Problem1

October 18, 2016

In [13]:
```
# Assuming 13 weeks per quarter
alpha = 0.8
beta = 0.7
A = [0 2*alpha/5 2*alpha/5 0 alpha/5 0 0 0 0 1-alpha
    beta/11 0 beta/11 4*beta/11 0 4*beta/11 0 0 0 1-beta
    alpha/2 0 0 0 0 0 0 0 alpha/2 1-alpha
    0 alpha/4 0 0 0 0 alpha/2 alpha/4 0 1-alpha
    0 alpha 0 0 0 0 0 0 0 1-alpha
    2*alpha/3 alpha/3 0 0 0 0 0 0 0 1-alpha
    0 alpha 0 0 0 0 0 0 0 1-alpha
    0 alpha 0 0 0 0 0 0 0 1-alpha
    0 0 0 0 0 0 0 0 1 0
    0 0 0 0 0 0 0 0 0 1]

# V, D = eig(A)
# Steady_State = V[:,1]/sum(V[:,1])
```

Out[13]:
```
10x10 Array{Float64,2}:
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.193462   0.755733
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.090845   0.810266
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.477385   0.502293
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0617746  0.870981
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.072676   0.848213
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.127405   0.819128
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.072676   0.848213
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.072676   0.848213
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0        0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0        1.0
```

In [253]:
```
function productACapacity(maxMachine)
    tEnd = 10
    hourWeek = round(Int64,((21 * 7) - 20) * 0.85)
    InvStations = zeros(8, hourWeek*(tEnd+1))
    processInv = zeros(30, hourWeek*(tEnd+1))
    finishedInv = zeros(1, hourWeek*(tEnd+1))
    speed = [25.3, 23.2, 100.5, 15.5, 32.9, 60.0, 50.2, 33.5]
    #maxMachine = [1 1 1 1 1 1 1 1]
    t = 0
    hour = 0

    while t< tEnd
        t+=1
        endHour = (t)*hourWeek
        hour = (t-1)*hourWeek
```

```
finishedInv[hour+1:end] = 0
while hour < endHour
    #@show hour
    hour += 1
    InvStations[1, hour:end] += speed[1]*2
    notBusy1 = true
    notBusy2 = true
    notBusy3 = true
    notBusy4 = true
    notBusy5 = true
    notBusy6 = true
    notBusy7 = true
    notBusy8 = true
    nMacUse = [0 0 0 0 0 0 0 0]
    #  At station 1
    while notBusy1
        if rand() <= 0.85
            if (processInv[15,hour] > 0) && notBusy1
                nProcess = min(speed[1], processInv[15,hour])
                InvStations[5,hour:end] += nProcess #* 0.8
                InvStations[6,hour:end] -= nProcess

                processInv[16,hour:end] += nProcess #* 0.8
                processInv[15, hour:end] -= nProcess

                nMacUse[1] += 1
                if nMacUse[1] == maxMachine[1]
                    notBusy1 = false
                end
            end
            if (processInv[7,hour] > 0) && notBusy1
                nProcess = min(speed[1], processInv[7,hour])
                InvStations[5,hour:end] += nProcess #* 0.8
                InvStations[3,hour:end] -= nProcess

                processInv[8,hour:end] += nProcess #* 0.8
                processInv[7, hour:end] -= nProcess

                nMacUse[1] += 1
                if nMacUse[1] == maxMachine[1]
                    notBusy1 = false
                end
            end
            if (processInv[5,hour] > 0) && notBusy1
                nProcess = min(speed[1], processInv[5,hour])
                InvStations[3,hour:end] += nProcess  #* 0.8
                InvStations[6,hour:end] -= nProcess

                processInv[6,hour:end] += nProcess #* 0.8
                processInv[5, hour:end] -= nProcess

                nMacUse[1] += 1
                if nMacUse[1] == maxMachine[1]
```

```
                notBusy1 = false
            end
        end
        if (processInv[2,hour] > 0) && notBusy1
            nProcess = min(speed[1], processInv[2,hour])
            InvStations[2,hour:end] += nProcess #* 0.8
            InvStations[1,hour:end] -= nProcess

            processInv[3,hour:end] += nProcess #* 0.8
            processInv[2, hour:end] -= nProcess

            nMacUse[1] += 1
            if nMacUse[1] == maxMachine[1]
                notBusy1 = false
            end
        end
        if notBusy1
            nProcess = speed[1] #* 0.8
            InvStations[2,hour:end] += nProcess

            processInv[1,hour:end] += nProcess

            nMacUse[1] += 1
            if nMacUse[1] == maxMachine[1]
                notBusy1 = false
            end
        end
    else
        nMacUse[1] += 1
            if nMacUse[1] == maxMachine[1]
                notBusy1 = false
            end
    end
end
# At station 2
while notBusy2
    if rand() <= 0.6
        if (processInv[29,hour] > 0) && notBusy2
            nProcess = min(speed[2], processInv[29,hour])
            InvStations[3,hour:end] += nProcess #* 0.7
            InvStations[8,hour:end] -= nProcess

            processInv[30,hour:end] += nProcess #* 0.7
            processInv[29, hour:end] -= nProcess

            nMacUse[2] += 1
            if nMacUse[2] == maxMachine[2]
                notBusy2 = false
            end
        end
        if (processInv[26,hour] > 0) && notBusy2
            nProcess = min(speed[2], processInv[26,hour])
            InvStations[4,hour:end] += nProcess #* 0.7
            InvStations[7,hour:end] -= nProcess
```

```
    processInv[27,hour:end] += nProcess #* 0.7
    processInv[26, hour:end] -= nProcess

    nMacUse[2] += 1
    if nMacUse[2] == maxMachine[2]
        notBusy2 = false
    end
end
if (processInv[23,hour] > 0) && notBusy2
    nProcess = min(speed[2], processInv[23,hour])
    InvStations[4,hour:end] += nProcess #* 0.7
    InvStations[5,hour:end] -= nProcess

    processInv[24,hour:end] += nProcess #* 0.7
    processInv[23, hour:end] -= nProcess

    nMacUse[2] += 1
    if nMacUse[2] == maxMachine[2]
        notBusy2 = false
    end
end
if (processInv[20,hour] > 0) && notBusy2
    nProcess = min(speed[2], processInv[20,hour])
    InvStations[4,hour:end] += nProcess #* 0.7
    InvStations[7,hour:end] -= nProcess

    processInv[21,hour:end] += nProcess #* 0.7
    processInv[20, hour:end] -= nProcess

    nMacUse[2] += 1
    if nMacUse[2] == maxMachine[2]
        notBusy2 = false
    end
end
if (processInv[17,hour] > 0) && notBusy2
        nProcess = min(speed[2], processInv[17,hour])
        InvStations[4,hour:end] += nProcess #* 0.7
        InvStations[5,hour:end] -= nProcess

        processInv[18,hour:end] += nProcess #* 0.7
        processInv[17, hour:end] -= nProcess

        nMacUse[2] += 1
        if nMacUse[2] == maxMachine[2]
            notBusy2 = false
        end
end
if (processInv[13,hour] > 0) && notBusy2
    nProcess = min(speed[2], processInv[13,hour])
    InvStations[6,hour:end] += nProcess #* 0.7
    InvStations[6,hour:end] -= nProcess

    processInv[14,hour:end] += nProcess #* 0.7
```

```
        processInv[13, hour:end] -= nProcess

        nMacUse[2] += 1
        if nMacUse[2] == maxMachine[2]
            notBusy2 = false
        end
    end
    if (processInv[11,hour] > 0) && notBusy2
        nProcess = min(speed[2], processInv[11,hour])
        InvStations[6,hour:end] += nProcess #* 0.7
        InvStations[4,hour:end] -= nProcess

        processInv[12,hour:end] += nProcess #* 0.7
        processInv[11, hour:end] -= nProcess

        nMacUse[2] += 1
        if nMacUse[2] == maxMachine[2]
            notBusy2 = false
        end
    end
    if (processInv[9,hour] > 0) && notBusy2
            nProcess = min(speed[2], processInv[9,hour])
            InvStations[4,hour:end] += nProcess #* 0.7
            InvStations[5,hour:end] -= nProcess

            processInv[10,hour:end] += nProcess #* 0.7
            processInv[9, hour:end] -= nProcess

            nMacUse[2] += 1
            if nMacUse[2] == maxMachine[2]
                notBusy2 = false
            end
    end
    if (processInv[3,hour] > 0) && notBusy2
        nProcess = min(speed[2], processInv[3,hour])
        InvStations[6,hour:end] += nProcess #* 0.7
        InvStations[1,hour:end] -= nProcess

        processInv[4,hour:end] += nProcess #* 0.7
        processInv[3, hour:end] -= nProcess

        nMacUse[2] += 1
        if nMacUse[2] == maxMachine[2]
            notBusy2 = false
        end
    end
    if (processInv[1,hour] > 0) && notBusy2
        nProcess = min(speed[2], processInv[1,hour])
        InvStations[1,hour:end] += nProcess #* 0.7
        InvStations[1,hour:end] -= nProcess

        processInv[2,hour:end] += nProcess #* 0.7
        processInv[1, hour:end] -= nProcess
```

```
            nMacUse[2] += 1
            if nMacUse[2] == maxMachine[2]
                notBusy2 = false
            end
        end
    else
        nMacUse[2] += 1
            if nMacUse[2] == maxMachine[2]
                notBusy2 = false
            end
    end
end

# At station 3
while notBusy3
    if rand() <= 0.80
        if (processInv[30,hour] > 0) && notBusy3
            nProcess = min(speed[3], processInv[30,hour])
            finishedInv[hour:end] += nProcess #* 0.8
            InvStations[2,hour:end] -= nProcess

            processInv[30, hour:end] -= nProcess

            nMacUse[3] += 1
            if nMacUse[3] == maxMachine[3]
                notBusy3 = false
            end
        end
        if (processInv[6,hour] > 0) && notBusy3
            nProcess = min(speed[3], processInv[6,hour])
            InvStations[1,hour:end] += nProcess #* 0.8
            InvStations[1,hour:end] -= nProcess

            processInv[7,hour:end] += nProcess #* 0.8
            processInv[6, hour:end] -= nProcess

            nMacUse[3] += 1
            if nMacUse[3] == maxMachine[3]
                notBusy3 = false
            end
        end
    else
        nMacUse[3] += 1
        if nMacUse[3] == maxMachine[3]
            notBusy3 = false
        end
    end
end

# At station 4
while notBusy4
    if rand() <= 0.60
        if (processInv[27,hour] > 0) && notBusy4
            nProcess = min(speed[4], processInv[27,hour])
```

```
        InvStations[8,hour:end] += nProcess# * 0.8
        InvStations[2,hour:end] -= nProcess

        processInv[28,hour:end] += nProcess #* 0.8
        processInv[27, hour:end] -= nProcess

        nMacUse[4] += 1
        if nMacUse[4] == maxMachine[4]
            notBusy4 = false
        end
    end
    if (processInv[24,hour] > 0) && notBusy4
        nProcess = min(speed[4], processInv[24,hour])
        InvStations[5,hour:end] += nProcess# * 0.8
        InvStations[2,hour:end] -= nProcess

        processInv[25,hour:end] += nProcess# * 0.8
        processInv[24, hour:end] -= nProcess

        nMacUse[4] += 1
        if nMacUse[4] == maxMachine[4]
            notBusy4 = false
        end
    end
    if (processInv[21,hour] > 0) && notBusy4
        nProcess = min(speed[4], processInv[21,hour])
        InvStations[7,hour:end] += nProcess# * 0.8
        InvStations[2,hour:end] -= nProcess

        processInv[22,hour:end] += nProcess# * 0.8
        processInv[21, hour:end] -= nProcess

        nMacUse[4] += 1
        if nMacUse[4] == maxMachine[4]
            notBusy4 = false
        end
    end
    if (processInv[18,hour] > 0) && notBusy4
        nProcess = min(speed[4], processInv[18,hour])
        InvStations[7,hour:end] += nProcess# * 0.8
        InvStations[2,hour:end] -= nProcess

        processInv[19,hour:end] += nProcess #* 0.8
        processInv[18, hour:end] -= nProcess

        nMacUse[4] += 1
        if nMacUse[4] == maxMachine[4]
            notBusy4 = false
        end
    end
    if (processInv[10,hour] > 0) && notBusy4
        nProcess = min(speed[4], processInv[10,hour])
        InvStations[2,hour:end] += nProcess# * 0.8
        InvStations[2,hour:end] -= nProcess
```

```
            processInv[11,hour:end] += nProcess# * 0.8
            processInv[10, hour:end] -= nProcess

            nMacUse[4] += 1
            if nMacUse[4] == maxMachine[4]
                notBusy4 = false
            end
        end
    else
        nMacUse[4] += 1
        if nMacUse[4] == maxMachine[4]
            notBusy4 = false
        end
    end
end
# At station 5
while notBusy5
    if rand() <= 0.85
        if (processInv[22,hour] > 0) && notBusy5
            nProcess = min(speed[5], processInv[22,hour])
            InvStations[2,hour:end] += nProcess #* 0.8
            InvStations[4,hour:end] -= nProcess

            processInv[23,hour:end] += nProcess #* 0.8
            processInv[22, hour:end] -= nProcess

            nMacUse[5] += 1
            if nMacUse[5] == maxMachine[5]
                notBusy5 = false
            end
        end
        if (processInv[16,hour] > 0) && notBusy5
            nProcess = min(speed[5], processInv[16,hour])
            InvStations[2,hour:end] += nProcess# * 0.8
            InvStations[1,hour:end] -= nProcess

            processInv[17,hour:end] += nProcess# * 0.8
            processInv[16, hour:end] -= nProcess

            nMacUse[5] += 1
            if nMacUse[5] == maxMachine[5]
                notBusy5 = false
            end
        end
        if (processInv[8,hour] > 0) && notBusy5
            nProcess = min(speed[5], processInv[8,hour])
            InvStations[2,hour:end] += nProcess #* 0.8
            InvStations[1,hour:end] -= nProcess

            processInv[9,hour:end] += nProcess #* 0.8
            processInv[8, hour:end] -= nProcess

            nMacUse[5] += 1
```

```
            if nMacUse[5] == maxMachine[5]
                notBusy5 = false
            end
        end
    else
        nMacUse[5] += 1
            if nMacUse[5] == maxMachine[5]
                notBusy5 = false
            end
    end
end
# At station 6
while notBusy6
    if rand() <= 0.60
        if (processInv[14,hour] > 0) && notBusy6
            nProcess = min(speed[6], processInv[14,hour])
            InvStations[1,hour:end] += nProcess #* 0.8
            InvStations[2,hour:end] -= nProcess

            processInv[15,hour:end] += nProcess #* 0.8
            processInv[14, hour:end] -= nProcess

            nMacUse[6] += 1
            if nMacUse[6] == maxMachine[6]
                notBusy6 = false
            end
        end
        if (processInv[12,hour] > 0) && notBusy6
            nProcess = min(speed[6], processInv[12,hour])
            InvStations[2,hour:end] += nProcess# * 0.8
            InvStations[2,hour:end] -= nProcess

            processInv[13,hour:end] += nProcess# * 0.8
            processInv[12, hour:end] -= nProcess

            nMacUse[6] += 1
            if nMacUse[6] == maxMachine[6]
                notBusy6 = false
            end
        end
        if (processInv[4,hour] > 0) && notBusy6
            nProcess = min(speed[6], processInv[4,hour])
            InvStations[1,hour:end] += nProcess# * 0.8
            InvStations[2,hour:end] -= nProcess

            processInv[5,hour:end] += nProcess# * 0.8
            processInv[4,hour:end] -= nProcess

            nMacUse[6] += 1
            if nMacUse[6] == maxMachine[6]
                notBusy6 = false
            end
        end
    else
```

```
            nMacUse[6] += 1
            if nMacUse[6] == maxMachine[6]
                notBusy6 = false
            end
        end
    end
        # At station 7
    while notBusy7
        if rand() <= 0.70
            if (processInv[25,hour] > 0) && notBusy7
                nProcess = min(speed[7], processInv[25,hour:end])
                InvStations[2,hour:end] += nProcess #* 0.8
                InvStations[4,hour:end] -= nProcess

                processInv[26,hour:end] += nProcess #* 0.8
                processInv[25, hour:end] -= nProcess

                nMacUse[7] += 1
                if nMacUse[7] == maxMachine[7]
                    notBusy7 = false
                end
            end
            if (processInv[19,hour] > 0) && notBusy7
                nProcess = min(speed[7], processInv[19,hour])
                InvStations[2,hour:end] += nProcess #* 0.8
                InvStations[4,hour:end] -= nProcess

                processInv[20,hour:end] += nProcess #* 0.8
                processInv[19, hour:end] -= nProcess

                nMacUse[7] += 1
                if nMacUse[7] == maxMachine[7]
                    notBusy7 = false
                end
            end

        else
            nMacUse[7] += 1
            if nMacUse[7] == maxMachine[7]
                notBusy7 = false
            end
        end
    end
    # At station 8
    while notBusy8
        if rand() <= 0.85
            if (processInv[28,hour] > 0) && notBusy8
                nProcess = min(speed[8], processInv[28,hour])
                InvStations[2,hour:end] += nProcess #* 0.8
                InvStations[4,hour:end] -= nProcess

                processInv[29,hour:end] += nProcess #* 0.8
                processInv[28, hour:end] -= nProcess
```

```
                                nMacUse[8] += 1
                                if nMacUse[8] == maxMachine[8]
                                    notBusy8 = false
                                end
                            end
                        else
                            nMacUse[8] += 1
                            if nMacUse[8] == maxMachine[8]
                                notBusy8 = false
                            end
                        end
                    end

                end
            end
            return finishedInv
        end
```

```
Out[253]: productACapacity (generic function with 2 methods)
```

```
In [278]: n = 100
          hourW = round(Int64,((21 * 7) - 20) * 0.85)
          machineSetup = [1 2 1 1 1 1 1 1]
          capacity = zeros(n,1)
          for i=1:n
              temp = productACapacity(machineSetup)
              capacity[i] += temp[end]
          end

          mean(capacity)
```

```
Out[278]: 193.69400000000002
```

```
In [282]: n = 100
          hourW = round(Int64,((21 * 7) - 20) * 0.85)
          machineSetup = [1 2 1 2 1 1 1 1]
          capacity = zeros(n,1)
          for i=1:n
              temp = productACapacity(machineSetup)
              capacity[i] += temp[end]
          end

          mean(capacity)
```

```
Out[282]: 250.04700000000014
```

```
In [291]: n = 100
          hourW = round(Int64,((21 * 7) - 20) * 0.85)
          machineSetup = [1 3 2 2 1 1 1 1]
          capacity = zeros(n,1)
          for i=1:n
              temp = productACapacity(machineSetup)
              capacity[i] += temp[end]
          end

          mean(capacity)
```

```
Out[291]: 390.3

In [300]: n = 100
          hourW = round(Int64,((21 * 7) - 20) * 0.85)
          machineSetup = [2 3 2 2 1 1 1 1]
          capacity = zeros(n,1)
          for i=1:n
              temp = productACapacity(machineSetup)
              capacity[i] += temp[end]
          end

          mean(capacity)

Out[300]: 431.9340000000001

In [309]: n = 100
          hourW = round(Int64,((21 * 7) - 20) * 0.85)
          machineSetup = [2 4 2 2 1 1 1 1]
          capacity = zeros(n,1)
          for i=1:n
              temp = productACapacity(machineSetup)
              capacity[i] += temp[end]
          end

          mean(capacity)

Out[309]: 449.728

In [318]: n = 100
          hourW = round(Int64,((21 * 7) - 20) * 0.85)
          machineSetup = [2 4 2 3 1 1 1 1]
          capacity = zeros(n,1)
          for i=1:n
              temp = productACapacity(machineSetup)
              capacity[i] += temp[end]
          end

          mean(capacity)

Out[318]: 619.723

In [327]: n = 100
          hourW = round(Int64,((21 * 7) - 20) * 0.85)
          machineSetup = [2 4 3 4 1 1 1 1]
          capacity = zeros(n,1)
          for i=1:n
              temp = productACapacity(machineSetup)
              capacity[i] += temp[end]
          end

          mean(capacity)

Out[327]: 639.4259999999999

In [335]: n = 100
          hourW = round(Int64,((21 * 7) - 20) * 0.85)
```

```
machineSetup = [2 5 3 4 1 1 1 1]
capacity = zeros(n,1)
for i=1:n
    temp = productACapacity(machineSetup)
    capacity[i] += temp[end]
end

mean(capacity)
```

```
Out[335]: 637.533
```

```
In [344]: n = 100
hourW = round(Int64,((21 * 7) - 20) * 0.85)
machineSetup = [4 5 3 4 2 1 1 1]
capacity = zeros(n,1)
for i=1:n
    temp = productACapacity(machineSetup)
    capacity[i] += temp[end]
end

mean(capacity)
```

```
Out[344]: 785.6830000000002
```

```
In [354]: n = 100
hourW = round(Int64,((21 * 7) - 20) * 0.85)
machineSetup = [4 5 3 4 2 2 1 1]
capacity = zeros(n,1)
for i=1:n
    temp = productACapacity(machineSetup)
    capacity[i] += temp[end]
end

mean(capacity)
```

```
Out[354]: 845.1729999999995
```

```
In [371]: n = 100
hourW = round(Int64,((21 * 7) - 20) * 0.85)
machineSetup = [4 6 3 4 2 2 1 1]
capacity = zeros(n,1)
for i=1:n
    temp = productACapacity(machineSetup)
    capacity[i] += temp[end]
end

@show mean(capacity)
@show std(capacity)
;
```

```
mean(capacity) = 1011.4280000000003
std(capacity) = 45.631466371708065
```

```
In [374]: n = 100
hourW = round(Int64,((21 * 7) - 20) * 0.85)
machineSetup = [4 6 3 5 2 2 1 1]
```

```
capacity = zeros(n,1)
for i=1:n
    temp = productACapacity(machineSetup)
    capacity[i] += temp[end]
end

mean(capacity)
```

Out[374]: 1068.2430000000004

In [ ]: