KY2: The Three Bears Problem

Kieran Molloy

January 2020

Abstract

Investigate Three Bears Porridge production plans to minimise costs. Looking at current stock levels and using MILP to calculate an optimal plan, then calculating optimal initial/final stock levels for the 12 month scope.

1 Introduction

The Three Bears Porridge Company makes canned porridge for the home breakfast market. Three flavours of porridge are made: Natural, Honey and Fruit. The sale of canned porridge is highly seasonal, an optimal production plan needs to be created. It is a multi-item production planning problem, with the additional complexity of wage constraints. function.*KY2 Specification* 2019

2 Analysis

2.1 Details of the Problem

The direct cost of manufacturing is £11.50 per tonne. Canned porridge can be stored almost indefinitely at a cost of £0.22 per tonne per month. Each tonne of product made requires 1.8 man hours of supervision time on the machinery. Regular time costs £10.00 per man-hour. Overtime is available in two bands. The first band (evenings) corresponds to a maximum of 10% of available regular time and each man-hour in this band costs £13.00. The second band of overtime (weekends) corresponds to a maximum of 20% of available regular time and costs £16.00 per man hour. There are 3,200 man-hours of regular time available in each month.*KY2 Specification* 2019

Month	1	2	3	4	5	6	7	8	9	10	11	12
Natural	1900	1800	1300	900	600	400	400	500	800	1200	1500	1700
Honey	700	600	600	500	350	200	200	300	500	600	800	900
Fruit	300	300	300	250	250	200	200	200	250	300	300	40

Table 1: Porridge Demand per Type & Month

2.2 Formulation

2.3 Single-Item Planning Period

Since this is a production planning problem we can base the model on that, T is the set of time periods and D_t is the demand per time period. Such can be modelled r and v, these are production and storage costs respectively. Usually there is also S_I, this is the initial stock. The decision variables required are x_t and s_t, these are produced and stored for each time period. This basic model is as follows:

$$\begin{aligned} \underset{X}{\text{minimise}} \quad & \sum_{t=1}^{T} rx_i + vs_t \\ \text{subject to} \quad & s_t = S_I + x_t - D_t,\ t = 1, \\ & s_t = s_{t-1} + x_t - D_t,\ t \neq 1, \\ & x_t, s_t \succeq 0,\ t \in T. \end{aligned} \tag{1}$$

2.4 Multi-Item Planning Period

In a multi-item problem, the previous notation 1 is extended to include P the set of products. This changes the notation to:

Variable	Description
$D_{t,p}$	Demand for time period t and product p
v_p	Cost of Production of product p
w_p	Cost of Storage of product p
S_p	Initial Stock Level of product p
$x_{t,p}$	Number of product p produced in time period t
$s_{t,p}$	Number of product p stored in time period t

Table 2: Multi-Item Notation

$$\begin{aligned} \underset{X}{\text{minimise}} \quad & \sum_{t=1}^{T} \sum_{p=1}^{P} r_p x_{t,p} + v_p s_{t,p} \\ \text{subject to} \quad & s_{t,p} = S_p + x_{t,p} - D_{t,p}, \qquad t = 1, \forall p \in P, \\ & s_{t,p} = s_{t-1,p} + x_{t,p} - D_{t,p}, \quad t \neq 1, \forall p \in P, \\ & x_{t,p}, s_{t,p} \succeq 0,\ t \in T, p \in P. \end{aligned} \tag{2}$$

2.5 Three Bears Problem

The Three Bears problem is a multi-item production planning problem with the additional complexity of wage bands and overtime. The following are modified versions of 2

Variable	Description
T	Set of Time Periods
P	Set of Products
B	Set of Wage Bands
$D_{t,p}$	Demand for time period t and product p
r	Cost of Production
v	Cost of Storage
i_p	Initial Stock Level of product p
f_p	Final Stock Level of product p
g	Man Time required for production
c_b	Max Capacity of wage band b
z_b	Wage per hour of wage band b
$x_{t,p}$	Number of product p produced in time period t
$s_{t,p}$	Number of product p stored in time period t
$w_{t,b}$	Hours worked in band b in time period t

