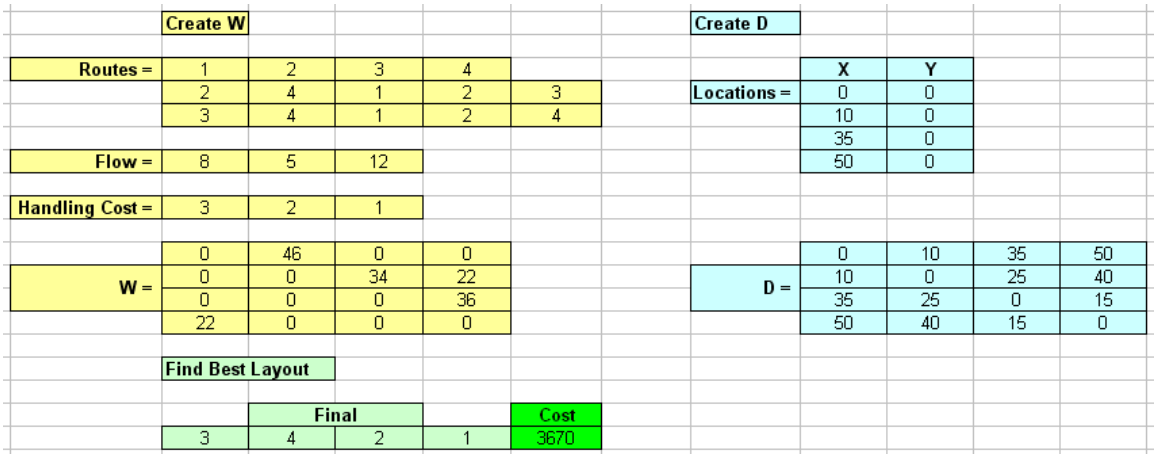
Example: Using SDPI Excel for Machine Layout Problems

ISE 453: Design of PLS Systems **Author: Shawn M. Richardson**

This Example illustrates the use of the machine layout functions in the spreadsheet *SDPI Machine Layout.xls* that you can download from the course webpage.

Solving the Example 1 using EXCEL

Using the same 4-machine SDPI example, the EXCEL function sdpi that implements the SDPI heuristic is used. Also, the EXCEL function RandPerm(*N*) is used to construct random *N*-element assignment vectors.



	Create W				
Routes =	1	2	3	4	
	2	4	1	2	3
	3	4	1	2	4
Flow =	8	5	12		
Handling Cost =	3	2	1		
W =	0	46	0	0	
	0	0	34	22	
	0	0	0	36	
	22	0	0	0	
	Find Best Layout				
		Final			Cost
	3	4	2	1	3670

	Create D			
	X	Y		
Locations =	0	0		
	10	0		
	35	0		
	50	0		
	0	10	35	50
D =	10	0	25	40
	35	25	0	15
	50	40	15	0

Creating the W (weight) Matrix:

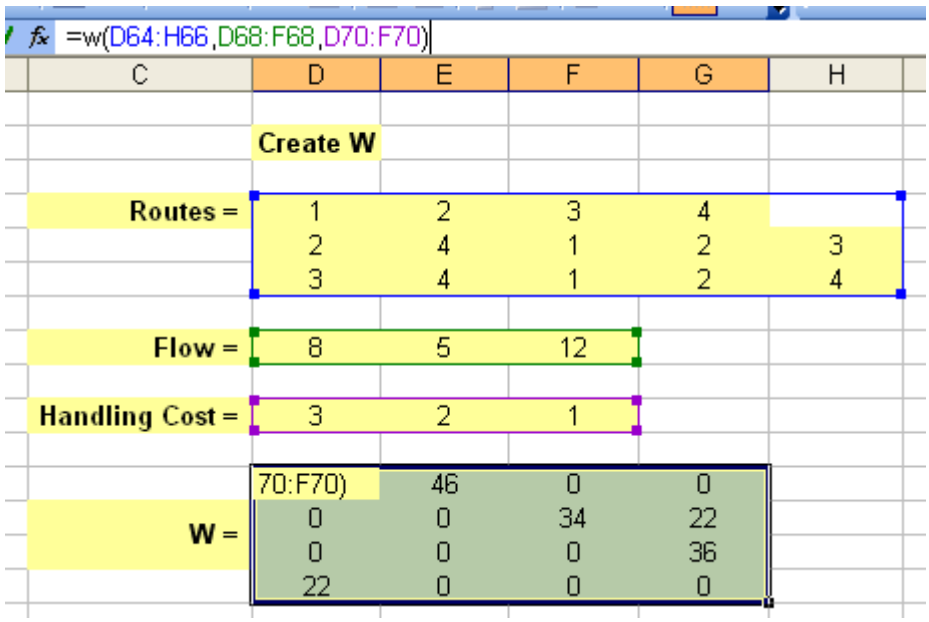
To create the W matrix, use the EXCEL function W(Routes, Flow, Handling Cost).

Recall from ISE110 that in order to execute a function across an array you:

1) Select the area that the matrix will occupy (in this case a 4x4 range)
2) Type the function and then hit Ctrl+Shift+Enter simultaneously

Alternately, you can:

1) Type the function in one cell and hit Enter
2) Select the area that the matrix will occupy with the cell that you typed the function in Step 1 being the upper left cell of the selected range and hit F2
3) Hit Ctrl+Shift+Enter simultaneously



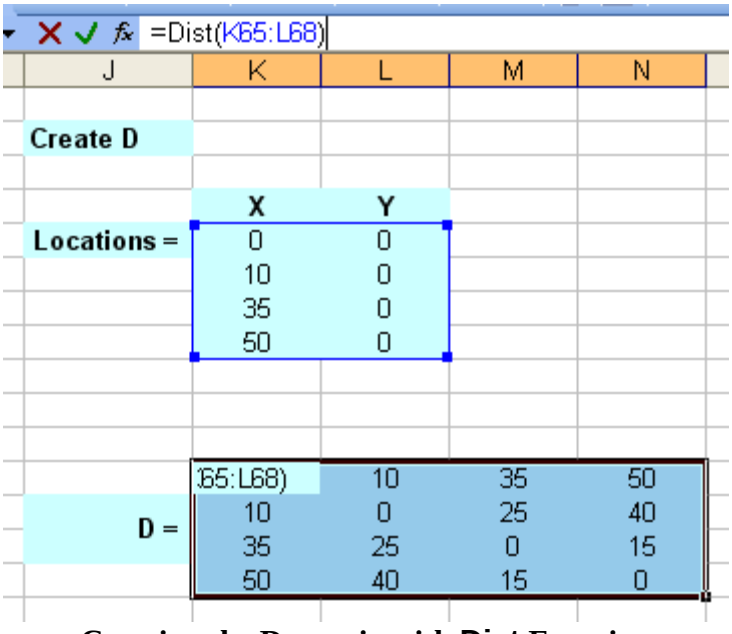
Creating the W matrix

Creating the D (distance) Matrix:

To create the D matrix, use either the EXCEL function;

Dist(Locations, [1 or 2]). The second parameter is entered as 1 or omitted if dealing with rectangular grid and entered as 2 if dealing with open space. Use this when dealing with;

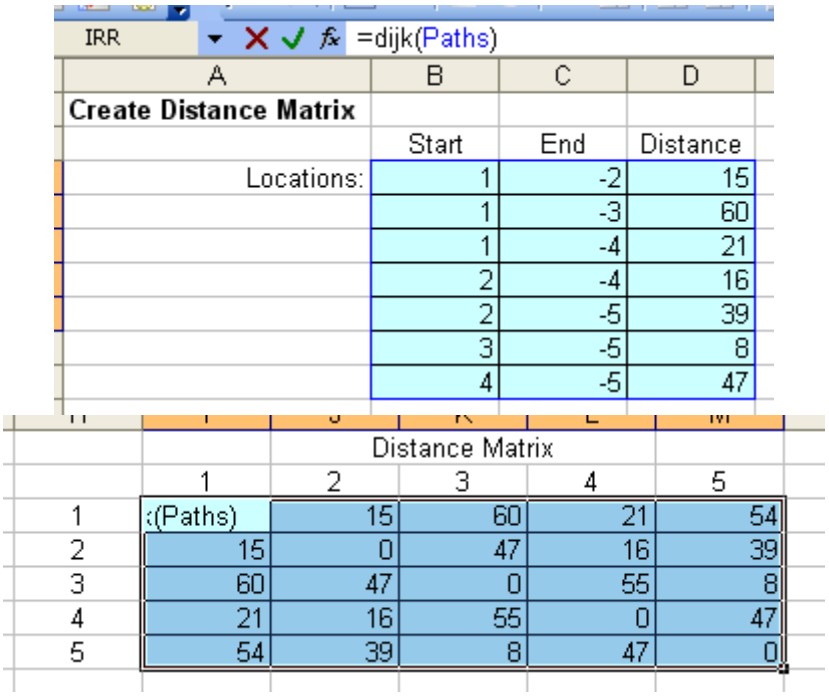
1) Rectangular Grid
2) Open Space



Creating the D matrix with Dist Function

Dijk(Routes). A single route is defined by the starting location, destination location, and the distance between the two. If the route is bidirectional, then a negative sign is associated with the destination location. Use this when dealing with;

1) Circular Conveyor
2) General Network

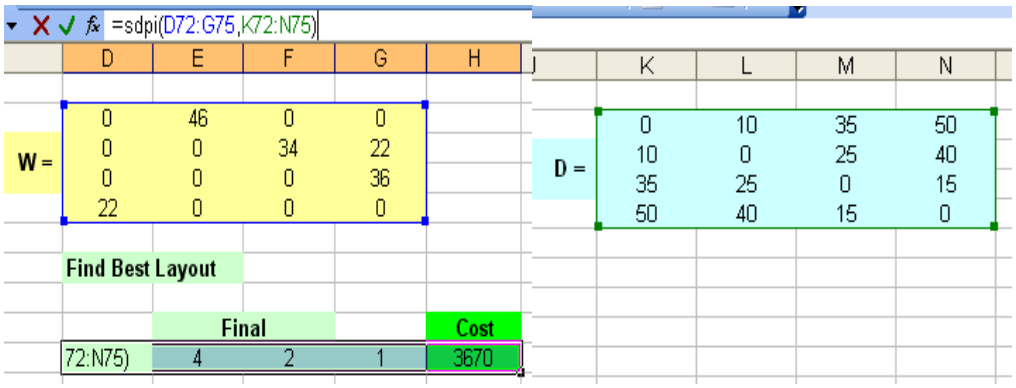


Creating the D matrix with Dijk Function (not this example)

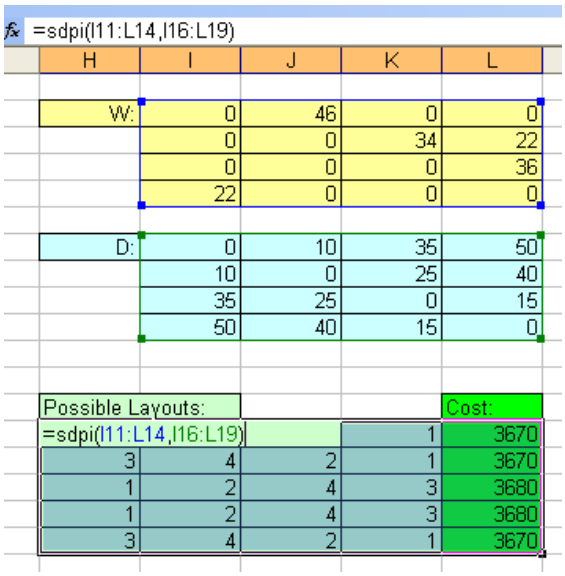
Finding the Best Layout:

To find the best layout, use the EXCEL function sdpi(w_matrix, d_Matrix, [1]). This function uses the weight matrix (W) and the distance matrix (D) calculated above.

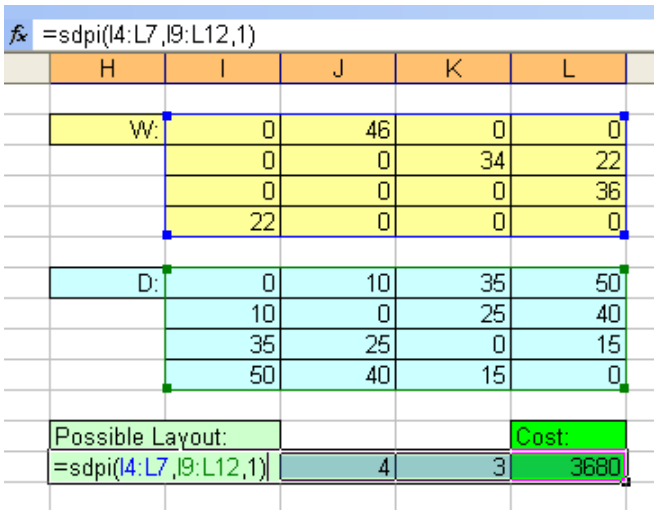
- If it is desired to find the best possible layout and cost then choose a horizontal vector (M+1) cells in length to accommodate the layout and the cost and omit the third parameter. The function will run for M^2 iterations.
- If it is desired to find multiple iterations' of amount R results, choose an array (M+1 x R) in size to accommodate the layout and the cost of each desired iteration. These layouts and costs are not the best necessarily, just the local minimums found from each of the iterations.
- If it is desired to find a single iteration's layout and cost, choose a horizontal vector (M+1) cells in length to accommodate the layout and the cost and set the third parameter of the function to 1.



Finding the best layout using sdpi function and omitting third parameter



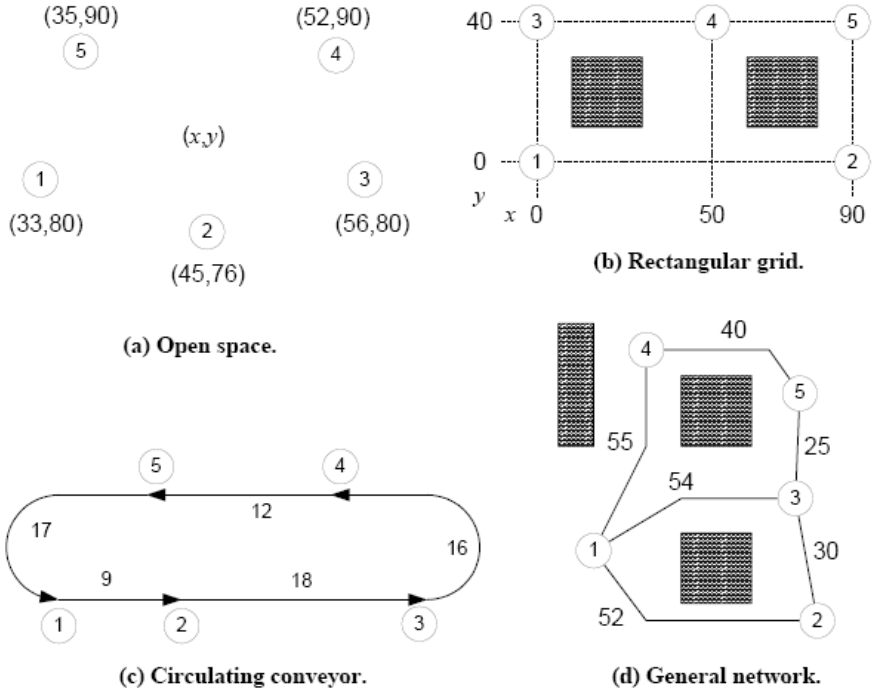
Finding multiple layouts using sdpi function by choosing multiple rows and omitting the third parameter



Finding single layout using sdpi function by setting third parameter to 1

Solving Example 2 using EXCEL

In this example, a 5 × 5 matrix **W** is created by adding a fifth row and fifth column to the 4 × 4 matrix **W** used in Example 1. The function `dist` is available to determine the (l_p norm) distances between each pair of site locations, and the function `dijk` is available to determine the shortest paths between each pair of site locations. In the function `dist`, the second parameter is used to specify the type of distance: 1, for rectilinear, and 2, for Euclidean distances. `dijk` determines the matrix **D** of shortest distances between all pairs of sites.



(a) Open space.

(b) Rectangular grid.

(c) Circulating conveyor.

(d) General network.

Figure 4.11. Four machine layout scenarios used in Example 2.

=Dist(D94:E98,2)

	D	E	F	G	H	I
	W:					
	0	46	0	0	0	
	0	0	34	22	0	
	0	0	0	36	0	
	22	0	0	0	0	
	0	0	0	0	0	
	Locations:					
	X	**Y**				
	33	80				
	45	76				
	56	80				
	52	90				
	35	90				
	D:					
	I:E98,2)	12.65	23.00	21.47	10.20	
	12.65	0.00	11.70	15.65	17.20	
	23.00	11.70	0.00	10.77	23.26	
	21.47	15.65	10.77	0.00	17.00	
	10.20	17.20	23.26	17.00	0.00	
	Best Layout:					**Cost:**
	1	2	3	4	5	2184.265

=sdpi(D86:H90,D101:H105)

	D	E	F	G	H	I
	W:					
	0	46	0	0	0	
	0	0	34	22	0	
	0	0	0	36	0	
	22	0	0	0	0	
	0	0	0	0	0	
	Locations:					
	X	**Y**				
	33	80				
	45	76				
	56	80				
	52	90				
	35	90				
	D:					
	0.00	12.65	23.00	21.47	10.20	
	12.65	0.00	11.70	15.65	17.20	
	23.00	11.70	0.00	10.77	23.26	
	21.47	15.65	10.77	0.00	17.00	
	10.20	17.20	23.26	17.00	0.00	
	Best Layout:					**Cost:**
	=sdpi(D86:	2	3	4	5	2184.265

Distance matrix for scenario (a) **Best layout for scenario (a)**

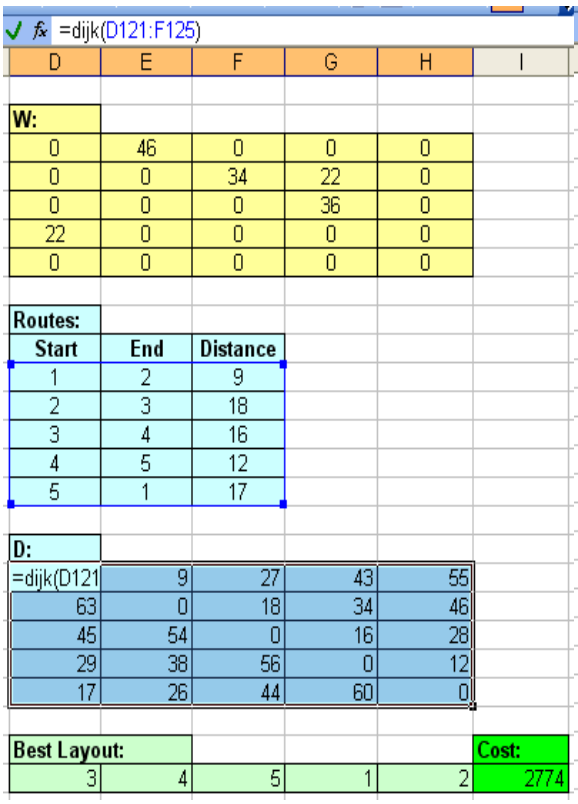
=Dist(L94:M98,1)

	L	M	N	O	P	Q
	W:					
	0	46	0	0	0	
	0	0	34	22	0	
	0	0	0	36	0	
	22	0	0	0	0	
	0	0	0	0	0	
	Locations:					
	X	**Y**				
	0	0				
	90	0				
	0	40				
	50	40				
	90	40				
	D:					
	=Dist(L94:	90	40	90	130	
	90	0	130	80	40	
	40	130	0	50	90	
	90	80	50	0	40	
	130	40	90	40	0	
	Best Layout:					**Cost:**
	5	3	1	2	4	9320

