Transshipment With All Plants

November 5, 2016

1 Allow manufacturing anywhere.

In [2]:
```
using JuMP
using Clp
```

```
INFO: Precompiling module JuMP.
INFO: Precompiling module Clp.
```

2 2016 - All Manufacturing

In [16]:
```
ORIG = ["LAF", "MDP", "PAB"];
DEST = ["LAF", "MDP", "PAB", "ILN", "BLG", "JPN", "CHN", "MEX", "PER", "SINK"];

supply = [16 16 15];
sinkD = sum(supply)- sum([0 0 0 8.86 0.74 8.26 8.86 2.96 4.42])
demand = [ 0 0 0 8.86 0.74 8.26 8.86 2.96 4.42 sinkD];

@assert sum(supply) == sum(demand);

cost = [
  0   750  650   120   1250   2580   2300   450   720   0;
600     0  120   720   1330   1550   1450   450   220   0;
700   150    0   670   1350   1550   1450   400   200   0
]

m = Model();

@variable(m, Trans[i=1:length(ORIG), j=1:length(DEST)] >= 0);


@objective(m, Min, sum{cost[i,j] * Trans[i,j], i=1:length(ORIG), j=1:length(DEST)});

@constraint(m, xyconstr[i=1:1:length(ORIG)], sum{Trans[i,j], j=1:length(DEST)} == supply[i]);

@constraint(m, xyconstr[j = 1:length(DEST)], sum{Trans[i,j], i=1:length(ORIG)} == demand[j]);

println("Solving original problem...")
status = solve(m);

if status == :Optimal
    @printf("Optimal!\n");
    @printf("Objective value: %.2f\n", getobjectivevalue(m));
    @printf("Transpotation:\n");
```

```
            for j = 1:length(DEST)
                @printf("\t%s", DEST[j]);
            end
            @printf("\n");
            for i = 1:length(ORIG)
                @printf("%s", ORIG[i]);
                for j = 1:length(DEST)
                    @printf("\t%.2f", getvalue(Trans[i,j]));
                end
                @printf("\n");
            end
        else
            @printf("No solution\n");
        end
```

```
Solving original problem...
Optimal!
Objective value: 29706.20
Transpotation:
```

	LAF	MDP	PAB	ILN	BLG	JPN	CHN	MEX	PER
LAF	0.00	0.00	0.00	8.86	0.74	0.00	0.00	0.00	
MDP	0.00	0.00	0.00	0.00	0.00	8.26	7.74	0.00	
PAB	0.00	0.00	0.00	0.00	0.00	0.00	1.12	2.96	

3 2017 - All Manufacturing

In [17]:
```
ORIG = ["LAF", "MDP", "PAB"];
DEST = ["LAF", "MDP", "PAB", "ILN", "BLG", "JPN", "CHN", "MEX", "PER", "SINK"];

supply = [16 18 18];
sinkD = sum(supply)- sum([ 9.66 0.81 9 9.66 3.22 4.83])
demand = [ 0 0 0 9.66 0.81 9 9.66 3.22 4.83 sinkD];

@assert sum(supply) == sum(demand);

cost = [
  0   750  650   120   1250   2580   2300   450   720   0;
600     0  120   720   1330   1550   1450   450   220   0;
700   150    0   670   1350   1550   1450   400   200   0
]

m = Model();

@variable(m, Trans[i=1:length(ORIG), j=1:length(DEST)] >= 0);


@objective(m, Min, sum{cost[i,j] * Trans[i,j], i=1:length(ORIG), j=1:length(DEST)});

@constraint(m, xyconstr[i=1:1:length(ORIG)], sum{Trans[i,j], j=1:length(DEST)} == supply[i]);

@constraint(m, xyconstr[j = 1:length(DEST)], sum{Trans[i,j], i=1:length(ORIG)} == demand[j]);

println("Solving original problem...")
status = solve(m);
```

```
if status == :Optimal
    @printf("Optimal!\n");
    @printf("Objective value: %.2f\n", getobjectivevalue(m));
    @printf("Transpotation:\n");
    for j = 1:length(DEST)
        @printf("\t%s", DEST[j]);
    end
    @printf("\n");
    for i = 1:length(ORIG)
        @printf("%s", ORIG[i]);
        for j = 1:length(DEST)
            @printf("\t%.2f", getvalue(Trans[i,j]));
        end
        @printf("\n");
    end
else
    @printf("No solution\n");
end
```

```
Solving original problem...
Optimal!
Objective value: 32382.70
Transpotation:
        LAF         MDP         PAB         ILN         BLG         JPN         CHN         MEX         PER
LAF        0.00        0.00        0.00        9.66        0.81        0.00        0.00        0.00
MDP        0.00        0.00        0.00        0.00        0.00        9.00        9.00        0.00
PAB        0.00        0.00        0.00        0.00        0.00        0.00        0.66        3.22
```

4 2018 - All manufacturing

In [18]:
```
ORIG = ["LAF", "MDP", "PAB"];
DEST = ["LAF", "MDP", "PAB", "ILN", "BLG", "JPN", "CHN", "MEX", "PER", "SINK"];

supply = [21 18 12];
sinkD = sum(supply)- sum([11.88 0.99 11.09 11.88 3.96 5.94])
demand = [ 0 0 0 11.88 0.99 11.09 11.88 3.96 5.94 sinkD];

@assert sum(supply) == sum(demand);

cost = [
  0   750  650   120   1250   2580   2300   450   720   0;
600     0  120   720   1330   1550   1450   450   220   0;
700   150    0   670   1350   1550   1450   400   200   0
]

m = Model();

@variable(m, Trans[i=1:length(ORIG), j=1:length(DEST)] >= 0);


@objective(m, Min, sum{cost[i,j] * Trans[i,j], i=1:length(ORIG), j=1:length(DEST)});

@constraint(m, xyconstr[i=1:1:length(ORIG)], sum{Trans[i,j], j=1:length(DEST)} == supply[i]);
```

```
@constraint(m, xyconstr[j = 1:length(DEST)], sum{Trans[i,j], i=1:length(ORIG)} == demand[j]);

println("Solving original problem...")
status = solve(m);

if status == :Optimal
    @printf("Optimal!\n");
    @printf("Objective value: %.2f\n", getobjectivevalue(m));
    @printf("Transpotation:\n");
    for j = 1:length(DEST)
        @printf("\t%s", DEST[j]);
    end
    @printf("\n");
    for i = 1:length(ORIG)
        @printf("%s", ORIG[i]);
        for j = 1:length(DEST)
            @printf("\t%.2f", getvalue(Trans[i,j]));
        end
        @printf("\n");
    end
else
    @printf("No solution\n");
end
```

```
Solving original problem...
Optimal!
Objective value: 39994.10
Transpotation:
        LAF         MDP         PAB         ILN         BLG         JPN         CHN         MEX         PER
LAF        0.00        0.00        0.00       11.88        0.99        0.00        0.00        2.87
MDP        0.00        0.00        0.00        0.00        0.00        6.12       11.88        0.00
PAB        0.00        0.00        0.00        0.00        0.00        4.97        0.00        1.09
```

5 2019 - All manufacturing

In [19]:
```
ORIG = ["LAF", "MDP", "PAB"];
DEST = ["LAF", "MDP", "PAB", "ILN", "BLG", "JPN", "CHN", "MEX", "PER", "SINK"];

supply = [22 20 20];
sinkD = sum(supply)- sum([ 0 0 0 12.56 1.04 11.64 12.47 4.16 6.24])
demand = [ 0 0 0 12.56 1.04 11.64 12.47 4.16 6.24 sinkD];

@assert sum(supply) == sum(demand);

cost = [
  0   750  650   120   1250   2580   2300   450   720   0;
600     0  120   720   1330   1550   1450   450   220   0;
700   150    0   670   1350   1550   1450   400   200   0
]

m = Model();

@variable(m, Trans[i=1:length(ORIG), j=1:length(DEST)] >= 0);
```

```
@objective(m, Min, sum{cost[i,j] * Trans[i,j], i=1:length(ORIG), j=1:length(DEST)});

@constraint(m, xyconstr[i=1:1:length(ORIG)], sum{Trans[i,j], j=1:length(DEST)} == supply[i]);

@constraint(m, xyconstr[j = 1:length(DEST)], sum{Trans[i,j], i=1:length(ORIG)} == demand[j]);

println("Solving original problem...")
status = solve(m);

if status == :Optimal
    @printf("Optimal!\n");
    @printf("Objective value: %.2f\n", getobjectivevalue(m));
    @printf("Transpotation:\n");
    for j = 1:length(DEST)
        @printf("\t%s", DEST[j]);
    end
    @printf("\n");
    for i = 1:length(ORIG)
        @printf("%s", ORIG[i]);
        for j = 1:length(DEST)
            @printf("\t%.2f", getvalue(Trans[i,j]));
        end
        @printf("\n");
    end
else
    @printf("No solution\n");
end
```

```
Solving original problem...
Optimal!
Objective value: 41842.70
Transpotation:
        LAF         MDP         PAB         ILN         BLG         JPN         CHN         MEX         PER
LAF        0.00        0.00        0.00       12.56        1.04        0.00        0.00        0.00
MDP        0.00        0.00        0.00        0.00        0.00       11.64        8.36        0.00
PAB        0.00        0.00        0.00        0.00        0.00        0.00        4.11        4.16
```

6 2020 - All manufacturing

In [20]:
```
ORIG = ["LAF", "MDP", "PAB"];
DEST = ["LAF", "MDP", "PAB", "ILN", "BLG", "JPN", "CHN", "MEX", "PER", "SINK"];

supply = [28 24 21];
sinkD = sum(supply)- sum([ 14.02 1.17 13.09 14.02 4.67 7.01])
demand = [ 0 0 0 14.02 1.17 13.09 14.02 4.67 7.01 sinkD];

@assert sum(supply) == sum(demand);

cost = [
  0   750  650   120   1250   2580   2300   450   720   0;
600     0  120   720   1330   1550   1450   450   220   0;
700   150    0   670   1350   1550   1450   400   200   0
```

```
]

m = Model();

@variable(m, Trans[i=1:length(ORIG), j=1:length(DEST)] >= 0);


@objective(m, Min, sum{cost[i,j] * Trans[i,j], i=1:length(ORIG), j=1:length(DEST)});

@constraint(m, xyconstr[i=1:1:length(ORIG)], sum{Trans[i,j], j=1:length(DEST)} == supply[i]);

@constraint(m, xyconstr[j = 1:length(DEST)], sum{Trans[i,j], i=1:length(ORIG)} == demand[j]);

println("Solving original problem...")
status = solve(m);

if status == :Optimal
    @printf("Optimal!\n");
    @printf("Objective value: %.2f\n", getobjectivevalue(m));
    @printf("Transpotation:\n");
    for j = 1:length(DEST)
        @printf("\t%s", DEST[j]);
    end
    @printf("\n");
    for i = 1:length(ORIG)
        @printf("%s", ORIG[i]);
        for j = 1:length(DEST)
            @printf("\t%.2f", getvalue(Trans[i,j]));
        end
        @printf("\n");
    end
else
    @printf("No solution\n");
end
```

```
Solving original problem...
Optimal!
Objective value: 47033.40
Transpotation:
        LAF        MDP        PAB        ILN        BLG        JPN        CHN        MEX        PER
LAF        0.00        0.00        0.00        14.02        1.17        0.00        0.00        0.00
MDP        0.00        0.00        0.00        0.00        0.00        13.09        10.91        0.00
PAB        0.00        0.00        0.00        0.00        0.00        0.00        3.11        4.67
```

In [ ]: