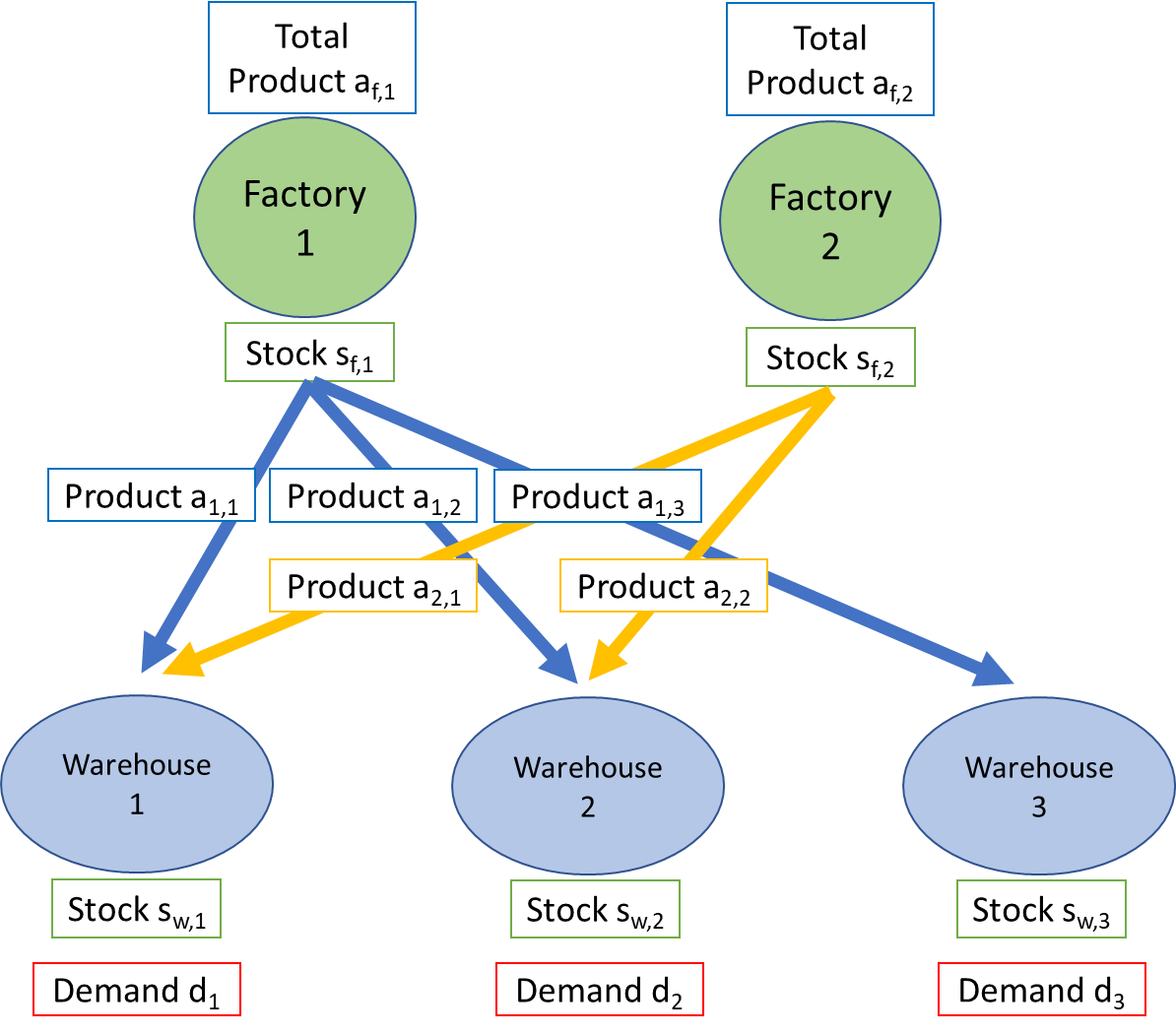
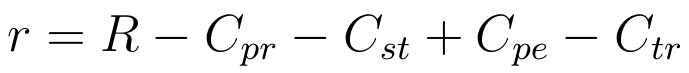
**Explanation on** *SupplyChainEnv*

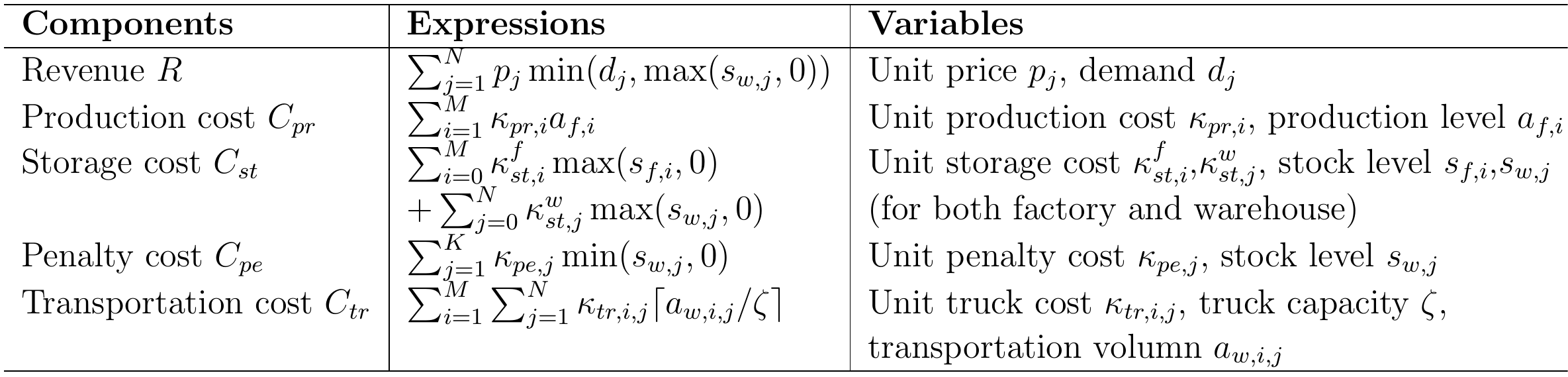
Consider a supply chain example including M factories and N warehouses (in this example 2 factories and 3 warehouses) as follows:



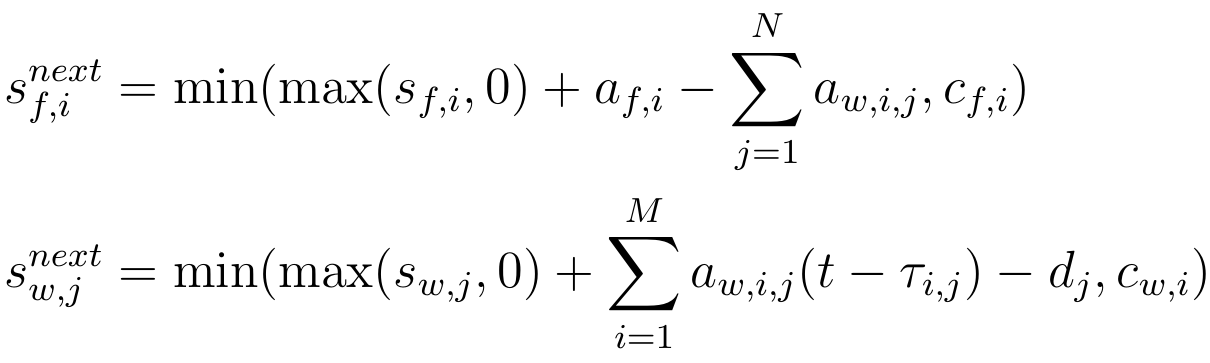
The connection matrix is [[1,1,1],[1,1,0]] in this case, where 1 or 0 denotes whether there is a connection between certain factory and warehouse.

The definition of one-step reward is given as below:



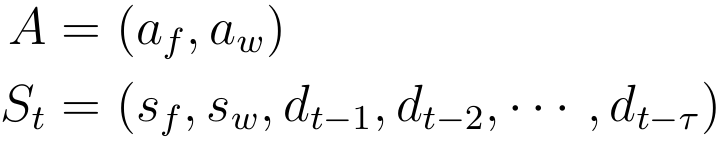


The state transition functions are provided as:



where the lead time is considered by adding a delay into in the second equation.

Also, the action ***A*** and state ***S*** with respect to reinforcement learning is defined by



where state ***S*** includes a demand history of units.

Having the network structure, system dynamics, and system parameters well setup, I will next explain the detail of this supply chain environment code in python.

* **SupplyChainEnv(con\_mat, price, k\_pr, k\_st, k\_pe, k\_tr, lead\_time, st\_max, de\_hist\_len, zeta)**

The main function well establishes the whole supply chain network structure, and the definitions of each argument are:

**con\_mat:** Connection matrix, a M by N matrix (python 2D array)

In the above example, it is given as [[1,1,1],[1,1,0]].

**price:** Unit price vector, a length-N vector

In the above example, it is given as [].

**k\_pr:** Unit production cost vector, a length-M vector

In the above example, it is given as [].

**k\_st:** Unit storage cost vector, [[length-M vector],[ length-N vector]]

In the above example, it is given as [].

**k\_pe:** Unit penalty cost vector, a length-N vector

In the above example, it is given as [].

**k\_tr:** Unit transportation cost matrix, a M by N matrix

In the above example, it is given as [[,,],[,, 0]].

**lead\_time:** Lead time matrix, a M by N matrix

In the above example, it is given as [[,,],[,, 0]].

**st\_max:** Capacity of stock, [[length-M vector],[ length-N vector]]

In the above example, it is given as [[,],[*,,* ]].

**de\_hist\_len:** Length of demand history for state vector *S*, scalar

**zeta:** Truck capacity, scalar

The main function will also build blank vector/matrix to store stock level , , production/distribution value *,* , and demand value .

* **reset(IC\_s\_f** *(optional)***, IC\_s\_w** *(optional)***)**

Reset method will clear the vector/matrix for stock level , , production/distribution value *,* , and demand value . The arguments **IC\_s\_f** and **IC\_s\_w** can be used to set the initial conditions for , (or 0 value if not provided). It will return the state vector *S* at the first time step.

* **step(action\_f, action\_w, demand)**

Step method takes the follows arguments as inputs:

**action\_f:** Current action (production value) of factory, a length-M vector

**action\_w:** Current action (distribution value) of warehouse, a M by N matrix

**demand:** Current demand, a length-N vector

It will give the output **S\_next, one\_step\_r, terminal:**

**S\_next:** State vector *S* for the next time step

**one\_step\_r:** One step reward, scalar

**terminal:** Terminal indicator, scalar (always zero here)