

TSPLIB 95

格哈德Reinelt

海德堡大学数学与数学研究所,

2004

D-69120 海德堡 Gerhard.Reinelt@IWR.

Uni-Heidelberg.DE

TSPLIB是来自各种来源和类型的TSP(及相关问题)的示例实例库。下列问题类的实例是可用的。

对称旅行商问题(TSP)

给定一组 n 个节点和每对节点的距离, 找到一个总长度最小的往返访问每个节点仅一次。从节点 i 到节点 j 的距离与从节点 j 到节点 i 的距离相同。

哈密顿循环问题

给定一个图, 检验图是否包含哈密顿环。

非对称旅行商问题(ATSP)

给定一组 n 个节点和每对节点的距离, 找到一个总长度最小的往返访问每个节点仅一次。在这种情况下, 节点 i 到节点 j 的距离和节点 j 到节点 i 的距离可能是不同的。

顺序排序问题(SOP)

这个问题是一个带有附加约束的非对称旅行商问题。给定一组 n 个节点和每对节点的距离, 找到一条从节点1到节点 n 的最小长度且考虑给定优先约束的哈密顿路径。每个优先约束都要求必须先访问某个节点 i , 再访问另一个节点 j 。

有能力车辆路径问题(CVRP)

我们给定 n 个 $_1$ 节点, 一个车场, 以及从节点到车场的距离, 以及节点之间的距离。所有节点都有仓库可以满足的需求。对于交付给节点, 可以使用具有相同容量的卡车。问题是找到满足节点要求的最小总长度卡车的路线, 同时又不违反卡车容量的限制。卡车的数量没有指定。每次参观都会访问节点的一个子集, 并在仓库开始和结束。(注: 在一些数据文件中给出了备选仓库的集合。CVRP是通过选择其中一个仓库获得的。)

除了哈密顿循环问题, 所有的问题都定义在完全图上, 目前, 所有的距离都是整数。有一种可能要求某些边出现在问题的解中。

1. 文件格式

每个文件由一个规范部分和一个数据部分组成。规范部分包含有关文件格式及其内容的信息。数据部分包含显式数据。

1.1 规范部分

本节中的所有条目都采用<keyword>: <value>的形式, 其中<keyword>表示字母数字关键字, <value>表示字母数字或数值数据。术语<string>、<integer>和<real>分别表示字符串、整数或实数数据。数据文件中关键字的规范顺序是任意的(原则上), 但必须是一致的, 即每当指定一个关键字时, 必须知道该关键字正确解释的所有必要信息。下面我们给出了所有可用关键词的列表。

1.1.1 名称:<string>

标识数据文件。

1.1.2 类型:<string>

指定数据的类型。可能的类型有

对称旅行商问题的TSP数据ATSP数据非对称旅行商问题的

SOP数据顺序排序问题的数据

HCP哈密顿循环问题数据

CVRP有能力车辆路线问题数据TOUR一组行程

1.1.3 评论:<string>

附加注释(这里通常给出问题实例的贡献者或创建者的名字)。

1.1.4 维度:<整数>

对于TSP或ATSP, 维度是其节点的数量。对于CVRP, 它是节点和仓库的总数。对于TOUR文件, 它是相应问题的维度。

1.1.5 容量:<整数>

指定CVRP中的卡车容量。

1.1.6 边重类型:<string>

指定如何给出边权重(或距离)。权重在相应章节EUC 2D中明确列出。权重

是2-D中的欧氏距离

EUC3D权重是3-D中的欧氏距离

-

MAX 2D中明确列出。权重是2-D中的最大距离
 MAX 3D权重是3-D中的最大距离
 MAN2D中明确列出。权重是2-D的曼哈顿距离
 MAN 3D权重是3-D的曼哈顿距离
 2D中明确列出。权重是2-D欧几里得距离的四舍五入
 GEO权重是地理距离
 att48和att532问题的ATT特殊距离函数XRAY1晶体学问题的特殊距离函数(版本1)
 XRAY2晶体学问题的特殊距离函数(版本2)特殊别处记载了一个特殊的距离函数

1.1.7 边重格式:<string>

描述边权重的格式，如果明确给出的话。权重由函数给出(见上文)

完整的矩阵

权重由一个满矩阵给出
 UPPER ROW 上三角矩阵(逐行不含对角项) LOWER ROW 下三角矩阵(逐行不含对角项)
 UPPER DIAG ROW 下三角矩阵(逐行不含对角项) LOWER DIAG ROW 下三角矩阵(逐行不含对角项)
 UPPER COL 上三角矩阵(逐列不含对角项) LOWER COL 下三角矩阵(逐列不含对角项)
 UPPER DIAG COL 上三角矩阵(逐列不含对角项) LOWER DIAG COL 下三角矩阵(按列包括对角项)

1.1.7 EDGE DATA格式:<string>

描述了在图形不完整的情况下，给出图形的边的格式。值为:

EDGE LIST图由边列表给出
 列表图以邻接表的形式给出

1.1.9 节点COORD TYPE:<string>

指定坐标是否与每个节点相关联(例如，可以用于图形显示或距离计算)。值为:

TWOD坐标节点由2-D THREED坐标节点由3-D NO COORDS指定
 节点没有关联坐标

默认值为“NO COORDS”。

1.1.10 DISPLAYDATA TYPE:<string>

指定如何获得节点的图形显示。这些值是

COORD DISPLAY显示是从节点坐标生成的
 TWOD-DISPLAY给出2-D的显式坐标
 NODISPLAY不支持图形显示

如果指定了节点坐标，则默认值为COORD DISPLAY，否则为NO DISPLAY。

1.1.11 EOF:

终止输入数据。这个条目是可选的。

1.2 数据部分

根据规格的选择，可能需要一些额外的数据。这些数据在规范部分之后的相应数据部分中给出。每个数据部分都以相应的关键字开头。节的长度要么从格式规范中隐式知道，要么以适当的节结束标识符结束。

1.2.1 节点轴段: -

本节给出节点坐标。每一行都是这样的形式

<integer> <real> <real>

如果“NODE COORD TYPE”为“TWO D COORD”，则为

<integer> <real> <real> <real>

如果“NODE COORD TYPE”为“THREE D COORD”。整数给出了各自节点的编号。实数给出了相关的坐标。

1.2.2 车辆段: -

包含可能的备用车场节点列表。该列表以a₋₁结尾。

1.2.3 需求部分: -

CVRP所有节点的需求以表格形式给出(每行)

<integer> <integer>

第一个整数指定一个节点编号，第二个是它的需求。仓库节点也必须出现在这一节中。它们的需求为0。

1.2.4 EDGE DATA段:

图的边可以用EDGE DATA FORMAT条目中允许的两种格式中的任意一种指定。如果类型是EDGE LIST，那么边缘将作为该表单的一行序列给出

<integer> <integer>

每个条目给出某条边的终端节点。链表以a₋₁结尾。

如果类型为ADJ LIST，则该节由节点的邻接表列表组成。节点x的邻接表指定为

<integer> <integer> ... <integer> -1

其中第一个整数给出了节点x的数目，接下来的整数(以-1结尾)给出了与x相邻的节点的数目。邻接表的列表以一个附加的-1结尾。

1.2.5 固定边缘部分:

在这一节中,列出了每个问题解决方案中需要出现的边。需要固定的边在表格中给出(每行)

$\langle integer \rangle \langle integer \rangle$

意思是从第一个节点到第二个节点的边(弧)必须包含在一个解决方案中。这一节被 a_{-1} 终止。

1.2.6 显示数据部分:

如果显示数据TYPE为TWOD DISPLAY,则可以从生成显示的二维坐标以以下形式给出(每行)

$\langle integer \rangle \langle real \rangle \langle real \rangle$

整数指定了各自的节点,实数给出了相关的坐标。

1.2.7 旅游版块:

这部分指定了一个旅游集合。每个路线由一个整数列表给出,该整数列表给出了在该路线中访问节点的序列。每一次这样的遍历都以 a_{-1} 结束。额外的 $_{-1}$ 终止本节。

1.2.8 边重截面: -

边权值以边重格式项指定的格式给出。目前,所有显式数据都是积分的,并以一种(自解释的)矩阵格式给出。具有隐式已知的长度。

2. 距离函数

对于各种选择的EGDE WEIGHTTYPE, 我们现在描述了代表距离的计算。在每种情况下, 我们给出一个(简化的)c实现来计算与输入坐标的距离。所有涉及浮点数的计算都在双倍精度运算中进行。这些整数被假定为用32位的单词表示。由于距离要求是整数, 我们四舍五入到最近的整数(在大多数情况下)。下面我们使用了舍入函数“nint”(“nint(x)”可以用“(int)(x+0.5)”来代替)。

2.1 欧几里得距离(L_2 -metric)

对于边权类型EUC 2D和EUC 3D, 必须为每个节点指定浮点坐标。设 $x[i]$ 、 $y[i]$ 、 $z[i]$ 为节点 i 的坐标。

在二维情况下, 两点 i 和 j 之间的距离计算如下: $xd = x[i] - x[j]$;

$Yd = y[i] - y[j]$;

$Dij = \text{nint}(\text{sqrt}(xd*xd + yd*yd));$

在三维的情况下, 我们有:

$xd = x[i] - x[j]$;

$yd = y[i] - y[j]$;

$zd = z[i] - z[j]$;

$Dij = \text{nint}(\text{sqrt}(xd*xd + yd*yd + zd*zd));$ 其中, 根号(sqrt)是C的平方根函数。

2.2 曼哈顿距离(L_1 -metric)

如果边权值类型为MAN 2D或MAN 3D, 则距离以曼哈顿距离给出。它们的计算方法如下。-

二维的例子:

$xd = \text{abs}(x[i] - x[j]);$

$yd = \text{abs}(y[i] - y[j]);$

$dij = \text{nint}(xd + yd);$

三维的例子:

$Xd = \text{abs}(x[i] - x[j]); Yd = \text{abs}(y[i] -$

$y[j]); Zd = \text{abs}(z[i] - z[j]); Dij = \text{nint}$

$(xd + yd + zd);$

2.3 最大距离(L_∞ -metric)

如果边权重类型为MAX 2D或MAX 3D, 则计算最大距离。二维的例子: -

$Xd = \text{abs}(x[i] - x[j]);$

$Yd = \text{abs}(y[i] - y[j]);$

$Dij = \text{max}(\text{nint}(xd), \text{nint}(yd));$

三维的例子:

```
Xd = abs(x[i] - x[j]);  
Yd = abs(y[i] - y[j]);  
Zd = abs(z[i] - z[j]);  
Dij = max(nint(xd), nint(yd), nint(zd));
```

2.4 地理距离

如果旅行商问题是一个地理问题，那么节点对应地球上的点，两点之间的距离就是它们在半径为6378.388公里的理想球体上的距离。节点坐标给出了地球上对应点的地理纬度和经度。纬度和经度以DDD的形式给出。其中DDD是度数，MM是分钟数。纬度为正的假定为“北”，纬度为负的假定为“南”。正经度表示“东”，负经度则假定为“西”。例如，奥格斯堡的输入坐标为48.23和10.53，表示48°23′北和10°53′东。

设 $x[i]$ 和 $y[i]$ 为上述格式的城市 i 的坐标。首先，输入被转换为以弧度表示的地理纬度和经度。

```
PI = 3.141592;  
deg = nint( x[i] );  
min = x[i] - deg;  
latitude[i] = PI * (deg + 5.0 * min / 3.0 ) / 180.0;  
deg = nint( y[i] );  
min = y[i] - deg;  
longitude[i] = PI * (deg + 5.0 * min / 3.0 ) / 180.0;
```