=sdpi(L86:P90,L101:P105)

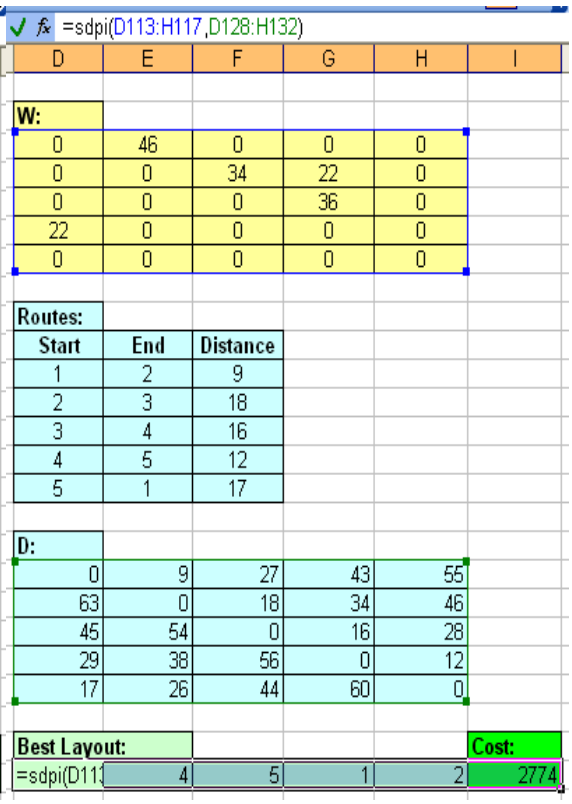
	L	M	N	O	P	Q
	W:					
	0	46	0	0	0	
	0	0	34	22	0	
	0	0	0	36	0	
	22	0	0	0	0	
	0	0	0	0	0	
	Locations:					
	X	**Y**				
	0	0				
	90	0				
	0	40				
	50	40				
	90	40				
	D:					
	0	90	40	90	130	
	90	0	130	80	40	
	40	130	0	50	90	
	90	80	50	0	40	
	130	40	90	40	0	
	Best Layout:					**Cost:**
	=sdpi(L86:	3	1	2	4	9320

Distance matrix for scenario (b) **Best layout for scenario (b)**



=dijk(D121:F125)

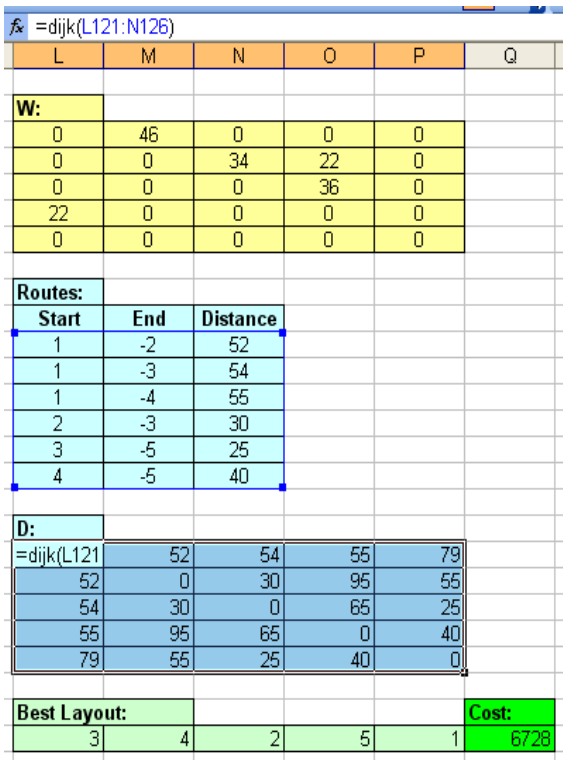
	D	E	F	G	H	I
	W:					
	0	46	0	0	0	
	0	0	34	22	0	
	0	0	0	36	0	
	22	0	0	0	0	
	0	0	0	0	0	
	Routes:					
	Start	**End**	**Distance**			
	1	2	9			
	2	3	18			
	3	4	16			
	4	5	12			
	5	1	17			
	D:					
	=dijk(D121	9	27	43	55	
	63	0	18	34	46	
	45	54	0	16	28	
	29	38	56	0	12	
	17	26	44	60	0	
	Best Layout:					**Cost:**
	3	4	5	1	2	2774



=sdpi(D113:H117,D128:H132)

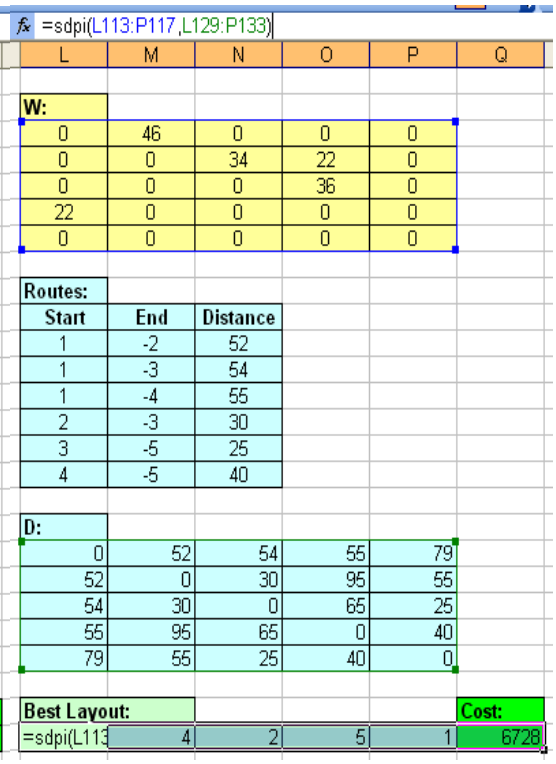
	D	E	F	G	H	I
	W:					
	0	46	0	0	0	
	0	0	34	22	0	
	0	0	0	36	0	
	22	0	0	0	0	
	0	0	0	0	0	
	Routes:					
	Start	**End**	**Distance**			
	1	2	9			
	2	3	18			
	3	4	16			
	4	5	12			
	5	1	17			
	D:					
	0	9	27	43	55	
	63	0	18	34	46	
	45	54	0	16	28	
	29	38	56	0	12	
	17	26	44	60	0	
	Best Layout:					**Cost:**
	=sdpi(D11	4	5	1	2	2774

Distance matrix for scenario (c) **Best layout for scenario (c)**



=dijk(L121:N126)

	L	M	N	O	P	Q
	W:					
	0	46	0	0	0	
	0	0	34	22	0	
	0	0	0	36	0	
	22	0	0	0	0	
	0	0	0	0	0	
	Routes:					
	Start	**End**	**Distance**			
	1	-2	52			
	1	-3	54			
	1	-4	55			
	2	-3	30			
	3	-5	25			
	4	-5	40			
	D:					
	=dijk(L121	52	54	55	79	
	52	0	30	95	55	
	54	30	0	65	25	
	55	95	65	0	40	
	79	55	25	40	0	
	Best Layout:					**Cost:**
	3	4	2	5	1	6728



=sdpi(L113:P117,L129:P133)

	L	M	N	O	P	Q
	W:					
	0	46	0	0	0	
	0	0	34	22	0	
	0	0	0	36	0	
	22	0	0	0	0	
	0	0	0	0	0	
	Routes:					
	Start	**End**	**Distance**			
	1	-2	52			
	1	-3	54			
	1	-4	55			
	2	-3	30			
	3	-5	25			
	4	-5	40			
	D:					
	0	52	54	55	79	
	52	0	30	95	55	
	54	30	0	65	25	
	55	95	65	0	40	
	79	55	25	40	0	
	Best Layout:					**Cost:**
	=sdpi(L113	4	2	5	1	6728

Distance matrix for scenario (d) **Best layout for scenario (d)**