

Data 604 final

July 19, 2020

1 Data 604 Final Project

1.1 Cricket Bat Factory? (Simulation)

1.1.1 V Patel

1.2 Objective

There is a company who recieved a big order (100 bats) to deliver Cricket Bats in one month. I will create a simulation to find out whether company can delivered it on time.

```
[1]: # Configure Jupyter so figures appear in the notebook
      %matplotlib inline

      # Configure Jupyter to display the assigned value after an assignment
      %config InteractiveShell.ast_node_interactivity='last_expr_or_assign'

      # import functions from the modsim.py module
      from modsim import *
```

2 Process

2.0.1 Parameters

```
[2]: import simpy
      import random

      bats_made = 0
      #-----

      #Parameters

      #working hours
      hours = 8

      #business days
      days = 23

      #total working time (hours)
```

```

total_time = hours * days

#containers
    #wood
wood_capacity = 500
initial_wood = 200

    #glue
supply_capacity = 500
initial_supply = 200

    #dispatch
dispatch_capacity = 100

#employees per activity

#body_claft
num_body = 1
mean_body = 3
std_body = 1

#handle
num_neck = 1
mean_neck = 1
std_neck = 0.2

#Fitting Handle
num_ensam = 1
mean_ensam = 1
std_ensam = 0.2

#Shaping the blade
num_shapers = 1
mean_shapers = 3
std_shapers = 0.2

#Binding
num_binding = 1
mean_binding = 1
std_binding = .1

#Polishing
num_polish = 1
mean_polish = 1

```

```

std_polish = 0.1

#final touches adding grip,stickers, oils
num_final = 1
mean_final = 1
std_final = 0.1

```

[2]: 0.1

[]:

```

[3]: #-----
#critical levels
#critical stock should be 1 business day greater than supplier take to come

class Bat_Factory:
    def __init__(self, env):
        self.wood = simpy.Container(env, capacity = wood_capacity, init = ↵
↵initial_wood)
        self.wood_control = env.process(self.wood_stock_control(env))
        self.supply = simpy.Container(env, capacity = supply_capacity, init = ↵
↵initial_supply)
        self.supply_control = env.process(self.supply_stock_control(env))
        self.dispatch = simpy.Container(env ,capacity = dispatch_capacity, init ↵
↵= 0)
        self.dispatch_control = env.process(self.dispatch_bats_control(env))

    def wood_stock_control(self, env):
        yield env.timeout(0)
        while True:
            if self.wood.level <= 100:
                print('cleft stock bellow critical level ({0}) at day {1}, hour ↵
↵{2}'.format(
                    self.wood.level, int(env.now/8), env.now % 8))

                yield env.timeout(5)
                print('cleft supplier arrives at day {0}, hour {1}'.format(
                    int(env.now/8), env.now % 8))
                yield self.wood.put(300)
                print('new wood stock is {0}'.format(
                    self.wood.level))
                print('-----')
                yield env.timeout(5)
            else:
                yield env.timeout(1)

```

```

def supply_stock_control(self, env):
    yield env.timeout(0)
    while True:
        if self.supply.level <= 50:
            print('supply stock bellow critical level ({0}) at day {1},_
→hour {2}'.format(
                self.supply.level, int(env.now/8), env.now % 8))
            print('Ordered supply/shineing supplier')
            print('-----')
            yield env.timeout(9)
            print('supply supplier arrives at day {0}, hour {1}'.format(
                int(env.now/8), env.now % 8))
            yield self.supply.put(30)
            print('new supply stock is {0}'.format(
                self.supply.level))
            print('-----')
            yield env.timeout(8)
        else:
            yield env.timeout(1)

def dispatch_bats_control(self, env):
    global bats_made
    yield env.timeout(0)
    while True:
        if self.dispatch.level >= 1:
            print('dispatch stock is {0}, calling store to pick bats at day_
→{1}, hour {2}'.format(
                self.dispatch.level, int(env.now/8), env.now % 8))
            print('-----')
            yield env.timeout(4)
            print('store picking {0} bat at day {1}, hour {2}'.format(
                self.dispatch.level, int(env.now/8), env.now % 8))
            bats_made += self.dispatch.level
            yield self.dispatch.get(self.dispatch.level)
            print('-----')
            yield env.timeout(8)
        else:
            yield env.timeout(1)

## work

def cleft_maker(env, bat_factory):
    while True:
        yield bat_factory.wood.get(1)
        body_time = random.gauss(mean_body, std_body)
        yield env.timeout(body_time)

