TA: Amir Eaman
Aid sheet for Assignment 0 – Part 1

Linear programming (LP) is a procedure used to find the maximum or minimum value of a function subject to given conditions called constraints (inequalities). Therefore, LP consists of three basic components:

- Decision **variables** that we seek to determine.
- **Objectives** (goal) that we need to optimize (maximize or minimize)
- **Constraints** that the solution must satisfy

The *Transportation Algorithm* steps:

- Generally, we assume that **(**total number of supplied items**) = (**total number of demanded items**)**.

1. **Step 1) Determination of the Starting Solution.** (by using Minimum Cell Cost Method)

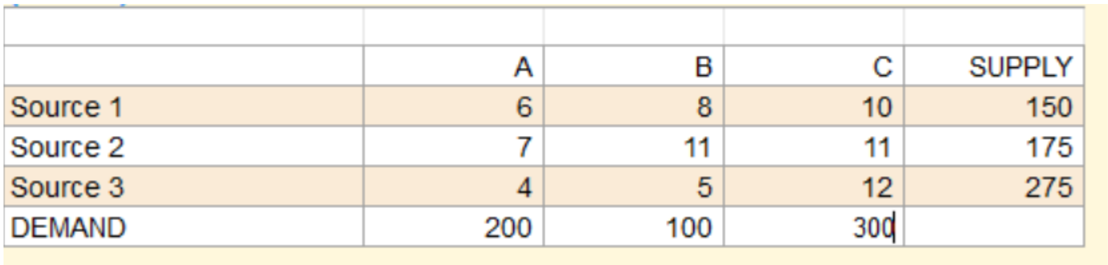
- Allocate to the cells with the lowest costs

(for *m* sources and *n* destinations) → reducing the model to (m + n -1) independent equations (non-empty cells) so at the end of this step we have (m+n-1) non-empty cells in the transportation tableau as an initial solution (feasible solution) for Stepping Stone Solution Method.

2. **Step 2) Iterative Computations of the Transportation Algorithm.** → (by using Stepping Stone method -->finding the *closed path*):

- **A)**Determine the stepping-stone paths and cost changes for each empty cell in the tableau.
- **B)**Allocate as much as possible to the empty cell with the greatest net decrease in cost.
- Repeat **A** and **B** until all empty cells have positive cost changes that indicate an optimal solutionl
- The procedure determines whether an empty cell would result in a lower total cost or not
- It starts with an empty cell and form a closed path of cells that now have allocations. In developing the path, it is possible to skip over both unused and used cells.
- The path can cross itself at one point, which is perfectly acceptable.
- Change direction in non-empty cells!
- If we find such a rout for an empty cell, then we will allocate as much as possible to it.
- A closed path or loops is a sequence of cells in the transportation table such that the first cell is empty and all the other cells are non-empty cells.
- Each pair of consecutive empty/non-empty cells lies in either the same row or column
- Units can be added to and subtracted from (+? Or -?) *only* those cells that already have allocations(except the cell that we start the procedure)
- No three consecutive empty/non-empty cells lies in the same row or column
- No cell appears more than one in a sequence
- Only horizontal and vertical moves allowed and can only change directions at non-empty cells.
- It is a method with *multiple optimal solution*

Example 1)



	A	B	C	SUPPLY
Source 1	6	8	10	150
Source 2	7	11	11	175
Source 3	4	5	12	275
DEMAND	200	100	300	

Iterations

	A	B	C
Iteration 1			
Source 1	(4)	(5)	150
Source 2	175	(3)	(-4)
Source 3	25	100	150
Iteration 2			
Source 1	(0)	(1)	150
Source 2	25	(3)	150
Source 3	175	100	(4)

Transportation Results

solution value = **$4525**	A	B	C
Source 1			150
Source 2	25		150
Source 3	175	100	

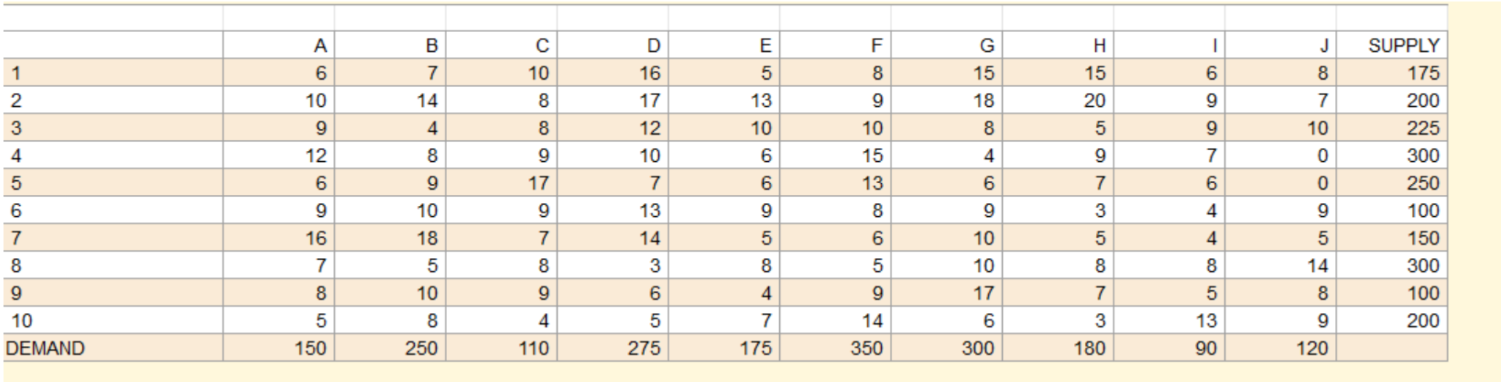
Marginal Costs

	A	B	C
Source 1	0	1	
Source 2		3	
Source 3			4

Final Solution Table

	A	B	C
Source 1	6	8	10
Source 2	7	11	11
Source 3	4	5	12

Example 2)



	A	B	C	D	E	F	G	H	I	J	SUPPLY
1	6	7	10	16	5	8	15	15	6	8	175
2	10	14	8	17	13	9	18	20	9	7	200
3	9	4	8	12	10	10	8	5	9	10	225
4	12	8	9	10	6	15	4	9	7	0	300
5	6	9	17	7	6	13	6	7	6	0	250
6	9	10	9	13	9	8	9	3	4	9	100
7	16	18	7	14	5	6	10	5	4	5	150
8	7	5	8	3	8	5	10	8	8	14	300
9	8	10	9	6	4	9	17	7	5	8	100
10	5	8	4	5	7	14	6	3	13	9	200
DEMAND	150	250	110	275	175	350	300	180	90	120	

Iterations:

	A	B	C	D	E	F	G	H	I	J
Iteration 1										
1	100	(-2)	(5)	(9)	75	(-5)	(9)	(11)	(-5)	(6)
2	(8)	(9)	(7)	(14)	(12)	200	(16)	(20)	(2)	(9)
3	(8)	225	(8)	(10)	(10)	(2)	(7)	(6)	(3)	(13)

4	(8)	(1)	(6)	(5)	(3)	(4)	180	(7)	(-2)	120
5	40	(0)	(12)	(0)	(1)	90	120	(3)	(-5)	(-2)
6	(4)	(2)	(5)	(7)	(5)	(-4)	(4)	100	(-6)	(8)
7	(17)	(16)	(9)	(14)	(7)	60	(11)	(8)	90	(10)
8	(5)	25	(7)	275	(7)	(-4)	(8)	(8)	(1)	(16)
9	(3)	(2)	(5)	(0)	100	(-3)	(12)	(4)	(-5)	(7)
10	10	(0)	110	(-1)	(3)	(2)	(1)	80	(3)	(8)
Iteration 2										
1	100	(-2)	(-1)	(9)	75	(-5)	(9)	(5)	(-5)	(6)
2	(8)	(9)	(1)	(14)	(12)	200	(16)	(14)	(2)	(9)
3	(8)	225	(2)	(10)	(10)	(2)	(7)	(0)	(3)	(13)
4	(8)	(1)	(0)	(5)	(3)	(4)	180	(1)	(-2)	120
5	50	(0)	(6)	(0)	(1)	80	120	(-3)	(-5)	(-2)
6	(10)	(8)	(5)	(13)	(11)	(2)	(10)	90	10	(14)
7	(17)	(16)	(3)	(14)	(7)	70	(11)	(2)	80	(10)
8	(5)	25	(1)	275	(7)	(-4)	(8)	(2)	(1)	(16)
9	(3)	(2)	(-1)	(0)	100	(-3)	(12)	(-2)	(-5)	(7)
10	(6)	(6)	110	(5)	(9)	(8)	(7)	90	(9)	(14)
Iteration 3										
1	20	(-2)	(4)	(9)	75	80	(9)	(10)	(0)	(6)
2	(3)	(4)	(1)	(9)	(7)	200	(11)	(14)	(2)	(4)
3	(8)	225	7)	(10)	(10)	(7)	(7)	(5)	(8)	(13)
4	(8)	(1)	(5)	(5)	(3)	(9)	180	(6)	(3)	120
5	130	(0)	(11)	(0)	(1)	(5)	120	(2)	(0)	(-2)
6	(5)	(3)	(5)	(8)	(6)	(2)	(5)	90	10	(9)

7	(12)	(11)	(3)	(9)	(2)	70	(6)	(2)	80	(5)
8	(5)	25	(6)	275	(7)	(1)	(8)	(7)	(6)	(16)
9	(3)	(2)	(4)	(0)	100	(2)	(12)	(3)	(0)	(7)
10	(1)	(1)	110	(0)	(4)	(8)	(2)	90	(9)	(9)
Iteration 4										
1	20	(0)	(4)	(11)	75	80	(9)	(10)	(0)	(6)
2	(3)	(6)	(1)	(11)	(7)	200	(11)	(14)	(2)	(4)
3	(6)	225	(5)	(10)	(8)	(5)	(5)	(3)	(6)	(11)
4	(8)	(3)	(5)	(7)	(3)	(9)	180	(6)	(3)	120
5	130	(2)	(11)	(2)	(1)	(5)	120	(2)	(0)	(-2)
6	(5)	(5)	(5)	(10)	(6)	(2)	(5)	90	10	(9)
7	(12)	(13)	(3)	(11)	(2)	70	(6)	(2)	80	(5)
8	(3)	25	(4)	275	(5)	(-1)	(6)	(5)	(4)	(14)
9	(3)	(4)	(4)	(2)	100	(2)	(12)	(3)	(0)	(7)
10	(1)	(3)	110	(2)	(4)	(8)	(2)	90	(9)	(9)
Iteration 5										
1	20	(0)	(4)	(11)	75	80	(9)	(10)	(0)	(8)

2	(3)	(6)	(1)	(11)	(7)	200	(11)	(14)	(2)	(6)
3	(6)	225	(5)	(10)	(8)	(5)	(5)	(3)	(6)	(13)
4	(8)	(3)	(5)	(7)	(3)	(9)	300	(6)	(3)	(2)
5	130	(2)	(11)	(2)	(1)	(5)	(0)	(2)	(0)	120
6	(5)	(5)	(5)	(10)	(6)	(2)	(5)	90	10	(11)
7	(12)	(13)	(3)	(11)	(2)	70	(6)	(2)	80	(7)
8	(3)	25	(4)	275	(5)	(-1)	(6)	(5)	(4)	(16)
9	(3)	(4)	(4)	(2)	100	(2)	(12)	(3)	(0)	(9)
10	(1)	(3)	110	(2)	(4)	(8)	(2)	90	(9)	(11)
Iteration 6										
1	20	25	(4)	(10)	75	55	(9)	(10)	(0)	(8)
2	(3)	(6)	(1)	(10)	(7)	200	(11)	(14)	(2)	(6)
3	(6)	225	(5)	(9)	(8)	(5)	(5)	(3)	(6)	(13)
4	(8)	(3)	(5)	(6)	(3)	(9)	300	(6)	(3)	(2)
5	130	(2)	(11)	(1)	(1)	(5)	(0)	(2)	(0)	120
6	(5)	(5)	(5)	(9)	(6)	(2)	(5)	90	10	(11)
7	(12)	(13)	(3)	(10)	(2)	70	(6)	(2)	80	(7)
8	(4)	(1)	(5)	275	(6)	25	(7)	(6)	(5)	(17)
9	(3)	(4)	(4)	(1)	100	(2)	(12)	(3)	(0)	(9)
10	(1)	(3)	110	(1)	(4)	(8)	(2)	90	(9)	(11)

Transportation Results:

solution value = **$8900**	A	B	C	D	E	F	G	H	I	J
1	20	25			75	55				
2						200				
3		225								
4							300			
5	130						0			120
6								90	10	
7						70			80	
8				275		25				
9					100					
10			110					90		