。值得一提的是，有的人口普查区需求量 (`patient` 项) 为 0，并不影响也不出现在区位优化分析的结果中，可以把它们划在就近的诊所服务区内，以保证服务区分布图的完整。`outcent1` 表中的 `totdemand` 项为每个诊所服务区内总需求 (就是 `patient` 项的总和)。分析结果汇总成表 10.2。

### 10.4.2 第二部分：基于路网的分析方法

207

与基于面的分析方法相似，基于路网的分析方法同样要将需求和候选的供给区位这两种信息存贮在同一个图层中。在本案例中，需求信息存放在路网结点属性表 (`NAT` 表) 中，候选供给区位信息存放在文件 `CENTERS` 中 (其中标识码与路网结点属性表 `NAT` 一致)。基于路网的区位优化分析利用路网距离或时间来衡量交通阻滞。

#### 1. 准备路网资料

启动 `ArcToolbox` 模块, (1) 用 `cuyatrt` 所定义的地理坐标系将 `shape` 文件 `tgr390351ka` 投影变换到新的图层 `cuyard0`, (2) 用 `Conversion Tools>To Coverage>Feature Class To Coverage` 工具将 `shape` 文件 `cuyard0` 转换为 `Coverage` 格式的图层 `cuyard`, (3) 用 `Coverage Tools>Data Management>Topology>Build` 工具，创建图层 `cuyard` 中线和结点的拓扑关系。查看一下图层 `cuyard` 中点和线的属性文件 `cuyard.aat` 和 `cuyard.nat` (在将图层 `cuyard` 加入当前图框时，用鼠标双击 `cuyard` 就可选择是添加其中的 `arc` 图层还是 `node` 图层)。

## 2. 定义候选区位信息的 INFO 文件

与第一部分中的第 3 步相似，这一步的主要工作是将候选医院指派给最近的结点，并以此定义候选区位，具体操作在 ArcGIS 中分三步完成。(1) 用 **near** 工具来找到离医院最近的路网结点（选 **hosp** 为 **Input Feature**，**cuyard (node)**为 **Near Feature**），(2) 将 **cuyard** 结点属性表（**cuyard.nat**）连接到 **hosp** 的属性表中（这样 **hosp** 的属性表就增加了离其最近结点的编号信息），并将结果输出为一个新文件 **hospsnode.dbf**，(3) 在文件 **hospsnode.dbf** 中增加一个属性项 **clinic**，设值为 **clinic = 1**。与第一部分不同的是，这里每个医院都对应着一个唯一的结点，因此，共有 21 个候选区位（第一部分有两个医院合并，只有 20 个候选区位）。

定义 **INFO** 文件需要在 **ArcInfo** 工件站版的软件环境下完成。在 **ArcInfo** 命令行提示符下，(1) 键入命令 **dbaseinfo hospsnode.dbf centers**，将 **dBase** 文件 **hospsnode.dbf** 转换为 **INFO** 文件 **centers**，(2) 将 **INFO** 文件 **centers** 中的 **cuyard\_**和 **cuyard\_id** 这两个变量分别重新命名为 **cuyard#**和 **cuyard-id**，(3) 舍弃其他项，仅仅保留 **cuyard#**、**cuyard-id**、**clinic** 这三个变量。

## 3. 在路网属性表 NAT 中定义需求量

与上述第 2 步相似，这一步要将人口普查区的需求信息转移到离其最近的路网结点上，具体操作在 ArcGIS 中分两步完成。(1) 用 **near** 工具来找到离人口普查区最近的结点（选 **cuyatrpt** 为 **Input Feature**，**cuyard (node)**为 **Near Feature**），(2) 将 **cuyard** 结点属性表（**cuyard.nat**）连接到 **cuyatrpt** 的属性表中（这样 **cuyatrpt** 的属性表就增加了离其最近结点的编号信息），结果输出为一个新的文件 **trtnode.dbf**。

同样地，在 **ArcInfo** 工件站版的软件环境下，(1) 键入命令 **dbaseinfo trtnode.dbf demand**，将 **dBase** 文件 **trtnode.dbf** 变为 **INFO** 文件 **demand**，(2) 将 **INFO** 文件 **demand** 中的 **cuyard\_**和 **cuyard\_id** 两个变量分别重命名为 **cuyard#**和 **cuyard-id**，(3) 舍弃其



他项，仅仅保留 `cuyard#`、`cuyard-id`、`STFID` 和 `patient` 这几个变量，(4)键入命令  
`joinitem cuyard.nat demand cuyard.nat cuyard#` 将文件 `demand` 连接到路网  
 属性表 `cuyard.nat`。

#### 4. 执行基于路网的区位优化分析

在 `ArcInfo` 命令行中键入 `ap` 命令启动 `ArcPlot` 模块，然后，在 `ArcPlot` 提示符下键入如下  
 命令：

```
netcover cuyard cuyaroute
demand # patient
centers centers
locatecandidates centers clinic
locatecriteria mindistance
locateallocate outalloc2 outcent2 outglob2 5
```

在命令 `netcover` 中，定义 `cuyard` 图层为区位优化分析的路网系统，产生的路径文件为 `cuyaroute`。在命令 `demand` 中，用符号 `#` 跳过选项 “`arc_demand_item`” 到选项 “`node_demand_item`”，由路网结点属性表 `cuyard.nat` 中的属性项 `patient` 定义。命令 `centers` 定义 `INFO` 文件 `centers` 为候选区位。命令 `locatecandidates centers` 用 `CENTERS` 文件中的 `clinic` 项设置区位优化分析的环境。其他命令 `locatecriteria` 和 `locateallocate` 与第一部分（基于面的区位优化分析）相同。

图 10.4 以图解方式说明了在基于网络的区位优化分析中，相关的输入与输出文件。

#### 5. 分析结果显示

启动 `ArcMap`，(1) 根据关联码 `cuyard#`，把表 `outalloc2` 和表 `outcent2` 连接到 `INFO` 文件 `demand` 中去，将合并后的结果输出为外部文件 `demand.dbf`，(2) 根据关联码 `STFID`，将文件 `demand.dbf` 连接到图层 `cuyatrt` 的属性表中，(3) 用 `dissolve` 工具，根据  
 属性项 `site1` 对图层 `cuyatrt` 中的人口普查区分组合并，可以生成 5 个临时诊所的服务区

分布图，见图 10.5。分析结果汇总在表 12.2。

比较图 10.3 和图 10.5 可以发现，基于面和基于路网的区位优势法选择了同样的区位来设这五个诊所，但在服务区的划分上略有差别，后者反映了路网格局的影响。例如，沿湖的州际高速公路（见图 Figure 10.6 中的 I-90）使得东北角的一些地区到 Meridia Huron 医院更方便，而不是到直线距离较近的 Meridia Hillcrest 医院。东边的州级公路 SR91 使本县东南角上的地区更容易到 Meridia Hillcrest 医院，而不是 Marymount 医院。同样原理，Parma Community General 医院的服务区范围向北扩展，但在西南和东南两角收缩了。

## 10.5 总结

线性规划技术广泛地应用在科研、工程、社会经济规划、区位分析和其它领域。本章首先介绍了单纯形法，及其在 SAS 软件支持下的线性规划求解方法。利用 SAS（OR 模块）中的 LP 程序可以求解一般型线性规划、整数型线性规划和混合型线性规划等多种问题。在编写 SAS 的线性规划程序时，整型变量通过 TYPE 语句定义为 INTEGER，0-1 变量通过 TYPE 语句定义为 BINARY（虽然第 10.2 节中讲解的案例不涉及整型变量）。当涉及到大型矩阵编码时，最好使用松散式来编写程序代码。SAS 软件中还提供有 ASSIGN 程序（用于求解分配委派问题）、CPM 程序（用于求解工程时间安排）、GANTT 程序（用来绘制 Grantt 图）、NETDRAW 程序（用于绘制网络图）、NETFLOW 程序（用于网络物流规划）、TRANS 程序（用于求解交通问题）等许多专业工具，用来求解更专业性的线性规划问题。

本章第一个分析案例是浪费性通勤的计算，基于 SAS 软件讲解了求解线性规划问题的具体过程。在这里，目标函数是整个城市通勤距离（时间）的总和最小，而求得的最优解中，许多通勤路径是居住地和工作地在同一区内，从而通勤量较小。显然，这种“最优通勤量”与实际通勤量相比，相差悬殊，存在很大的浪费性通勤。准确估算区内距离（时间）是十分困难的，除非我们有居住地和工作地精确到点的街道地址，用 GIS 以点带面的方法估算出来的

区内通勤时间总会有较大误差。浪费性通勤这个具体问题的研究也许意义有限，但问题的研究引发了与交通通勤相关的其它问题，比如“空间错位”（spatial mismatch）和弱势群体是否享有社会公正（见 Kain, 2004）等敏感的公共政策问题。

区位优化模型是线性规划在区位分析中的一种应用。本章中第二个案例是  $p$ -中位优化（ $p$ -median）模型的一个应用实例，即医疗卫生设施的最优区位优化问题。通过案例分析，我们熟悉了如何利用 ArcInfo 软件的内置模块来求解这类问题，具体分为基于面（点）和基于路网两种方法。基于面（点）的方法使用欧氏（直线）距离来度量供需点之间的交通阻滞，在 ArcInfo 中求解过程相对容易一些。基于路网的方法使用通行距离（时间）来度量交通阻滞，更准确地反映了实际情况，分析时需要提前准备交通路网数据。两种方法都要求将需求与供给这两种信息存放在同一个图层中。在基于面（点）的方法中，既可以将供给区位图层和需求区位图层合并成一个新图层的方法来实现，也可以通过在供给图层中找到最近的需求区位，然后用这些最近的需求区位来代替供给点的方法来实现（第 10.4 节案例研究用的是后者）。在基于路网的方法中，需求区位和供给区位，都是通过寻找其最近的路网结点，然后将它们指派到这些结点上，从而将两者都转换到同一个图层上。利用 ArcInfo 可以很方便地求解区位优化问题，分析结果可以直接在 ArcGIS 中制图输出，可惜 ArcInfo 相关的模块中，用户不能自己定义一些较为复杂的约束条件。求解更为复杂的线性规划问题需要在 SAS 下编程，或借助于更专业的软件包来实现。

#### 附录 10A. 哈密尔顿的浪费性通勤模型

经济学家通常要在一些假设的基础上来构造模型。虽然假设离真实世界有一定的距离，但有利于简化模型，突出某些本质性的问题。哈密尔顿的浪费性通勤模型（Hamilton, 1982）也不例外，它是将城市简化为单中心（类似于第六章中的城市经济模型）。大家应该注意到，在二十世纪八十年代前，城市内部就业分布的资料是很难拿到的，另外，GIS 技术当时还处

于开发阶段。

首先，在一个单中心的城市，假设就业都集中在市中心的 CBD 地区。假定人口分布密度函数为  $P(x)$ ，其中  $x$  为到 CBD 的距离，可以计算出以 CBD 为圆心、距离为  $x$  的同心圆环的面积为  $2\pi x dx$ ，相应的人口数为  $2\pi x P(x) dx$ 。通过积分可以计算出在以 CBD 为中心、 $R$  为半径的市区内，整个市区人口 ( $N$ ) 产生的通勤距离总量  $D$  为：

$$D = \int_0^R x(2\pi x P(x)) dx = 2\pi \int_0^R x^2 P(x) dx$$

这样，每个人的平均通勤距离  $A$  为：

$$A = \frac{D}{N} = \frac{2\pi}{N} \int_0^R x^2 P(x) dx \quad (\text{A10.1})$$

现在假设就业分散地分布在整个城市中，符合分布函数  $E(x)$ 。哈密尔顿相信分散的就业模型比单中心模型更加符合现实情况。与式 A10.1 相似，到 CBD 的平均就业距离为：

$$B = \frac{2\pi}{J} \int_0^R x^2 E(x) dx \quad (\text{A10.2})$$

这里， $J$  是城市中总的就业人数。

假如所有居民都可以自由地交换居住地以减少通勤时间，则现在的问题就是怎样才能使总的通勤时间最小。注意到就业分布比居民人口分布更为集中，上班的通勤方向总是从外往 CBD 方向，并选择就近的就业机会。正如哈密尔顿所言，与单中心城市相比，“就业机会的分散节省了上班族的通勤距离，节省的距离正好等于工作地到 CBD 的距离”（Hamilton, 1982, 1040 页）。因此，最优通勤，或称必要通勤、最小通勤，就是居民离 CBD 的平均距离 ( $A$ )

214

与就业离 CBD 的平均距离 ( $B$ ) 之差：

$$C = A - B = \frac{2\pi P_0}{N} \int_0^R x^2 e^{-tx} dx - \frac{2\pi E_0}{J} \int_0^R x^2 e^{-rx} dx \quad (\text{A10.3})$$

这里人口分布密度函数和就业分布密度函数都假定为指数函数，也就是， $P(x) = P_0 e^{-tx}$ （

见第六章的方程 6.1）， $E(x) = E_0 e^{-rx}$ 。

计算 A10.3 中的定积分可以得到下式：

$$C = -\frac{2\pi P_0}{tN} R^{-2} e^{-tR} + \frac{2}{t} + \frac{2\pi E_0}{rJ} R^{-2} e^{-rR} - \frac{2}{r}$$

哈密尔顿研究了美国 14 个不同大小的城市，发现最优通勤只占实际通勤的 13%，剩余的 87% 都是浪费性通勤。为了更深入地阐述他的观点，他假设人们随机地选择居住地和就业地，发现那样的通勤距离只是比实际通勤多出 25%。也就是说，人们在居住地和就业地选址时更接近于随机行为，离最优通勤的理想状态相距甚远。

为什么实际通勤量要比哈密尔顿模型预测的最优通勤要多得多呢？其中原因很多，有些原因哈密尔顿自己已经认识到了，比如假设的就业和居民分布函数以及无数的放射状城市路网都离现实相距甚远（即二者之间存在偏差）。另外，当人们更换工作时，不一定马上就迁往到离工作地点最近的区位，因为还涉及搬迁成本（具体费用及一些感情因素）和其它考虑。显然，住自己房子的人要比租房子住的人搬家成本更高，因此，居民是否拥有住房产权也会影响人们通勤是否优化的可能性。如果家中有多人工作，除非刚好都在同一个地点工作，否则，不可能在住宅选址上，为家庭中的每个工作成员都优化通勤。更为重要的是，居民选择居所时不单会考虑上班远近的问题，还有其他方面的因素，如购物、服务、娱乐、休闲等设施是否方便？学校和公共设施的质量如何？邻里是否和睦？小区是否安全？在研究城市内部通勤差异时，有的研究者已经考虑到了这些因素，参见有关文献（Wang, 2001b; Shen, 2000）。

## 附录 10B. 浪费性通勤测算的 SAS 程序

```

/* LP.sas minimizes total commute time in Columbus, Ohio.
   By Fahui Wang on 4-14-2005 */

/* Input the data of resident workers & jobs */
data study;
  infile 'c:\gis_quant_book\projects\columbus\urbtaz.txt';
  input taz $1-6 work emp; /*TAZ codes, # Workers, # jobs*/
  proc sort; by taz;
data work; set study (rename=(taz=tazr)); if work>0;
  oindex+1; /*Create an index for origin TAZs */
data emp; set study (rename=(taz=tazw)); if emp>0;
  dindex+1; /*Create an index for destination TAZs */
/* Input the data of O-D commute time */
data netdist0;
  infile 'c:\gis_quant_book\projects\columbus\odtime.txt';
  input tazr $1-6 tazw $9-14 @15 d; /*from_taz, to_taz, time*/
  proc sort; by tazw;
data netdist1; /*attach index for destination taz */
  merge emp(in=a) netdist0; by tazw; if a;
  proc sort; by tazr;
data netdist2; /*attach index for origin taz */
  merge work(in=a) netdist1; by tazr; if a;
  route=oindex*1000+dindex; /*Create unique code for a route*/
  proc sort; by route;

/* Build the LP model in sparse format */
data model;
  length _type_ $8 _row_ $8 _col_ $8;
  keep _type_ _row_ _col_ _coef_; /*four variables needed */
NI=812; /*total number of origin TAZs */
NJ=931; /*total number of destination TAZs */
/* Create the Constraints on Jobs */
Do j=1 to NJ; set emp;
/* 1st entry defines the upper bound (#jobs) for a TAZ */
_row_='EMP'||put(j,3.); /* Increase the space limit "3" if >999
TAZs */
_type_='LE';
_col_='_RHS_';
_coef_=emp;
output;
/* the following defines variables & coefficients in the same row */
_type_=' ';
Do I=1 to NI;
  if emp~=. then do; /* for non-zero emp TAZs only */
    _col_='X'||put(i,3.)||put(j,3.); /* Xij */
    _coef_=1.0; /* all coefficients are 1 */
    output;
  end;
end;
end;

/* Create the Constraints on Resident Workers */
Do i=1 to Ni;

```

215

216

```

set work;
_row_='WRK' || put(i,3.);
_type_='EQ'; /* All resident workers must be assigned */
              /* Note total resident workers < total jobs */
_col_='_RHS_';
_coef_=work;
output;
_type_=' ';
Do j=1 to Nj;
  if work~=. then do;
    _col_='X' || put(i,3.) || put(j,3.);
    _coef_=1.0;
    output;
  end;
end;
end;

/* Create the objective function */
_row_='OBJ';
Do I=1 to NI;
  Do J= 1 to NJ;
    _type_=' ';
    set netdist2;
    if d~=. then do;
      _col_='X' || put(i,3.) || put(j,3.);
      _coef_=D;
      output;
    end;
  end;
end;
_type_='MIN';
_col_=' ';
_coef_=.;
output;

/* Run the LP Problem */
proc lp sparsedata
  primalout=result noprint time=60000 maxit1=5000 maxit2=50000;
  reset time=60000 maxit1=5000 maxit2=50000;

data result; set result;
  if _value_>0 and _price_>0;
/* Save the result to an external file for review */
data _null_; set result;
  file 'c:\gis_quant_book\projects\columbus\junk.txt';
  put _var_ _value_;
/* convert the X variable to From_TAZ and To_TAZ index codes */
data junk;
  infile 'c:\gis_quant_book\projects\columbus\junk.txt';
  input oindex 2-4 dindex 5-7 Ncom ;
  route=oindex*1000+dindex;
  proc sort; by route;

/* attach the From_TAZ and To_TAZ codes, #commuters and time

```

```
        and save the result to an external file          */
data f;
  merge netdist2 junk(in=a); by route; if a;
  file 'c:\gis_quant_book\projects\columbus\min_com.txt';
  put tazr $1-6 tazw $8-13 Ncom 15-21 @23 d 12.5;
run;
```



表 10.1 区位优化模型

模型	目标函数	约束条件	ArcInfo 模块
I. $p$ -中位优化 ( $p$ -median) 问题	总距离 (时间) 最小	分配 $p$ 个服务设施; 满足所有需求	MINDISTANCE
II. 最大服务半径内的 $p$ -中位问题 ( $p$ -median with a max distance constraint)	总距离 (时间) 最小	所有模型 I 的约束条件; 需求到服务设施的距离 (时间) 在规定值之内	MINDISTANCE (constrained)
III. 最少服务点模型 (location set covering problem 或 LSCP)	服务设施点最少	满足所有需求; 需求到服务设施的距离 (时间) 在规定值之内	无
IV. 最大服务面模型 (maximum covering location problem 或 MCLP)	满足的需求量最大	分配 $p$ 个服务设施; 如需求到服务设施的距离 (时间) 在规定值之内, 需求得以满足	MAXCOVER
V. 最大服务半径内的最大服务面模型 (MCLP with mandatory closeness constraints)	满足的需求量最大	所有模型 IV 的约束条件; 需求到服务设施的距离 (时间) 在第二个规定值 (比第一个规定值大) 之内	MAXCOVER (constrained)

表 10.2 区位优化案例分析结果

诊所选址位置	基于面的区位优化服务区		基于路网的区位优化服务区	
	普查区数	服务总人数	普查区数	服务总人数
Meridia Hillcrest 医院	34	30,159	32	28,428
Meridia Huron 医院	168	72,311	168	79,005
Fairview 医院	120	78,241	109	75,110
Marymount 医院	100	60,041	96	55,064
Parma Community General 医院	80	68,071	92	71,213