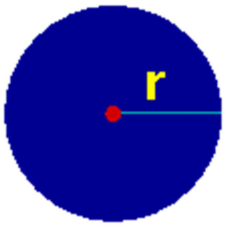
2.5.5 SOLVING TSP AS A LINEAR PROGRAMMING PROBLEM (LPP)

There are a number of different strategies for finding good lower bounds for the TSP. In Concorde and other modern TSP solvers, the bounding technique of choice goes back to G. Dantzig, R. Fulkerson, and S. Johnson and is based on the solvability of mathematical models known as **LPP**. We will introduce the Dantzig-Fulkerson-Johnson idea by using a geometric interpretation that was described by M. Juenger and W. R. Pulleyblank in the 1980s.

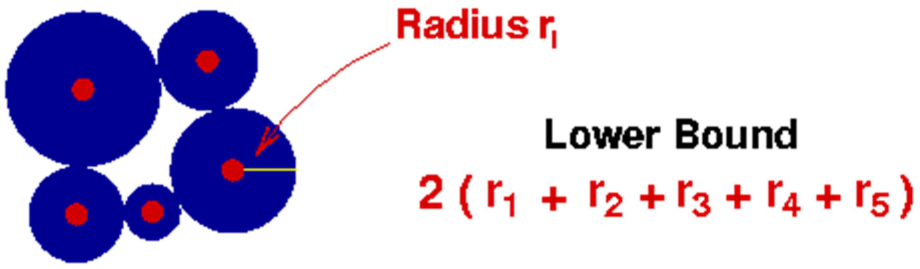
Let us suppose our TSP consists of a set of points in the plane and that the cost to travel between two cities is the (Euclidean) distance between the corresponding points. To begin, we draw a disk of radius r centered at city 1 in such a way that the disk does not touch any of the other cities. Juenger and Pulleyblank call such a disk a **control zone**.



Now the salesman must at some point in his/her tour visit city 1 and to do so they will need to travel at least distance r to arrive at the city and at least distance r to leave the city (since each other city is at least distance r from city 1). We can conclude that every tour has length at least 2r, that is, we can set B = 2r and have a lower bound for this TSP instance. When judging a LB, bigger is better.

Since we have many cities, it is convenient to have a name for the radius of each of the disks we are creating. There seems no way around introducing a bit of standard mathematical notation, so for each city i let us denote by r_i the radius of its disk.

If we have five cities, as in the following figure, then by specifying values for each of the five radii r_1, r_2, r_3, r_4, r_5 we can obtain a lower bound for the TSP.



Since we want the bound to be as large as possible, we would like to choose the five radii so as to maximize twice their sum subject to the condition that the disks do not overlap.

The non-overlapping condition can be expressed succinctly as follows. For a pair of cities i and j, let dist(i,j) denote the distance from i to j. To ensure that the disks for i and j do not overlap, we must choose the radii so that

$r_i + r_j \leq \text{dist}(i,j)$

that is, the sum of the two radii can be at most the distance between the cities. Putting everything together, the problem of getting the best TSP **upper bound** (UB) from our collection of disks can be written as follows.

Maximize $2r_1 + 2r_2 + 2r_3 + 2r_4 + 2r_5$

Subject To

$r_1 + r_2 \leq \text{dist}(1,2)$
$r_1 + r_3 \leq \text{dist}(1,3)$
$r_1 + r_4 \leq \text{dist}(1,4)$
$r_1 + r_5 \leq \text{dist}(1,5)$
$r_2 + r_3 \leq \text{dist}(2,3)$
$r_2 + r_4 \leq \text{dist}(2,4)$
$r_2 + r_5 \leq \text{dist}(2,5)$
$r_3 + r_4 \leq \text{dist}(3,4)$
$r_3 + r_5 \leq \text{dist}(3,5)$
$r_4 + r_5 \leq \text{dist}(4,5)$
$r_j \geq 0$, for $j=1,\ldots,5$

This is an example of a LPP. The important features are that we are maximizing a weighted sum of some unknown quantities (in our case the radii of the disks), subject to constraints that can be expressed as various weighted sums being at most or at least some specified values (in our case the non-overlapping constraints and the condition that no disk should not have a negative radius).

In general, for n-cities:

Maximize $Z = 2\sum_{i=1}^{n} r_i$

Subject To

$r_1 + r_2 \le \text{dist}(1,2)$
$r_1 + r_3 \le \text{dist}(1,3)$
⋮
$r_{n-1} + r_n \le \text{dist}(n-1,n)$
$r_j \ge 0$, for $j=1,\ldots,n$

Note that the no. of constraints is: $C_2^n = \frac{n(n-1)}{2}$.

Example (2.5): Solve the following TSP using DP.

	A	B	C
A	0	7	6
B	7	0	8
C	6	8	0

Solution:

Maximize $Z=2r_1 + 2r_2 + 2r_3$

Subject To

$r_1 + r_2 \le 7$
$r_1 + r_3 \le 6$

$r_2 + r_3 \leq 8$
$r_j \geq 0$, for j=1,…,3.

Adding slack variables:

$r_1 + r_2 + S_1 = 7$
$r_1 + r_3 + S_2 = 6$
$r_2 + r_3 + S_3 = 8$
$Z - 2r_1 - 2r_2 - 2r_3 = 0$

Construct the initial table:

B.V.	r_1	r_2	r_3	S_1	S_2	S_3	Sol.
S_1	1	1	0	1	0	0	7
S_2	1	0	1	0	1	0	6
S_3	0	1	1	0	0	1	8
Z	-2	-2	-2	0	0	0	0
S_1	0	1	-1	1	-1	0	1
r_1	1	0	1	0	1	0	6
S_3	0	1	1	0	0	1	8
Z	0	-2	0	0	2	0	12
r_2	0	1	-1	1	-1	0	1
r_1	1	0	1	0	1	0	6
S_3	0	0	2	-1	1	1	7
Z	0	0	-2	2	0	0	14
r_2	0	1	0	1/2	-1	1/2	9/2
r_1	1	0	0	1/2	1/2	-1/2	5/2
r_3	0	0	1	-1/2	1/2	1/2	7/2
Z	0	0	0	2	-1	0	21

Optimal solution is: Z=21, at r_1=5/2, r_2=9/2, and r_3=7/2.

Exercises (2.4):

1. A-B-C-A, Z=12.

	A	B	C
A	--	3	4
B	3	--	5
C	4	5	--

2. A-B-C-D-A, Z=16.

	A	B	C	D
A	--	3	6	2
B	3	--	5	7
C	6	5	--	6
D	2	7	6	--