然后以千米为单位计算两个不同节点 i 和 j 之间的距离，如下所示:

```
RRR = 6378.388;  
q1 = cos( longitude[i] - longitude[j] );  
q2 = cos( latitude[i] - latitude[j] );  
q3 = cos( latitude[i] + latitude[j] );  
dij = (int) ( RRR * acos( 0.5*((1.0+q1)*q2 - (1.0-q1)*q3) ) + 1.0);
```

函数“acos”是余弦函数的逆函数。

2.5 伪欧氏距离

边权值类型ATT对应于一个特殊的“伪欧几里德”距离函数。设 $x[i]$ 和 $y[i]$ 为节点 i 的坐标，两个点 i 和 j 之间的距离计算如下:

```
xd = x[i] - x[j];  
yd = y[i] - y[j];  
rij = sqrt( (xd*xd + yd*yd) / 10.0 );  
tij = nint( rij );  
if (tij<rij) dij = tij + 1;  
else dij = tij;
```

2.6 欧氏距离的上限

边缘权值类型CEIL 2D要求二维欧氏距离四舍五入到下一个整数。

2.7 晶体学问题的距离

我们已经将[1]中描述的晶体学问题纳入TSPLIB。这些问题没有明确给出，但提供了子程序来生成参考文献中提到的12个问题及其子问题(参见3.2节)。

要计算这些问题的距离，必须考虑到三个电机的运动。有两种类型的距离函数:一种假设电机的速度相等(XRAY1)，一种使用不同的速度(XRAY2)。相应的距离函数作为FORTRAN实现给出(files deq)。f,分别地。Duneq.f)在分发文件中。

为了获得整数距离，我们建议将原始子程序计算的距离乘以100.0并四舍五入到最接近的整数。

我们在下面的FORTRAN版本中列出了在电机速度相等的情况下修改的距离函数。

```
整数函数ICOST (v, w)
  整数V, W
  双精度dmin1, dmax1, dabs
  双精度 distp, distc, distc, COST distp = dmin1 (dabs (PHI (v) -phi (w)), dabs (dabs
  (PHI (v) -phi (w)) -360.0e +0)) distc = dabs (CHI (v) -chi (w)))
  Distt = dabs(TWOTH(v)-第二次(w)) COST = dmax1 (distp / 1.00e +
  0, distc / 1.00e +0, Distt / 1.00e +0)
C ***使积分距离*** ICOST= int (100.0E+
0*COST+0.5E+0)
  返回
  结束
```

数字PHI()、CHI()和TWOTH()分别是生成的旅行推销员问题中点的x、y和z坐标。注意，没有上述修改，TSPLIB95只包含原始的距离计算。

2.7 验证

为了验证距离函数实现的正确性，我们给出了一些“规范”旅行的长度1,2,3, ..., n。

pcb442、gr666和latt532的规范线路长度分别为221 440、423 710和309 636。

距离为XRAY1的问题xray14012([21]中考虑的第8个问题)的canonical tour长度为15 429 219。

距离XRAY2的长度为12 943 294。

3. 库文件的描述

在本节中，我们给出了目前可用的所有问题实例的列表，以及关于最佳游览长度或此长度的下限和上限的信息(如果可用)。

3.1 对称旅行商问题

TSP实例包含在TSP目录中。表1给出了问题名称以及城市数量、问题类型和已知的最佳旅行长度的下界和上界(单个数字表示已知的最佳长度)。入口矩阵表明数据是以1.1.7的一种矩阵格式给出的。对应数据文件的名称是通过附加后缀"Tsp "还提供了一些最佳的旅游路线。对应的文件名称以“。opt.tour”为后缀。

Name	#cities	Type	Bounds
a280	280	EUC_2D	2579
ali535	535	GEO	202310
att48	48	ATT	10628
att532	532	ATT	27686
bayg29	29	GEO	1610
bays29	29	GEO	2020
berlin52	52	EUC_2D	7542
bier127	127	EUC_2D	118282
brazil58	58	MATRIX	25395
brd14051	14051	EUC_2D	[468942, 469445]
brg180	180	MATRIX	1950
burma14	14	GEO	3323
ch130	130	EUC_2D	6110
ch150	150	EUC_2D	6528
d198	198	EUC_2D	15780
d493	493	EUC_2D	35002
d657	657	EUC_2D	48912
d1291	1291	EUC_2D	50801
d1655	1655	EUC_2D	62128
d2103	2103	EUC_2D	[79952, 80450]
d15112	15112	EUC_2D	[1564590, 1573152]
d18512	18512	EUC_2D	[644650, 645488]
dantzig42	42	MATRIX	699
dsj1000	1000	CEIL_2D	18659688
eil51	51	EUC_2D	426
eil76	76	EUC_2D	538
eil101	101	EUC_2D	629

表1对称旅行商问题(第一部分)

Name	#cities	Type	Bounds
f1417	417	EUC_2D	11861
f11400	1400	EUC_2D	20127
f11577	1577	EUC_2D	[22204,22249]
f13795	3795	EUC_2D	[28723,28772]
fnl4461	4461	EUC_2D	182566
fri26	26	MATRIX	937
gil262	262	EUC_2D	2378
gr17	17	MATRIX	2085
gr21	21	MATRIX	2707
gr24	24	MATRIX	1272
gr48	48	MATRIX	5046
gr96	96	GEO	55209
gr120	120	MATRIX	6942
gr137	137	GEO	69853
gr202	202	GEO	40160
gr229	229	GEO	134602
gr431	431	GEO	171414
gr666	666	GEO	294358
hk48	48	MATRIX	11461
kroA100	100	EUC_2D	21282
kroB100	100	EUC_2D	22141
kroC100	100	EUC_2D	20749
kroD100	100	EUC_2D	21294
kroE100	100	EUC_2D	22068
kroA150	150	EUC_2D	26524
kroB150	150	EUC_2D	26130
kroA200	200	EUC_2D	29368
kroB200	200	EUC_2D	29437
lin105	105	EUC_2D	14379
lin318	318	EUC_2D	42029
linhp318	318	EUC_2D	41345
nrw1379	1379	EUC_2D	56638
p654	654	EUC_2D	34643
pa561	561	MATRIX	2763
pcb442	442	EUC_2D	50778
pcb1173	1173	EUC_2D	56892
pcb3038	3038	EUC_2D	137694
pla7397	7397	CEIL_2D	23260728
pla33810	33810	CEIL_2D	[65913275,66116530]
pla85900	85900	CEIL_2D	[141904862,142487006]

表1对称旅行商问题(第二部分)

Name	#cities	Type	Bounds
pr76	76	EUC_2D	108159
pr107	107	EUC_2D	44303
pr124	124	EUC_2D	59030
pr136	136	EUC_2D	96772
pr144	144	EUC_2D	58537
pr152	152	EUC_2D	73682
pr226	226	EUC_2D	80369
pr264	264	EUC_2D	49135
pr299	299	EUC_2D	48191
pr439	439	EUC_2D	107217
pr1002	1002	EUC_2D	259045
pr2392	2392	EUC_2D	378032
rat99	99	EUC_2D	1211
rat195	195	EUC_2D	2323
rat575	575	EUC_2D	6773
rat783	783	EUC_2D	8806
rd100	100	EUC_2D	7910
rd400	400	EUC_2D	15281
rl1304	1304	EUC_2D	252948
rl1323	1323	EUC_2D	270199
rl1889	1889	EUC_2D	316536
rl5915	5915	EUC_2D	[565040,565530]
rl5934	5934	EUC_2D	[554070,556045]
rl11849	11849	EUC_2D	[920847,923368]
si175	175	MATRIX	21407
si535	535	MATRIX	48450
si1032	1032	MATRIX	92650
st70	70	EUC_2D	675
swiss42	42	MATRIX	1273
ts225	225	EUC_2D	126643
tsp225	225	EUC_2D	3919
u159	159	EUC_2D	42080
u574	574	EUC_2D	36905
u724	724	EUC_2D	41910
u1060	1060	EUC_2D	224094
u1432	1432	EUC_2D	152970
u1817	1817	EUC_2D	57201
u2152	2152	EUC_2D	64253
u2319	2319	EUC_2D	234256
ulysses16	16	GEO	6859
ulysses22	22	GEO	7013
usa13509	13509	EUC_2D	[19947008,19982889]
vm1084	1084	EUC_2D	239297
vm1748	1748	EUC_2D	336556

表1对称旅行商问题(第三部分)

晶体学问题

x射线文件。tsp目录中的问题我们分发了Bland和Shallcross编写的例程以及生成[1]中讨论的晶体学问题所需的数据。文件x射线。问题是续集中提到的单个文件合并到一个文件里。这些单文件必须从x射线中提取出来。使用编辑器的问题。提供了以下原始文件

read.me		f	.f
deq.f	duneq.	dau.f	gentsp
			f.data

此外，我们还包括了专门准备的数据文件，以生成[1]中提到的12个问题。这些文件的名称为xray1。数据通过xray12.data。

利用这些数据文件，gentsp.f程序可以生成12个对称tsp。我们建议将各自的问题实例命名为xray4472、xray2950、xray7008、xray2762、xray6922、xray9070、xray5888、xray14012、xray5520、xray13804、xray14464和xray13590。

为了验证生成例程的正确使用，我们列出了文件xray14012.tsp的一部分。名称:xray14012

评论:结晶学问题8 (Bland/Shallcross)

类型:TSP

尺寸:14012

边重类型:xray2

节点coord节 -

1	-91.802854544029	-6.4097888697337	176.39830490027
2	-87.715643397938	-6.4659384343446	165.56800324542
3	-83.587211962870	-6.4895404648110	163.53828545043
4	-79.460007412434	-6.4797580053949	165.86438271158
:			
:			
14009	100.539992581837	6.4797580053949	165.86438271158
14010	96.412788031401	6.4895404648110	163.53828545043
14011	92.284356596333	6.4659384343446	165.56800324542
14012	88.197145450242	6.4097888697337	176.39830490027

3.2 哈密顿循环问题

哈密顿循环问题的实例包含在目录hcp中。目前，我们有数据文件

alb1000。hcp alb3000b。hcp alb3000e。hcp alb2000。hcp alb3000c。hcp alb4000。hcp alb3000a。
hcp alb3000d。hcp alb5000.hcp

每个实例都包含一个对应.opt中给出的哈密顿循环。旅行文件。在问题实例alb4000中，两条边是固定的。

除了这些文件，该目录还包含M. Junger和G. Rinaldi编写的C-program tsplap .c。这个程序可以用来生成TSP实例(在TSPLIB格式)

起源于判断MXN棋盘上的(r,s)-leaper是否可以从棋盘的某个方格开始，对每个方格精确访问一次，然后返回到它的起始方格的问题。文件tsplap.c中给出了详细的文档。

3.3 不对称旅行商问题

表2列出了ATSP实例(在目录ATSP中)以及它们的最优解值。对应数据文件的名称是通过附加后缀"Atsp "ftv90、ftv100、ftv110、ftv120、ftv130、ftv140、ftv150、ftv160问题的数据文件不存在。这些实例都是从ftv170中获得的。例如ftv120是前121个节点定义的ftv170的子问题，ftv130是前131个节点定义的子问题，等等。