```

```

def handle_maker(env, bat_factory):
    while True:
        yield bat_factory.wood.get(1)
        neck_time = random.gauss(mean_neck, std_neck)
        yield env.timeout(neck_time)

def assembler(env, bat_factory):
    while True:
        yield bat_factory.supply.get(1)
        assembling_time = random.gauss(mean_ensam, std_ensam)
        yield env.timeout(assembling_time)

def shape_blade(env, bat_factory):
    while True:
        yield bat_factory.supply.get(1)
        shape_time = random.gauss(mean_shapers, std_shapers)
        yield env.timeout(shape_time)

def binding(env, bat_factory):
    while True:
        yield bat_factory.supply.get(1)
        binding_time = random.gauss(mean_binding, std_binding)
        yield env.timeout(binding_time)

def polish(env, bat_factory):
    while True:
        yield bat_factory.supply.get(1)
        polish_time = random.gauss(mean_polish, std_polish)
        yield env.timeout(polish_time)

def final(env, bat_factory):
    while True:
        yield bat_factory.supply.get(1)
        final_time = random.gauss(mean_final, std_final)
        yield env.timeout(final_time)
        yield bat_factory.dispatch.put(1)

#Generators

def cleft_maker_gen(env, bat_factory):
    for i in range(num_body):
        env.process(cleft_maker(env, bat_factory))
        yield env.timeout(1)

def handle_maker_gen(env, bat_factory):

```

```

    for i in range(num_neck):
        env.process(handle_maker(env, bat_factory))
        yield env.timeout(1)

def assembler_maker_gen(env, bat_factory):
    for i in range(num_ensam):
        env.process(assembler(env, bat_factory))
        yield env.timeout(1)

def shape_blade_gen(env, bat_factory):
    for i in range(num_shapers):
        env.process(shape_blade(env, bat_factory))
        yield env.timeout(1)

def binding_gen(env, bat_factory):
    for i in range(num_binding):
        env.process(shape_blade(env, bat_factory))
        yield env.timeout(1)

def polish_gen(env, bat_factory):
    for i in range(num_polish):
        env.process(polish(env, bat_factory))
        yield env.timeout(1)

def final_gen(env, bat_factory):
    for i in range(num_final):
        env.process(final(env, bat_factory))
        yield env.timeout(1)

#-----

```

```

[4]: env = simpy.Environment()
bat_factory = Bat_Factory(env)

print(f'STARTING SIMULATION')
print(f'-----')

body_gen = env.process(cleft_maker_gen(env, bat_factory))
neck_gen = env.process(handle_maker_gen(env, bat_factory))
assembler_gen = env.process(assembler_maker_gen(env, bat_factory))
shape_blade_gen = env.process(shape_blade_gen(env, bat_factory))
binding_gen = env.process(binding_gen(env, bat_factory))
polish_gen = env.process(polish_gen(env, bat_factory))
final_gen = env.process(final_gen(env, bat_factory))

```

```

env.run(until = total_time)

print(f'Dispatch has %d bats ready to go!' % bat_factory.dispatch.level)
print(f'-----')
print(f'total bats made: {0}'.format(bats_made + bat_factory.dispatch.level))
print(f'-----')
print(f'SIMULATION COMPLETED')

```

STARTING SIMULATION

```

-----
dispatch stock is 1, calling store to pick bats at day 0, hour 2
-----
store picking 5 bat at day 0, hour 6
-----
dispatch stock is 8, calling store to pick bats at day 1, hour 6
-----
store picking 12 bat at day 2, hour 2
-----
dispatch stock is 8, calling store to pick bats at day 3, hour 2
-----
store picking 11 bat at day 3, hour 6
-----
dispatch stock is 8, calling store to pick bats at day 4, hour 6
-----
supply stock bellow critical level (48) at day 5, hour 1
Ordered supply/shineing supplier
-----
store picking 12 bat at day 5, hour 2
-----
supply supplier arrives at day 6, hour 2
dispatch stock is 8, calling store to pick bats at day 6, hour 2
-----
new supply stock is 44
-----
store picking 12 bat at day 6, hour 6
-----
supply stock bellow critical level (16) at day 7, hour 2
Ordered supply/shineing supplier
-----
dispatch stock is 8, calling store to pick bats at day 7, hour 6
-----
store picking 10 bat at day 8, hour 2
-----
supply supplier arrives at day 8, hour 3
new supply stock is 25
-----

```

```

dispach stock is 7, calling store to pick bats at day 9, hour 2
-----
cleft stock bellow critical level (100) at day 9, hour 2
supply stock bellow critical level (0) at day 9, hour 3
Ordered supply/shineing supplier
-----
store picking 8 bat at day 9, hour 6
-----
cleft supplier arrives at day 9, hour 7
new wood stock is 394
-----
supply supplier arrives at day 10, hour 4
new supply stock is 25
-----
dispach stock is 1, calling store to pick bats at day 10, hour 6
-----
store picking 5 bat at day 11, hour 2
-----
supply stock bellow critical level (0) at day 11, hour 4
Ordered supply/shineing supplier
-----
dispach stock is 2, calling store to pick bats at day 12, hour 2
-----
supply supplier arrives at day 12, hour 5
new supply stock is 25
-----
store picking 2 bat at day 12, hour 6
-----
supply stock bellow critical level (0) at day 13, hour 5
Ordered supply/shineing supplier
-----
dispach stock is 8, calling store to pick bats at day 13, hour 6
-----
store picking 8 bat at day 14, hour 2
-----
supply supplier arrives at day 14, hour 6
new supply stock is 25
-----
dispach stock is 3, calling store to pick bats at day 15, hour 2
-----
supply stock bellow critical level (0) at day 15, hour 6
Ordered supply/shineing supplier
-----
store picking 8 bat at day 15, hour 6
-----
supply supplier arrives at day 16, hour 7
new supply stock is 25
-----

```



```

dispach stock is 2, calling store to pick bats at day 17, hour 1
-----
store picking 5 bat at day 17, hour 5
-----
supply stock bellow critical level (1) at day 17, hour 7
Ordered supply/shineing supplier
-----
dispach stock is 3, calling store to pick bats at day 18, hour 5
-----
supply supplier arrives at day 19, hour 0
new supply stock is 25
-----
store picking 3 bat at day 19, hour 1
-----
supply stock bellow critical level (0) at day 20, hour 0
Ordered supply/shineing supplier
-----
dispach stock is 8, calling store to pick bats at day 20, hour 1
-----
store picking 8 bat at day 20, hour 5
-----
supply supplier arrives at day 21, hour 1
new supply stock is 25
-----
dispach stock is 3, calling store to pick bats at day 21, hour 5
-----
supply stock bellow critical level (0) at day 22, hour 1
Ordered supply/shineing supplier
-----
store picking 7 bat at day 22, hour 1
-----
Dispatch has 1 bats ready to go!
-----
total bats made: 117
-----
SIMULATION COMPLETED

```

2.1 Conclusion:

From the simulation, We can conclude that this small factory can able to deliveverd the order on time.

[]:

2.2 Reference

1. <https://www.woodstockcricket.co.uk/about/the-bat-making-process>