Homework Assignment 2: Network Transshipment Problem

Reed Ballesteros

MSDS-460

4/24/2022

Instructor: Prof. Thomas Miller

A UK brewing company owns four breweries and three packaging facilities and ships its products to fifteen demand locations (retail stores and pubs). Beer product flows from breweries to packaging facilities and then from packaging facilities to demand locations. Assume that no beer product goes directly from breweries to demand locations. Costs vary between pairs of locations.

Mathematical programming can be used for supply chain optimization. For the brewery case, we can solve for a minimum-cost plan for shipping across breweries, packaging facilities, and demand points. We aggregate data across products to show the minimum and maximum quantities of beer that can be produced at each brewery, minimum and maximum quantities that can be processed at packaging facilities, and quantities on order at demand points. We simplify the problem by summing quantities and averaging costs across beer types (ale and lager) and container/packaging types.

Complete data for the complete brewery case are provided in Kallrath (2021), which is available on Course Reserves. Note that the case presented here is a simplification of the complete brewery case, as we are considering total liquid being shipped from one location to another. We make no distinction between ale and lager. We make no distinctions across packaging types. Regardless, the aggregate liquid values used in this assignment are consistent with data for the complete brewery case.

Table 1 shows transportation costs between breweries and packaging facilities. Table 2 shows transportation costs between packaging facilities and demand locations. Tables 3a and 3b show brewery and packaging facility capacities. Table 4 shows the units on order at each demand point.

Table 1. Transportation Costs between Breweries to Packaging Facilities

Brewery	Packaging Facility	Transportation Cost per Unit
1	1	1.55
1	2	0.51
1	3	0.9
2	1	0.81
2	2	3.18
2	3	0.65
3	1	2.13
3	2	0.97
3	3	0.51
4	1	1.23
4	2	2.15
4	3	2.08

Table 2. Transportation Costs from Packaging Facilities to Demand Points

Packaging Facility	Demand Point	Average Transportation Cost per Unit
1	1	4.82
1	2	2.05

Packaging Facility	Demand Point	Average Transportation Cost per Unit
1	3	4.42
1	4	3.83
1	5	0.97
1	6	3.04
1	7	3.91
1	8	4.03
1	9	5.11
1	10	0.9
1	11	4.39
1	12	0.85
1	13	2.81
1	14	3.94
1	15	1.04
2	1	1.83
2	2	4.03
2	3	3.95
2	4	4.21
2	5	4.78
2	6	3.2
2	7	1.88
2	8	2.96
2	9	5.11
2	10	2.67
2	11	4.14
2	12	1.22
2	13	5.1
2	14	3.47
2	15	1.92
3	1	2.66
3	2	0.95
3	3	3.94
3	4	2.04
3	5	2.35
3	6	1.42
3	7	3.6
3	8	3.17

Packaging Facility	Demand Point	Average Transportation Cost per Unit
3	9	1.34
3	10	4.51
3	11	0.74
3	12	0.94
3	13	1.98
3	14	4.77
3	15	2.04

Table 3a Brewery and Packaging Facility Capacities

Brewery	Minimum Units	Maximum Units
1	100	2000
2	150	2500
3	200	3500
4	100	2000

Table 3b Brewery and Packaging Facility Capacities

Packaging Facility	Minimum Units	Maximum Units
1	50	500
2	100	1500
3	150	2500

Table 4. Demand Point Units Ordered

Demand Point	Units Ordered
1	48
2	84
3	64
4	106
5	47
6	57
7	64
8	93
9	74
10	41
11	61
12	42
13	57

Demand Point	Units Ordered
14	70
15	41

Total transportation/shipping costs depend on the number of units shipped across each of 57 paths between locations: 12 paths between breweries and packaging facilities, and 45 paths between packaging facilities and demand points. The objective of supply chain optimization is to minimize total transportation costs.

To solve this supply chain optimization problem, let's consider using a Python program that draws on the PuLP package for mathematical programming.

As part of our program, let's include a multiplier for demand, so we can see how demand affects the solution.

Output from the supply chain optimization should include optimal numbers of units of beer product to be shipped across each of the 57 paths between locations. We should also review counts for total units of production from each of the four brewery locations and each of the three packaging facilities.

To complete this assignment, review the program output and answer these five questions:

(1) Solve the supply chain optimization problem with initial settings of parameters for brewing, packaging, and demand. Describe the solution by providing the total cost (minimum cost) and quantities of beer being shipped between each pair of locations.

In [47]:
```
import regex as re # regular expresstions used in manipulating output for repo
rting solution
import pulp # mathematical programming
```

In [48]:
```
# Define hard-coded matrices and constants here
demand_multiplier = 1  # default is 1

brewery = ["B1", "B2", "B3", "B4"]

packaging_facility = ["PF1", "PF2", "PF3"]

demand_point = ["DP1", "DP2", "DP3",
                "DP4", "DP5", "DP6",
                "DP7", "DP8", "DP9",
                "DP10", "DP11", "DP12",
                "DP13", "DP14", "DP15"]

brewery_to_packaging_facility_shipping_costs = [
         [1.55, 0.51, 0.9],  # "B1" Packaging Facilities in rows
         [0.81, 3.18, 0.65], # "B2" Packaging Facilities in rows
         [2.13, 0.97, 0.51], # "B3" Packaging Facilities in rows
         [1.23, 2.15, 2.08]] # "B4" Packaging Facilities in rows

packaging_facility_to_demand_point_shipping_costs = [
        # "PF1" Demand Point in rows
        [4.82, 2.05, 4.42, 3.83, 0.97, 3.04, 3.91, 4.03, 5.11, 0.9, 4.39, 0.85
, 2.81, 3.94, 1.04],
        # "PF2" Demand Point in rows
        [1.83, 4.03, 3.95, 4.21, 4.78, 3.2, 1.88, 2.96, 5.11, 2.67, 4.14, 1.22
, 5.1, 3.47, 1.92],
        # "PF3" Demand Point in rows
        [2.66, 0.95, 3.94, 2.04, 2.35, 1.42, 3.6, 3.17, 1.34, 4.51, 0.74, 0.94
, 1.98, 4.77, 2.04]]

brewery_minimum = {"B1": 100,
           "B2": 150,
           "B3": 200,
           "B4": 100}

brewery_maximum = {"B1": 2000,
           "B2": 2500,
           "B3": 3500,
           "B4": 2000}

packaging_facility_minimum = {"PF1":50,
          "PF2":100,
          "PF3":150}

packaging_facility_maximum = {"PF1":600,
          "PF2":1500,
          "PF3":2500}

demand_point_units_ordered = {"DP1": 48, "DP2": 84, "DP3": 64,
               "DP4": 106, "DP5": 47, "DP6": 57,
               "DP7": 64, "DP8": 93, "DP9": 74,
               "DP10": 41, "DP11": 61, "DP12": 42,
               "DP13": 57, "DP14": 70, "DP15": 41}
```

In [49]:
```
def calculate_demand(demand_point_units_ordered, demand_multiplier=1):
    demand = demand_point_units_ordered.copy()
    for key in list(demand_point_units_ordered.keys()):
        demand[key] = demand_multiplier * demand_point_units_ordered[key]
    return demand
```

In [50]:

```
# Pulp calculations here
def solve_network_problem(demand):
    # Create the 'prob' variable to contain the problem data
    prob = pulp.LpProblem("Network_Transshipment_Problem", pulp.LpMinimize)
    solver = pulp.getSolver("PULP_CBC_CMD") # available on Windows computers

    first_costs = pulp.makeDict([brewery,packaging_facility],brewery_to_packag
ing_facility_shipping_costs,0)
    second_costs = pulp.makeDict([packaging_facility,demand_point],packaging_f
acility_to_demand_point_shipping_costs,0)

    # Create list of tuples containing all the possible brewery-to-packaging f
acility routes for transport
    first_routes = [(i,j) for i in brewery for j in packaging_facility]
    # A dictionary called 'Vars' is created to contain the referenced variable
s(the routes)
    first_vars = pulp.LpVariable.dicts("route",(brewery,packaging_facility),0,
None)

    # Create list of tuples containing all the possible packaging facility-to-
demand point routes for transport
    second_routes = [(i,k) for i in packaging_facility for k in demand_point]
    # A dictionary called 'Vars' is created to contain the referenced variable
s(the routes)
    second_vars = pulp.LpVariable.dicts("route",(packaging_facility,demand_poi
nt),0,None)

    # Problem Constraints:

    # Outgoing Brewery Maximiums
    for i in brewery:
        prob += pulp.lpSum([first_vars[i][j] for j in packaging_facility]) <=
brewery_maximum[i], \
                "Brewery_Capacity_Max_%s"%i

    # Outgoing Brewery Minimums
    for i in brewery:
        prob += pulp.lpSum([first_vars[i][j] for j in packaging_facility]) >=
brewery_minimum[i], \
                "Brewery_Capacity_Min_%s"%i

    # Outgoing Packaging Facility Maximiums
    for j in packaging_facility:
        prob += pulp.lpSum([first_vars[i][j] for i in brewery]) <= packaging_f
acility_maximum[j], \
                "Packaging_Facility_Capacity_Max_%s"%j

    # Outgoing Packaging Facility Minimiums
    for j in packaging_facility:
        prob += pulp.lpSum([first_vars[i][j] for i in brewery]) >= packaging_f
acility_minimum[j], \
                "Packaging_Facility_Capacity_Min_%s"%j

    # Packaging Facility incoming from Breweries needs to equal output to Dema
nd Points
```

```
    for j in packaging_facility:
        prob += pulp.lpSum(first_vars[i][j] for i in brewery) == pulp.lpSum(se
cond_vars[j][k] for k in demand_point), \
                "Packaging_Facility_Input_Output_%s"%j

    # Incoming Demand Point Requirements
    for k in demand_point:
        prob += pulp.lpSum([second_vars[j][k] for j in packaging_facility]) >=
demand[k], \
                "Demand_Point_Input_%s"%k

    # iteratively create route LpVariables from dictionaries
    # brewery-to-packaging facility routes
    for (i,j) in first_routes:
            prob += first_vars[i][j]

    # facility-to-demand point routes
    for (j,k) in second_routes:
            prob += second_vars[j][k]

    # The objective function for all transportation costs
    prob += pulp.lpSum([first_vars[i][j]*first_costs[i][j] for (i,j) in first_
routes]) + \
            pulp.lpSum([second_vars[i][k]*second_costs[i][k] for (i,k) in seco
nd_routes]), "All_Tansportation_Costs"

    prob.solve(pulp.GLPK(options=['--ranges prob_hw2.sen']))
    prob.writeLP("prob_hw2.lp")

    return prob
```

In [51]:
```python
#output
def print_output(prob, disp_prob=False):
    print("Status:\n", pulp.LpStatus[prob.status])

    if (disp_prob):
        print("\n\nProblem objective and constraints:\n\n", prob)

    print("\nRoute Distributions:")
    for v in prob.variables():
        if (v.varValue > 0):
            print(v.name, "=", round(v.varValue))

    # Brewery output totals
    B1_output = 0
    B2_output = 0
    B3_output = 0
    B4_output = 0

    # Packaging Facility output totals
    PF1_received = 0
    PF2_received = 0
    PF3_received = 0

    # Packaging Facility output totals
    PF1_output = 0
    PF2_output = 0
    PF3_output = 0

    # Demand Point received totals
    DP1_received = 0
    DP2_received = 0
    DP3_received = 0
    DP4_received = 0
    DP5_received = 0
    DP6_received = 0
    DP7_received = 0
    DP8_received = 0
    DP9_received = 0
    DP10_received = 0
    DP11_received = 0
    DP12_received = 0
    DP13_received = 0
    DP14_received = 0
    DP15_received = 0

    for v in prob.variables():
        if re.search("route_B1_..",v.name):
            B1_output += round(v.varValue)
        if re.search("route_B2_..",v.name):
            B2_output += round(v.varValue)
        if re.search("route_B3_..",v.name):
            B3_output += round(v.varValue)
        if re.search("route_B4_..",v.name):
            B4_output += round(v.varValue)
```

```
        # need dollar sign as the end of regex for calculating PFX_received
        # or will get the other routes containing '_PFX'
        # like 'route_PFX_' which designate packaging facility output
        if re.search("_PF1$",v.name):
            PF1_received += round(v.varValue)
        if re.search("_PF2$",v.name):
            PF2_received += round(v.varValue)
        if re.search("_PF3$",v.name):
            PF3_received += round(v.varValue)

        if re.search("route_PF1_..",v.name):
            PF1_output += round(v.varValue)
        if re.search("route_PF2_..",v.name):
            PF2_output += round(v.varValue)
        if re.search("route_PF3_..",v.name):
            PF3_output += round(v.varValue)

        # need dollar sign as the end of regex for calculating DP1_received
        # or will get the other demamnd points containing '_DP1'
        # like '_DP11' '_DP12' '_DP13' '_DP14' '_DP15'
        if re.search("_DP1$",v.name):
            DP1_received += round(v.varValue)

        if re.search("_DP2",v.name):
            DP2_received += round(v.varValue)
        if re.search("_DP3",v.name):
            DP3_received += round(v.varValue)
        if re.search("_DP4",v.name):
            DP4_received += round(v.varValue)
        if re.search("_DP5",v.name):
            DP5_received += round(v.varValue)
        if re.search("_DP6",v.name):
            DP6_received += round(v.varValue)
        if re.search("_DP7",v.name):
            DP7_received += round(v.varValue)
        if re.search("_DP8",v.name):
            DP8_received += round(v.varValue)
        if re.search("_DP9",v.name):
            DP9_received += round(v.varValue)
        if re.search("_DP10",v.name):
            DP10_received += round(v.varValue)
        if re.search("_DP11",v.name):
            DP11_received += round(v.varValue)
        if re.search("_DP12",v.name):
            DP12_received += round(v.varValue)
        if re.search("_DP13",v.name):
            DP13_received += round(v.varValue)
        if re.search("_DP14",v.name):
            DP14_received += round(v.varValue)
        if re.search("_DP15",v.name):
            DP15_received += round(v.varValue)

    total_brewery_output = B1_output + B2_output + B3_output + B4_output
    total_packaging_facility_received = PF1_received + PF2_received + PF3_rece
ived
    total_packaging_facility_output = PF1_output + PF2_output + PF3_output
```

```
    total_demand_point_received = DP1_received + DP2_received + DP3_received + 
DP4_received + DP5_received + \
                            DP6_received + DP7_received + DP8_received + DP9_r
eceived + DP10_received + \
                            DP11_received + DP12_received + DP13_received + DP
14_received + DP15_received

    # demand values for output, based on demand_multiplier
    DP1_demand = demand['DP1']
    DP2_demand = demand['DP2']
    DP3_demand = demand['DP3']
    DP4_demand = demand['DP4']
    DP5_demand = demand['DP5']
    DP6_demand = demand['DP6']
    DP7_demand = demand['DP7']
    DP8_demand = demand['DP8']
    DP9_demand = demand['DP9']
    DP10_demand = demand['DP10']
    DP11_demand = demand['DP11']
    DP12_demand = demand['DP12']
    DP13_demand = demand['DP13']
    DP14_demand = demand['DP14']
    DP15_demand = demand['DP15']

    total_demand = sum(demand.values())

    print()
    print("total_brewery_output:",total_brewery_output)
    print("total_packaging_facility_received:",total_packaging_facility_receiv
ed)
    print("total_packaging_facility_output:",total_packaging_facility_output)
    print("total_demand_point_received:", total_demand_point_received)

    print()
    print("B1_output:",B1_output)
    print("B2_output:",B2_output)
    print("B3_output:",B3_output)
    print("B4_output:",B4_output)

    print()
    print("PF1_received:",PF1_received)
    print("PF2_received:",PF2_received)
    print("PF3_received:",PF3_received)

    print()
    print("PF1_output:",PF1_output)
    print("PF2_output:",PF2_output)
    print("PF3_output:",PF3_output)

    print()
    print("total demand requested:", total_demand)
    if (total_demand_point_received < total_demand):
        print("demand point requirements not met")
    if (total_demand_point_received >= total_demand):
        print("demand point requirements met")
    print("DP1_demand:", DP1_demand, "DP1_received:",DP1_received)
```

```
print("DP2_demand:", DP2_demand, "DP2_received:",DP2_received)
print("DP3_demand:", DP3_demand, "DP3_received:",DP3_received)
print("DP4_demand:", DP4_demand, "DP4_received:",DP4_received)
print("DP5_demand:", DP5_demand, "DP5_received:",DP5_received)
print("DP6_demand:", DP6_demand, "DP6_received:",DP6_received)
print("DP7_demand:", DP7_demand, "DP7_received:",DP7_received)
print("DP8_demand:", DP8_demand, "DP8_received:",DP8_received)
print("DP9_demand:", DP9_demand, "DP9_received:",DP9_received)
print("DP10_demand:", DP10_demand, "DP10_received:",DP10_received)
print("DP11_demand:", DP11_demand, "DP11_received:",DP11_received)
print("DP12_demand:", DP12_demand, "DP12_received:",DP12_received)
print("DP13_demand:", DP13_demand, "DP13_received:",DP13_received)
print("DP14_demand:", DP14_demand, "DP14_received:",DP14_received)
print("DP15_demand:", DP15_demand, "DP15_received:",DP15_received)


print("\nTotal demand points received:", total_demand_point_received)

print()
print("Total shipping costs: $%d" %(pulp.value(prob.objective)))
```

In [52]:

```
demand = calculate_demand(demand_point_units_ordered)
prob = solve_network_problem(demand)
print_output(prob, True)
```

```
Status:
 Optimal


Problem objective and constraints:

 Network_Transshipment_Problem:
MINIMIZE
1.55*route_B1_PF1 + 0.51*route_B1_PF2 + 0.9*route_B1_PF3 + 0.81*route_B2_PF1
+ 3.18*route_B2_PF2 + 0.65*route_B2_PF3 + 2.13*route_B3_PF1 + 0.97*route_B3_P
F2 + 0.51*route_B3_PF3 + 1.23*route_B4_PF1 + 2.15*route_B4_PF2 + 2.08*route_B
4_PF3 + 4.82*route_PF1_DP1 + 0.9*route_PF1_DP10 + 4.39*route_PF1_DP11 + 0.85*
route_PF1_DP12 + 2.81*route_PF1_DP13 + 3.94*route_PF1_DP14 + 1.04*route_PF1_D
P15 + 2.05*route_PF1_DP2 + 4.42*route_PF1_DP3 + 3.83*route_PF1_DP4 + 0.97*rou
te_PF1_DP5 + 3.04*route_PF1_DP6 + 3.91*route_PF1_DP7 + 4.03*route_PF1_DP8 +
5.11*route_PF1_DP9 + 1.83*route_PF2_DP1 + 2.67*route_PF2_DP10 + 4.14*route_PF
2_DP11 + 1.22*route_PF2_DP12 + 5.1*route_PF2_DP13 + 3.47*route_PF2_DP14 + 1.9
2*route_PF2_DP15 + 4.03*route_PF2_DP2 + 3.95*route_PF2_DP3 + 4.21*route_PF2_D
P4 + 4.78*route_PF2_DP5 + 3.2*route_PF2_DP6 + 1.88*route_PF2_DP7 + 2.96*route
_PF2_DP8 + 5.11*route_PF2_DP9 + 2.66*route_PF3_DP1 + 4.51*route_PF3_DP10 + 0.
74*route_PF3_DP11 + 0.94*route_PF3_DP12 + 1.98*route_PF3_DP13 + 4.77*route_PF
3_DP14 + 2.04*route_PF3_DP15 + 0.95*route_PF3_DP2 + 3.94*route_PF3_DP3 + 2.04
*route_PF3_DP4 + 2.35*route_PF3_DP5 + 1.42*route_PF3_DP6 + 3.6*route_PF3_DP7
+ 3.17*route_PF3_DP8 + 1.34*route_PF3_DP9 + 0.0
SUBJECT TO
Brewery_Capacity_Max_B1: route_B1_PF1 + route_B1_PF2 + route_B1_PF3 <= 2000

Brewery_Capacity_Max_B2: route_B2_PF1 + route_B2_PF2 + route_B2_PF3 <= 2500

Brewery_Capacity_Max_B3: route_B3_PF1 + route_B3_PF2 + route_B3_PF3 <= 3500

Brewery_Capacity_Max_B4: route_B4_PF1 + route_B4_PF2 + route_B4_PF3 <= 2000

Brewery_Capacity_Min_B1: route_B1_PF1 + route_B1_PF2 + route_B1_PF3 >= 100

Brewery_Capacity_Min_B2: route_B2_PF1 + route_B2_PF2 + route_B2_PF3 >= 150

Brewery_Capacity_Min_B3: route_B3_PF1 + route_B3_PF2 + route_B3_PF3 >= 200

Brewery_Capacity_Min_B4: route_B4_PF1 + route_B4_PF2 + route_B4_PF3 >= 100

Packaging_Facility_Capacity_Max_PF1: route_B1_PF1 + route_B2_PF1
 + route_B3_PF1 + route_B4_PF1 <= 600

Packaging_Facility_Capacity_Max_PF2: route_B1_PF2 + route_B2_PF2
 + route_B3_PF2 + route_B4_PF2 <= 1500

Packaging_Facility_Capacity_Max_PF3: route_B1_PF3 + route_B2_PF3
 + route_B3_PF3 + route_B4_PF3 <= 2500

Packaging_Facility_Capacity_Min_PF1: route_B1_PF1 + route_B2_PF1
 + route_B3_PF1 + route_B4_PF1 >= 50

Packaging_Facility_Capacity_Min_PF2: route_B1_PF2 + route_B2_PF2
 + route_B3_PF2 + route_B4_PF2 >= 100
```

```
Packaging_Facility_Capacity_Min_PF3: route_B1_PF3 + route_B2_PF3
 + route_B3_PF3 + route_B4_PF3 >= 150

Packaging_Facility_Input_Output_PF1: route_B1_PF1 + route_B2_PF1
 + route_B3_PF1 + route_B4_PF1 - route_PF1_DP1 - route_PF1_DP10
 - route_PF1_DP11 - route_PF1_DP12 - route_PF1_DP13 - route_PF1_DP14
 - route_PF1_DP15 - route_PF1_DP2 - route_PF1_DP3 - route_PF1_DP4
 - route_PF1_DP5 - route_PF1_DP6 - route_PF1_DP7 - route_PF1_DP8
 - route_PF1_DP9 = 0

Packaging_Facility_Input_Output_PF2: route_B1_PF2 + route_B2_PF2
 + route_B3_PF2 + route_B4_PF2 - route_PF2_DP1 - route_PF2_DP10
 - route_PF2_DP11 - route_PF2_DP12 - route_PF2_DP13 - route_PF2_DP14
 - route_PF2_DP15 - route_PF2_DP2 - route_PF2_DP3 - route_PF2_DP4
 - route_PF2_DP5 - route_PF2_DP6 - route_PF2_DP7 - route_PF2_DP8
 - route_PF2_DP9 = 0

Packaging_Facility_Input_Output_PF3: route_B1_PF3 + route_B2_PF3
 + route_B3_PF3 + route_B4_PF3 - route_PF3_DP1 - route_PF3_DP10
 - route_PF3_DP11 - route_PF3_DP12 - route_PF3_DP13 - route_PF3_DP14
 - route_PF3_DP15 - route_PF3_DP2 - route_PF3_DP3 - route_PF3_DP4
 - route_PF3_DP5 - route_PF3_DP6 - route_PF3_DP7 - route_PF3_DP8
 - route_PF3_DP9 = 0

Demand_Point_Input_DP1: route_PF1_DP1 + route_PF2_DP1 + route_PF3_DP1 >= 48

Demand_Point_Input_DP2: route_PF1_DP2 + route_PF2_DP2 + route_PF3_DP2 >= 84

Demand_Point_Input_DP3: route_PF1_DP3 + route_PF2_DP3 + route_PF3_DP3 >= 64

Demand_Point_Input_DP4: route_PF1_DP4 + route_PF2_DP4 + route_PF3_DP4 >= 106

Demand_Point_Input_DP5: route_PF1_DP5 + route_PF2_DP5 + route_PF3_DP5 >= 47

Demand_Point_Input_DP6: route_PF1_DP6 + route_PF2_DP6 + route_PF3_DP6 >= 57

Demand_Point_Input_DP7: route_PF1_DP7 + route_PF2_DP7 + route_PF3_DP7 >= 64

Demand_Point_Input_DP8: route_PF1_DP8 + route_PF2_DP8 + route_PF3_DP8 >= 93

Demand_Point_Input_DP9: route_PF1_DP9 + route_PF2_DP9 + route_PF3_DP9 >= 74

Demand_Point_Input_DP10: route_PF1_DP10 + route_PF2_DP10 + route_PF3_DP10
 >= 41

Demand_Point_Input_DP11: route_PF1_DP11 + route_PF2_DP11 + route_PF3_DP11
 >= 61

Demand_Point_Input_DP12: route_PF1_DP12 + route_PF2_DP12 + route_PF3_DP12
 >= 42

Demand_Point_Input_DP13: route_PF1_DP13 + route_PF2_DP13 + route_PF3_DP13
 >= 57

Demand_Point_Input_DP14: route_PF1_DP14 + route_PF2_DP14 + route_PF3_DP14
```

```
 >= 70

Demand_Point_Input_DP15: route_PF1_DP15 + route_PF2_DP15 + route_PF3_DP15
 >= 41

VARIABLES
route_B1_PF1 Continuous
route_B1_PF2 Continuous
route_B1_PF3 Continuous
route_B2_PF1 Continuous
route_B2_PF2 Continuous
route_B2_PF3 Continuous
route_B3_PF1 Continuous
route_B3_PF2 Continuous
route_B3_PF3 Continuous
route_B4_PF1 Continuous
route_B4_PF2 Continuous
route_B4_PF3 Continuous
route_PF1_DP1 Continuous
route_PF1_DP10 Continuous
route_PF1_DP11 Continuous
route_PF1_DP12 Continuous
route_PF1_DP13 Continuous
route_PF1_DP14 Continuous
route_PF1_DP15 Continuous
route_PF1_DP2 Continuous
route_PF1_DP3 Continuous
route_PF1_DP4 Continuous
route_PF1_DP5 Continuous
route_PF1_DP6 Continuous
route_PF1_DP7 Continuous
route_PF1_DP8 Continuous
route_PF1_DP9 Continuous
route_PF2_DP1 Continuous
route_PF2_DP10 Continuous
route_PF2_DP11 Continuous
route_PF2_DP12 Continuous
route_PF2_DP13 Continuous
route_PF2_DP14 Continuous
route_PF2_DP15 Continuous
route_PF2_DP2 Continuous
route_PF2_DP3 Continuous
route_PF2_DP4 Continuous
route_PF2_DP5 Continuous
route_PF2_DP6 Continuous
route_PF2_DP7 Continuous
route_PF2_DP8 Continuous
route_PF2_DP9 Continuous
route_PF3_DP1 Continuous
route_PF3_DP10 Continuous
route_PF3_DP11 Continuous
route_PF3_DP12 Continuous
route_PF3_DP13 Continuous
route_PF3_DP14 Continuous
route_PF3_DP15 Continuous
route_PF3_DP2 Continuous
```

```
route_PF3_DP3 Continuous
route_PF3_DP4 Continuous
route_PF3_DP5 Continuous
route_PF3_DP6 Continuous
route_PF3_DP7 Continuous
route_PF3_DP8 Continuous
route_PF3_DP9 Continuous


Route Distributions:
route_B1_PF2 = 275
route_B2_PF1 = 29
route_B2_PF3 = 121
route_B3_PF3 = 424
route_B4_PF1 = 100
route_PF1_DP10 = 41
route_PF1_DP15 = 41
route_PF1_DP5 = 47
route_PF2_DP1 = 48
route_PF2_DP14 = 70
route_PF2_DP7 = 64
route_PF2_DP8 = 93
route_PF3_DP11 = 61
route_PF3_DP12 = 42
route_PF3_DP13 = 57
route_PF3_DP2 = 84
route_PF3_DP3 = 64
route_PF3_DP4 = 106
route_PF3_DP6 = 57
route_PF3_DP9 = 74

total_brewery_output: 949
total_packaging_facility_received: 949
total_packaging_facility_output: 949
total_demand_point_received: 949

B1_output: 275
B2_output: 150
B3_output: 424
B4_output: 100

PF1_received: 129
PF2_received: 275
PF3_received: 545

PF1_output: 129
PF2_output: 275
PF3_output: 545

total demand requested: 949
demand point requirements met
DP1_demand: 48 DP1_received: 48
DP2_demand: 84 DP2_received: 84
DP3_demand: 64 DP3_received: 64
DP4_demand: 106 DP4_received: 106
DP5_demand: 47 DP5_received: 47
```

```
DP6_demand: 57 DP6_received: 57
DP7_demand: 64 DP7_received: 64
DP8_demand: 93 DP8_received: 93
DP9_demand: 74 DP9_received: 74
DP10_demand: 41 DP10_received: 41
DP11_demand: 61 DP11_received: 61
DP12_demand: 42 DP12_received: 42
DP13_demand: 57 DP13_received: 57
DP14_demand: 70 DP14_received: 70
DP15_demand: 41 DP15_received: 41

Total demand points received: 949

Total shipping costs: $2358
```

(2) Due to low demand for its products, the brewing company is thinking about closing any brewing location that is operating at minimum capacity. Reviewing the solution to the optimization problem, which brewery would you close (if any)?

Brewery 4 is operating at minimum capacity, due to the fact that it has a higher average cost to distribute to the packaging facilities comapred to the other breweries. This seems to be the brewery that could be closed.

(3) Due to low demand, the company may also want to close one of its packaging facilities. Which packaging facility would you close (if any)?

Let's lower the demand_multiplier to some value (0.58) that meets the total bare minimum of the brewery capacities (just over 550 beer units total).

In [75]:
```
# create a 'low-demand' mapping, 550 beer units total
demand_point_units_ordered = {"DP1": 37, "DP2": 37, "DP3": 37,
               "DP4": 37, "DP5": 37, "DP6": 37,
               "DP7": 37, "DP8": 37, "DP9": 37,
               "DP10": 37, "DP11": 36, "DP12": 36,
               "DP13": 36, "DP14": 36, "DP15": 36}
demand = calculate_demand(demand_point_units_ordered)
prob = solve_network_problem(demand)
print_output(prob)
```

```
Status:
 Optimal

Route Distributions:
route_B1_PF2 = 100
route_B2_PF1 = 10
route_B2_PF3 = 140
route_B3_PF2 = 10
route_B3_PF3 = 190
route_B4_PF1 = 100
route_PF1_DP10 = 37
route_PF1_DP15 = 36
route_PF1_DP5 = 37
route_PF2_DP1 = 37
route_PF2_DP14 = 36
route_PF2_DP7 = 37
route_PF3_DP11 = 36
route_PF3_DP12 = 36
route_PF3_DP13 = 36
route_PF3_DP2 = 37
route_PF3_DP3 = 37
route_PF3_DP4 = 37
route_PF3_DP6 = 37
route_PF3_DP8 = 37
route_PF3_DP9 = 37

total_brewery_output: 550
total_packaging_facility_received: 550
total_packaging_facility_output: 550
total_demand_point_received: 550

B1_output: 100
B2_output: 150
B3_output: 200
B4_output: 100

PF1_received: 110
PF2_received: 110
PF3_received: 330

PF1_output: 110
PF2_output: 110
PF3_output: 330

total demand requested: 550
demand point requirements met
DP1_demand: 37 DP1_received: 37
DP2_demand: 37 DP2_received: 37
DP3_demand: 37 DP3_received: 37
DP4_demand: 37 DP4_received: 37
DP5_demand: 37 DP5_received: 37
DP6_demand: 37 DP6_received: 37
DP7_demand: 37 DP7_received: 37
DP8_demand: 37 DP8_received: 37
DP9_demand: 37 DP9_received: 37
```

```
DP10_demand: 37 DP10_received: 37
DP11_demand: 36 DP11_received: 36
DP12_demand: 36 DP12_received: 36
DP13_demand: 36 DP13_received: 36
DP14_demand: 36 DP14_received: 36
DP15_demand: 36 DP15_received: 36

Total demand points received: 550

Total shipping costs: $1356
```

In this case, if the total number of beer units demanded is 550, we notice that packaging facility 2 operates closer to its minimum capacity (110 with a 100 minimum, 10 over) compared to packaging facility 1 (110 with a 50 minimum, 60 over). Packaging facility 1 is leveraged more over its minimum in this scenario due to its lower average import and export costs compared to facility 2. If this lower demand persists, closing facility 2 would lead to leveraging lower costs a little more with facility 3 compared to facility 1 since facility 3 has the lowest average shipping costs and the higher max capacity, in or out.

That being said, since we are not sure of the operating costs for each packaging facility and how it can offset or add to the transportation costs of each facility, we cannot make a full assessment for ultimately deciding which packaging facility to close.

Simulated, though, closing Packaging Facility 1 does raise overall costs ($1606vs$1356) in a low demand scenario:

Route Distributions:

- route_B1_PF3 = 100
- route_B2_PF2 = 150
- route_B3_PF3 = 200
- route_B4_PF2 = 100
- route_PF2_DP1 = 37
- route_PF2_DP10 = 37
- route_PF2_DP14 = 36
- route_PF2_DP15 = 36
- route_PF2_DP3 = 30
- route_PF2_DP7 = 37
- route_PF2_DP8 = 37
- route_PF3_DP11 = 36
- route_PF3_DP12 = 36
- route_PF3_DP13 = 36
- route_PF3_DP2 = 37
- route_PF3_DP3 = 7
- route_PF3_DP4 = 37
- route_PF3_DP5 = 37
- route_PF3_DP6 = 37
- route_PF3_DP9 = 37
- total_brewery_output: 550
- total_packaging_facility_received: 550
- total_packaging_facility_output: 550
- total_demand_point_received: 550
- B1_output: 100
- B2_output: 150
- B3_output: 200
- B4_output: 100
- PF2_received: 250
- PF3_received: 300
- PF2_output: 250

- PF3_output: 300
- total demand requested: 550
- demand point requirements met
- DP1_demand: 37 DP1_received: 37
- DP2_demand: 37 DP2_received: 37
- DP3_demand: 37 DP3_received: 37
- DP4_demand: 37 DP4_received: 37
- DP5_demand: 37 DP5_received: 37
- DP6_demand: 37 DP6_received: 37
- DP7_demand: 37 DP7_received: 37
- DP8_demand: 37 DP8_received: 37
- DP9_demand: 37 DP9_received: 37
- DP10_demand: 37 DP10_received: 37
- DP11_demand: 36 DP11_received: 36
- DP12_demand: 36 DP12_received: 36
- DP13_demand: 36 DP13_received: 36
- DP14_demand: 36 DP14_received: 36
- DP15_demand: 36 DP15_received: 36
- Total demand points received: 550
- Total shipping costs: $1606

If we close Packaging Facility 2 instead, shipping costs will be about the same to closing Facility 1 ($1600 vs 1606):

Route Distributions:

- route_B1_PF3 = 100
- route_B2_PF1 = 150
- route_B3_PF3 = 200
- route_B4_PF1 = 100
- route_PF1_DP10 = 37
- route_PF1_DP12 = 36
- route_PF1_DP14 = 36
- route_PF1_DP15 = 36
- route_PF1_DP3 = 31
- route_PF1_DP5 = 37
- route_PF1_DP7 = 37
- route_PF3_DP1 = 37
- route_PF3_DP11 = 36
- route_PF3_DP13 = 36
- route_PF3_DP2 = 37
- route_PF3_DP3 = 6
- route_PF3_DP4 = 37
- route_PF3_DP6 = 37
- route_PF3_DP8 = 37
- route_PF3_DP9 = 37
- total_brewery_output: 550
- total_packaging_facility_received: 550
- total_packaging_facility_output: 550

- total_demand_point_received: 550
- B1_output: 100
- B2_output: 150
- B3_output: 200
- B4_output: 100
- PF1_received: 250
- PF3_received: 300
- PF1_output: 250
- PF3_output: 300
- total demand requested: 550
- demand point requirements met
- DP1_demand: 37 DP1_received: 37
- DP2_demand: 37 DP2_received: 37
- DP3_demand: 37 DP3_received: 37
- DP4_demand: 37 DP4_received: 37
- DP5_demand: 37 DP5_received: 37
- DP6_demand: 37 DP6_received: 37
- DP7_demand: 37 DP7_received: 37
- DP8_demand: 37 DP8_received: 37
- DP9_demand: 37 DP9_received: 37
- DP10_demand: 37 DP10_received: 37
- DP11_demand: 36 DP11_received: 36
- DP12_demand: 36 DP12_received: 36
- DP13_demand: 36 DP13_received: 36
- DP14_demand: 36 DP14_received: 36
- DP15_demand: 36 DP15_received: 36
- Total demand points received: 550
- Total shipping costs: $1600

(4) Try multiplying demand by 2, 3, 4, or higher multiples. You can do this by modifying one line of the Python program. You can see the initial setting: demand_multiplier = 1 Setting the multiplier to 2, 3, 4, or higher values, will increase the level of demand, which will change the optimal solution. At what point does demand exceed the company's production capacity? At this point (full capacity), would you close any of the breweries or packaging facilities?

In [76]:
```
# reset to original demand mapping
demand_point_units_ordered = {"DP1": 48, "DP2": 84, "DP3": 64,
                "DP4": 106, "DP5": 47, "DP6": 57,
                "DP7": 64, "DP8": 93, "DP9": 74,
                "DP10": 41, "DP11": 61, "DP12": 42,
                "DP13": 57, "DP14": 70, "DP15": 41}
```

In [77]:
```
demand = calculate_demand(demand_point_units_ordered, 2)
prob = solve_network_problem(demand)
print_output(prob)
```

```
Status:
 Optimal

Route Distributions:
route_B1_PF2 = 550
route_B2_PF1 = 158
route_B3_PF3 = 1090
route_B4_PF1 = 100
route_PF1_DP10 = 82
route_PF1_DP15 = 82
route_PF1_DP5 = 94
route_PF2_DP1 = 96
route_PF2_DP14 = 140
route_PF2_DP7 = 128
route_PF2_DP8 = 186
route_PF3_DP11 = 122
route_PF3_DP12 = 84
route_PF3_DP13 = 114
route_PF3_DP2 = 168
route_PF3_DP3 = 128
route_PF3_DP4 = 212
route_PF3_DP6 = 114
route_PF3_DP9 = 148

total_brewery_output: 1898
total_packaging_facility_received: 1898
total_packaging_facility_output: 1898
total_demand_point_received: 1898

B1_output: 550
B2_output: 158
B3_output: 1090
B4_output: 100

PF1_received: 258
PF2_received: 550
PF3_received: 1090

PF1_output: 258
PF2_output: 550
PF3_output: 1090

total demand requested: 1898
demand point requirements met
DP1_demand: 96 DP1_received: 96
DP2_demand: 168 DP2_received: 168
DP3_demand: 128 DP3_received: 128
DP4_demand: 212 DP4_received: 212
DP5_demand: 94 DP5_received: 94
DP6_demand: 114 DP6_received: 114
DP7_demand: 128 DP7_received: 128
DP8_demand: 186 DP8_received: 186
DP9_demand: 148 DP9_received: 148
DP10_demand: 82 DP10_received: 82
DP11_demand: 122 DP11_received: 122
```

```
DP12_demand: 84 DP12_received: 84
DP13_demand: 114 DP13_received: 114
DP14_demand: 140 DP14_received: 140
DP15_demand: 82 DP15_received: 82

Total demand points received: 1898

Total shipping costs: $4641
```

In [78]:

```
demand = calculate_demand(demand_point_units_ordered, 3)
prob = solve_network_problem(demand)
print_output(prob)
```

```
Status:
 Optimal

Route Distributions:
route_B1_PF2 = 825
route_B2_PF1 = 287
route_B3_PF3 = 1635
route_B4_PF1 = 100
route_PF1_DP10 = 123
route_PF1_DP15 = 123
route_PF1_DP5 = 141
route_PF2_DP1 = 144
route_PF2_DP14 = 210
route_PF2_DP7 = 192
route_PF2_DP8 = 279
route_PF3_DP11 = 183
route_PF3_DP12 = 126
route_PF3_DP13 = 171
route_PF3_DP2 = 252
route_PF3_DP3 = 192
route_PF3_DP4 = 318
route_PF3_DP6 = 171
route_PF3_DP9 = 222

total_brewery_output: 2847
total_packaging_facility_received: 2847
total_packaging_facility_output: 2847
total_demand_point_received: 2847

B1_output: 825
B2_output: 287
B3_output: 1635
B4_output: 100

PF1_received: 387
PF2_received: 825
PF3_received: 1635

PF1_output: 387
PF2_output: 825
PF3_output: 1635

total demand requested: 2847
demand point requirements met
DP1_demand: 144 DP1_received: 144
DP2_demand: 252 DP2_received: 252
DP3_demand: 192 DP3_received: 192
DP4_demand: 318 DP4_received: 318
DP5_demand: 141 DP5_received: 141
DP6_demand: 171 DP6_received: 171
DP7_demand: 192 DP7_received: 192
DP8_demand: 279 DP8_received: 279
DP9_demand: 222 DP9_received: 222
DP10_demand: 123 DP10_received: 123
DP11_demand: 183 DP11_received: 183
```

```
DP12_demand: 126 DP12_received: 126
DP13_demand: 171 DP13_received: 171
DP14_demand: 210 DP14_received: 210
DP15_demand: 123 DP15_received: 123

Total demand points received: 2847

Total shipping costs: $6941
```

In [79]:

```
demand = calculate_demand(demand_point_units_ordered, 4)
prob = solve_network_problem(demand)
print_output(prob)
```

```
Status:
 Optimal

Route Distributions:
route_B1_PF2 = 1100
route_B2_PF1 = 416
route_B3_PF3 = 2180
route_B4_PF1 = 100
route_PF1_DP10 = 164
route_PF1_DP15 = 164
route_PF1_DP5 = 188
route_PF2_DP1 = 192
route_PF2_DP14 = 280
route_PF2_DP7 = 256
route_PF2_DP8 = 372
route_PF3_DP11 = 244
route_PF3_DP12 = 168
route_PF3_DP13 = 228
route_PF3_DP2 = 336
route_PF3_DP3 = 256
route_PF3_DP4 = 424
route_PF3_DP6 = 228
route_PF3_DP9 = 296

total_brewery_output: 3796
total_packaging_facility_received: 3796
total_packaging_facility_output: 3796
total_demand_point_received: 3796

B1_output: 1100
B2_output: 416
B3_output: 2180
B4_output: 100

PF1_received: 516
PF2_received: 1100
PF3_received: 2180

PF1_output: 516
PF2_output: 1100
PF3_output: 2180

total demand requested: 3796
demand point requirements met
DP1_demand: 192 DP1_received: 192
DP2_demand: 336 DP2_received: 336
DP3_demand: 256 DP3_received: 256
DP4_demand: 424 DP4_received: 424
DP5_demand: 188 DP5_received: 188
DP6_demand: 228 DP6_received: 228
DP7_demand: 256 DP7_received: 256
DP8_demand: 372 DP8_received: 372
DP9_demand: 296 DP9_received: 296
DP10_demand: 164 DP10_received: 164
DP11_demand: 244 DP11_received: 244
```

```
DP12_demand: 168 DP12_received: 168
DP13_demand: 228 DP13_received: 228
DP14_demand: 280 DP14_received: 280
DP15_demand: 164 DP15_received: 164

Total demand points received: 3796

Total shipping costs: $9241
```

In [80]:
```
demand = calculate_demand(demand_point_units_ordered, 5)
prob = solve_network_problem(demand)
print_output(prob)
```

```
Status:
 Undefined

Route Distributions:

total_brewery_output: 0
total_packaging_facility_received: 0
total_packaging_facility_output: 0
total_demand_point_received: 0

B1_output: 0
B2_output: 0
B3_output: 0
B4_output: 0

PF1_received: 0
PF2_received: 0
PF3_received: 0

PF1_output: 0
PF2_output: 0
PF3_output: 0

total demand requested: 4745
demand point requirements not met
DP1_demand: 240 DP1_received: 0
DP2_demand: 420 DP2_received: 0
DP3_demand: 320 DP3_received: 0
DP4_demand: 530 DP4_received: 0
DP5_demand: 235 DP5_received: 0
DP6_demand: 285 DP6_received: 0
DP7_demand: 320 DP7_received: 0
DP8_demand: 465 DP8_received: 0
DP9_demand: 370 DP9_received: 0
DP10_demand: 205 DP10_received: 0
DP11_demand: 305 DP11_received: 0
DP12_demand: 210 DP12_received: 0
DP13_demand: 285 DP13_received: 0
DP14_demand: 350 DP14_received: 0
DP15_demand: 205 DP15_received: 0

Total demand points received: 0

Total shipping costs: $0
```

Setting the deamand multiplier to 5 'breaks' the system as Pulp solves it as 'Undefined' since 4745 units of beer exceed the total capacity for all three packaging facilities (4500 total).

Despite the increasing demand multiplier, brewery 4 still operates at minimum capcity due to its highest average shipping costs compared to the other breweries. If we're sticking to the rule presented in question two (close brewery if at minimum capacity), closing brewery 4 would still be suggested with rising demand.

At demand multipler 4 packaging facility 1 is over-exceeding its capacity by 16 units while facilities 2 and 3 are comfortably within their respective capacities. As long as total beer units in the network don't exceed 4000, Facilities 2 and 3 can operate comfortably and within reasonable costs by leveraging their capacities and average costs to ship to and ship out, especially with . Though Facility 2 might have more importing costs and exporting costs than Facility 1, its larger maximum capacity makes it more useful when the demand multiplier increases, and will leverage more facility 2's lower cost routes and facility 3's overall lower costs.

That being said, again since we are not sure of the operating costs for each packaging facility and how it can offset or add to the transportation costs of each facility, we cannot make a full assessment for ultimately deciding which packaging facility to close with increasing demand

(5) What have you learned from this supply chain optimization problem? Explain how you might apply methods of constrained optimization in your line of work.

I found that LP and the network flow problem is very helpful for figuring out the flow of units between places given the constraints, costs, and demands.

I can imagine with the current chip shortage and the high demand for graphics cards, this can be applied to figuring out a chip manufacturer (like Nvidia) would ship out their baseline cards from their factories to 3rd-party manufacturers and their respective factories like Asus or MSI which would make their own casing for the card, and these 3rd parties ship out the completed cards to distirbutors, and from there they would meet the demand of retailers such as a brick-and-motor shops like Best Buy or an online store like Amazon. Given the shipping costs between each layer, the capacity constraints of each one, and the demands of the stores the network flow problem would help figure out how many units of cards to send between the nodes of each layer. That said, because of the overall chip shortage, meeting the minimum capacities of each manufacturer and distribution center is a challenge in itself and stores are not able to meet the current demand for graphics cards by gamers, media/content creators, or crypocurrency miners.

The Distribution Planning for a Brewery case represents a constrained optimization problem (more specifically, a transshipment problem in mathematical programming). We are minimizing costs subject to production and demand constraints.

Another way to approach a transshipment problem would be to consider the network structure across nodes for the breweries, packaging facilities, and demand points. We could vary shipping distances, times, or costs across paths (edges/links) between locations. A discrete event simulation could be used to trace the flow of beer products from node to node. A common discrete event simulation would model the transshipment problem as a network of queues. In week 7, we introduce discrete event simulation.

Are supply chain optimization and logistics management important to companies? You bet they are. Take note of this story from siliconANGLE:

https://siliconangle.com/2022/02/07/flexport-raises-935m-8b-valuation-logistics-management-platform/ (https://siliconangle.com/2022/02/07/flexport-raises-935m-8b-valuation-logistics-management-platform/)