Name	#cities	Type	Optimum
br17	17	MATRIX	39
ft53	53	MATRIX	6905
ft70	70	MATRIX	38673
ftv33	34	MATRIX	1286
ftv35	36	MATRIX	1473
ftv38	39	MATRIX	1530
ftv44	45	MATRIX	1613
ftv47	48	MATRIX	1776
ftv55	56	MATRIX	1608
ftv64	65	MATRIX	1839
ftv70	71	MATRIX	1950
ftv90	91	MATRIX	1579
ftv100	101	MATRIX	1788
ftv110	111	MATRIX	1958
ftv120	121	MATRIX	2166
ftv130	131	MATRIX	2307
ftv140	141	MATRIX	2420
ftv150	151	MATRIX	2611
ftv160	161	MATRIX	2683
ftv170	171	MATRIX	2755
kro124	100	MATRIX	36230
p43	43	MATRIX	5620
rbg323	323	MATRIX	1326
rbg358	358	MATRIX	1163
rbg403	403	MATRIX	2465
rbg443	443	MATRIX	2720
ry48p	48	MATRIX	14422

表2不对称旅行商问题

3.4 顺序排序问题

顺序排序问题的每个实例由下列类型的完整矩阵C给出。如果节点i必须在节点j之前，则 C_{ji} 设置为 -1 。假设C相对于优先级是传递闭合的，即如果i必须在j之前，j必须在k之前，那么

这意味着 l 必须在 k 之前，因此，也必须将 C_{ki} 设置为 -1 。因为我们要求节点 1 是每个可行路径上的第一个节点，节点 n 是最后一个节点，所以SOP问题实例对于所有 $i = 2, \dots, n$ 总是有 $C_{ii} = -1$ ，对于所有 $j = 1, \dots, n$ $C_{nj} = -1$ 。条目 C_{1n} 被设置为无穷大。 C 的所有其他项都是非负整数值。

Name	#nodes	#prec.	Type	Bounds
ESC07	9	6	MATRIX	2125
ESC11	13	3	MATRIX	2075
ESC12	14	7	MATRIX	1675
ESC25	27	9	MATRIX	1681
ESC47	49	10	MATRIX	1288
ESC63	65	95	MATRIX	62
ESC78	80	77	MATRIX	18230
br17.10	17	10	MATRIX	55
br17.12	17	12	MATRIX	55
ft53.1	54	12	MATRIX	[7438, 7570]
ft53.2	54	25	MATRIX	[7630, 8335]
ft53.3	54	48	MATRIX	[9473, 10935]
ft53.4	54	63	MATRIX	14425
ft70.1	71	17	MATRIX	39313
ft70.2	71	35	MATRIX	[39739, 41778]
ft70.3	71	68	MATRIX	[41305, 44732]
ft70.4	71	86	MATRIX	[52269, 53882]
kro124p.1	101	25	MATRIX	[37722, 42845]
kro124p.2	101	49	MATRIX	[38534, 45848]
kro124p.3	101	97	MATRIX	[40967, 55649]
kro124p.4	101	131	MATRIX	[64858, 80753]
p43.1	44	9	MATRIX	27990
p43.2	44	20	MATRIX	[28175, 28330]
p43.3	44	37	MATRIX	[28366, 28680]
p43.4	44	50	MATRIX	[69569, 82960]
prob42	42	10	MATRIX	243
prob100	100	41	MATRIX	[1024, 1385]
rbg048a	50	192	MATRIX	351
rbg050c	52	256	MATRIX	467
rbg109a	111	622	MATRIX	1038
rbg150a	152	952	MATRIX	[1748, 1750]
rbg174a	176	1113	MATRIX	2053
rbg253a	255	1721	MATRIX	[2928, 2987]
rbg323a	325	2412	MATRIX	[3136, 3221]
rbg341a	343	2542	MATRIX	[2543, 2854]
rbg358a	360	3239	MATRIX	[2518, 2758]
rbg378a	380	3069	MATRIX	[2761, 3142]
ry48p.1	49	11	MATRIX	[15220, 15935]
ry48p.2	49	23	MATRIX	[15524, 17071]
ry48p.3	49	42	MATRIX	[18156, 20051]
ry48p.4	49	58	MATRIX	[29967, 31446]

表3顺序排序问题

表3列出了SOP实例(在目录SOP中)以及它们已知的最优路径长度的下界和上界。对应数据文件的名称是通过附加后缀"Sop" 后加上问题名称。

3.5 容量受限的车辆路径问题

有容量限制的车辆路径问题的数据包含在目录vrp中。数据文件以“。vrp”为后缀。目前，我们有数据文件

att48.vrp		eil7。vrp	eilB76。vrp	eil13。vrp	eilC76。vrp
eil31.vrp	eilA101.vrp	eil22。vrp	eil33。vrp	eil23。vrp	eil51。vrp
eilA76.vrp	eilD76.vrp	eilB101.vrp			
eil30.vrp					

在这些数据集上可以定义各种问题，例如，取决于车辆数量是否固定，所以我们在这里不列出最优解决方案。数据文件本身给出了一些值。

3.6 其他特殊文件

除了数据和解决方案文件，库中还包含以下特殊文件。

TSPLIB VERSION: 给出当前版本的库
 README: A short information on TSPLIB
 DOC.PS: Description of TSPLIB (PostScript)

4. 讲话

1. lin318问题原本是一个哈密顿路径问题。人们通过增加一个额外的要求来得到这个问题，即环中包含从1到214的边。数据在linhp318.tsp中给出。
2. 一些数据集在文献中以不同的名称引用。下面我们给出[3]和[2]中使用的对应名称。

TSPLIB	[3]	[2]	TSPLIB	[3]	[2]
att48	ATT048	-	att532	ATT532	-
dantzig42	-	42	eil101	EIL10	-
eil151	EIL08	-	eil76	EIL09	-
gil262	GIL249	-	gr120	-	120
gr137	GH137	137	gr202	GH202	202
gr229	GH229	229	gr431	GH431	431
gr666	GH666	666	gr96	GH096	96
hk48	-	48H	kroA100	KR0124	100A
kroB100	KR0125	100B	kroC100	KR0126	100C
kroD100	KR0127	100D	kroE100	KR0128	100E

kroA150	KR030	-	kroB150	KR031	-
kroA200	KR032	-	kroB200	KR033	-
lin105	LK105	-	lin318	LK318	-
linhp318	LK318P	-	pr1002	TK1002	-
pr107	TK107	-	pr124	TK124	-
pr136	TK136	-	pr144	TK144	-
pr152	TK152	-	pr226	TK226	-
pr2392	TK2392	-	pr264	TK264	-
pr299	TK299	-	pr439	TK439	-
pr76	TK076	-	st70	KR0070	70

3. 一些车辆路径问题在TSP版本中也是可用的。在这里，仓库只是被视为正常的节点。gil262问题原本包含两个相同的节点，其中一个被消除了。

4. 这个库的潜在贡献者应该以适当的格式提供他们的数据文件和联系方式

格哈德Reinelt

海德堡大学数学研究所

在Neuenheimer Feld 294, D-69120海德堡

德国

电话(6221)56 3171

传真(6221)56 5634

电子邮件Gerhard.Reinelt@IWR.Uni-Heidelberg.DE

5. 也欢迎有关图书馆问题的新界限或最佳解决方案的信息以及计算研究的参考资料(将包括在参考资料列表中)。

5. 访问

TSPLIB可从<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>获得

参考文献。

1. R.E. Bland & D.F. Shallcross (1989). Large Traveling Salesman Problems Arising from Experiments in X-ray Crystallography: A Preliminary Report on Computation, Operations Research Letters 8, 125–128.
2. M. Grötschel & O. Holland (1991). Solution of Large-Scale Symmetric Travelling Salesman Problems, Mathematical Programming 51, 141–202.
3. M.W. Padberg & G. Rinaldi (1991). A Branch & Cut Algorithm for the Resolution of Large-scale Symmetric Traveling Salesman Problems, SIAM Review 33, 60–100.