ARG+BRZ

November 8, 2016

```
In [1]: using JuMP
```

1 2016

```
In [2]: ORIG = ["MDP", "PAB", "LAF-MDP", "LAF-PAB"];
        DEST = ["ILN", "BLG", "JPN", "CHN", "MEX", "PER", "SINK"];

        supply = [16 15 16 16];
        US=16
        demand = [8.86 0.74 8.26 8.86 2.96 4.42 0]


        #@assert sum(supply) == sum(demand);
        la = 750
        lb = 650
        cost = [
        720         1330      1550      1450      450      220   0;
        670         1350      1550      1450      400      200   0;
        720+la   1330+la   1550+la   1450+la   450+la   220+la   0;
        670+lb   1350+lb   1550+lb   1450+lb   400+lb   200+lb   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]);

        @constraint(m, xyconstr[j = 1:length(DEST)], sum{Trans[i,j], i=3:(length(ORIG)-1)} <= US);


        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: 36653.40
Transpotation:
        ILN        BLG        JPN        CHN        MEX        PER        SINK
MDP        0.00        0.74        8.26        7.00        0.00        0.00        0.00
PAB        5.76        0.00        0.00        1.86        2.96        4.42        0.00
LAF-MDP        0.00        0.00        0.00        0.00        0.00        0.00        16.00
LAF-PAB        3.10        0.00        0.00        0.00        0.00        0.00        12.90
```

2 2017

```
In [3]: ORIG = ["MDP", "PAB", "LAF-MDP", "LAF-PAB"];
        DEST = ["ILN", "BLG", "JPN", "CHN", "MEX", "PER", "SINK"];

        supply = [18 18 16 16];
        US=16
        demand = [9.66 0.81 9 9.66 3.22 4.83 0]


        #@assert sum(supply) == sum(demand);
        la = 750
        lb = 650
        cost = [
        720         1330      1550      1450      450      220   0;
        670         1350      1550      1450      400      200   0;
        720+la   1330+la   1550+la   1450+la   450+la   220+la   0;
        670+lb   1350+lb   1550+lb   1450+lb   400+lb   200+lb   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]);

        @constraint(m, xyconstr[j = 1:length(DEST)], sum{Trans[i,j], i=3:(length(ORIG)-1)} <= US);


        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: 38527.50
Transpotation:
        ILN        BLG        JPN        CHN        MEX        PER        SINK
MDP        0.00        0.81        9.00        8.19        0.00        0.00        0.00
PAB        8.48        0.00        0.00        1.47        3.22        4.83        0.00
LAF-MDP        0.00        0.00        0.00        0.00        0.00        0.00        16.00
LAF-PAB        1.18        0.00        0.00        0.00        0.00        0.00        14.82
```

3 2018

```
In [4]: ORIG = ["MDP", "PAB", "LAF-MDP", "LAF-PAB"];
        DEST = ["ILN", "BLG", "JPN", "CHN", "MEX", "PER", "SINK"];

        supply = [12 18 21 21];
        US=21
        demand = [11.88 0.99 11.09 11.88 3.96 5.94 0]


        #@assert sum(supply) == sum(demand);
        la = 750
        lb = 650
        cost = [
        720         1330      1550      1450      450         220    0;
        670         1350      1550      1450      400         200    0;
        720+la   1330+la   1550+la   1450+la   450+la   220+la   0;
        670+lb   1350+lb   1550+lb   1450+lb   400+lb   200+lb   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]);

        @constraint(m, xyconstr[j = 1:length(DEST)], sum{Trans[i,j], i=3:(length(ORIG)-1)} <= US);


        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: 56694.80
Transpotation:
        ILN        BLG        JPN        CHN        MEX        PER        SINK
MDP        0.00        0.99        7.23        3.78        0.00        0.00        0.00
PAB        0.00        0.00        0.00        8.10        3.96        5.94        0.00
LAF-MDP        0.00        0.00        0.00        0.00        0.00        0.00        21.00
LAF-PAB        11.88        0.00        3.86        0.00        0.00        0.00        5.26
```

4 2019

```
In [5]: ORIG = ["MDP", "PAB", "LAF-MDP", "LAF-PAB"];
        DEST = ["ILN", "BLG", "JPN", "CHN", "MEX", "PER", "SINK"];
```

```
supply = [20 20 22 22];
US=22
demand = [12.56 1.04 11.64 12.47 4.16 6.24 0]


#@assert sum(supply) == sum(demand);
la = 750
lb = 650
cost = [
720         1330      1550      1450      450      220   0;
670         1350      1550      1450      400      200   0;
720+la   1330+la   1550+la   1450+la   450+la   220+la   0;
670+lb   1350+lb   1550+lb   1450+lb   400+lb   200+lb   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]);

@constraint(m, xyconstr[j = 1:length(DEST)], sum{Trans[i,j], i=3:(length(ORIG)-1)} <= US);


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: 54105.40
```

```
Transpotation:
        ILN        BLG        JPN        CHN        MEX        PER        SINK
MDP        0.00        1.04        11.64        7.32        0.00        0.00        0.00
PAB        4.45        0.00        0.00        5.15        4.16        6.24        0.00
LAF-MDP        0.00        0.00        0.00        0.00        0.00        0.00        22.00
LAF-PAB        8.11        0.00        0.00        0.00        0.00        0.00        13.89
```

5 2020

```
In [6]: ORIG = ["MDP", "PAB", "LAF-MDP", "LAF-PAB"];
        DEST = ["ILN", "BLG", "JPN", "CHN", "MEX", "PER", "SINK"];

        supply = [21 24 28 28];
        US=22
        demand = [14.02 1.17 13.09 14.02 4.67 7.01 0]

        #@assert sum(supply) == sum(demand);
        la = 750
        lb = 650
        cost = [
        720         1330      1550      1450      450      220   0;
        670         1350      1550      1450      400      200   0;
        720+la   1330+la   1550+la   1450+la   450+la   220+la   0;
        670+lb   1350+lb   1550+lb   1450+lb   400+lb   200+lb   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]);

        @constraint(m, xyconstr[j = 1:length(DEST)], sum{Trans[i,j], i=3:(length(ORIG)-1)} <= US);


        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: 61275.00
Transpotation:
        ILN        BLG        JPN        CHN        MEX        PER        SINK
MDP        0.00        1.17        7.09        12.74        0.00        0.00        0.00
PAB        11.04        0.00        0.00        1.28        4.67        7.01        0.00
LAF-MDP        0.00        0.00        6.00        0.00        0.00        0.00        22.00
LAF-PAB        2.98        0.00        0.00        0.00        0.00        0.00        25.02
```

In [ ]: