

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) \rightarrow 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. \rightarrow (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 solution!
- 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 route 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*