Table 3: Three Bears Notation

$$
\begin{aligned}
\underset{X}{\text{minimise}} \quad & \sum_{t=1}^{T}\left[\sum_{p=1}^{P} r x_{t,p} + v s_{t,p} + \sum_{b=1}^{B} z_b w_{t,b}\right] \\
\text{subject to} \quad & s_{t,p} = S_p + x_{t,p} - D_{t,p}, && t = 1, \forall p \in P, \\
& s_{t,p} = s_{t-1,p} + x_{t,p} - D_{t,p}, && t \neq 1, \forall p \in P, \\
& s_{t,p} \geq f_p, && \forall t \in T, \forall p \in P, \\
& \sum_{p=1}^{P} g x_{t,p} = \sum_{b=1}^{B} w_{t,b}, && \forall t \in T, \\
& w_{t,b} \leq c_b, && \forall t \in T, \forall b \in B, \\
& x_{t,p}, s_{t,p}, w_{t,b} \succeq 0, && t \in T, p \in P, b \in B.
\end{aligned} \tag{3}
$$

This model was then created in XPress Mosel 64-bit *XPress* 2019

3 Results

3.1 Part 1 - 12 Month Strategy

The code to make this section run is in Appendix A.

Running the model gives:

Total cost: 674954.86

Production plan:

	Month	1	2	3	4	5	6	7	8	9	10	11	12
1	Prod	1934	1746	1300	1028	472	401	1778	1778	0	0	864	1699
	Store	(54)	(0)	(0)	(128)	(0)	(1)	(1379)	(2657)	(1857)	(657)	(21)	(20)
2	Prod	0	0	350	500	350	933	0	0	1905	1955	0	257
	Store	(850)	(250)	(0)	(0)	(0)	(733)	(533)	(233)	(1638)	(2993)	(2193)	(1550)
3	Prod	0	210	300	250	956	444	0	0	0	0	1090	0
	Store	(90)	(0)	(0)	(0)	(706)	(950)	(750)	(550)	(300)	(0)	(790)	(390)

Table 4: Optimal Production Plan for part 1

Capacities used:

Month	Total	Standard	OT Band 1	OT Band 2
1	3481.2	3200	281.2	0
2	3520.8	3200	320	0.8
3	3510	3200	310	0
4	3200.4	3200	0.4	0
5	3200.4	3200	0.4	0
6	3200.4	3200	0.4	0
7	3200.4	3200	0.4	0
8	3200.4	3200	0.4	0
9	3429	3200	229	0
10	3519	3200	319	0
11	3517.2	3200	317.2	0
12	3520.8	3200	320	0.8
Total	40500	38400	2098.4	1.6

Table 5: Optimal Capacity Plan for part 1

I decided to make it a requirement that the model is cyclic, and could be performed year on year. This is largely the purpose of part 2, but the difference is optimising the actual initial/final stock levels. It can be seen that the total cost of the plan is £674954.86, this allows Three Bears to meet all demands required, and also allows for the minimal wage & storage costs. Every month the entire standard wage band is utilised, even where the month directly didn't require it. Band 1 is either near fully used or not used at all, and very rarely is band 2 used. This reflects the stockpiling preferences of the model. Without the wage band constraint, the production plan would consist of storing no stock and would be unrepresentative of the real world. With its inclusion, the stock is utilised to keep wage costs low as it is far cheaper to store large quantities. From Month 7 the plan is to start stockpiling for the new years plan, since it is expensive to do all the producing at once, it it done gradually in stages; months 7 & 8 are used for Natural, months 9 & 10 for Honey and months 11 & 12 for Fruit. At one point there is a stockpile of 2657 of Natural Porridge in Month 8, this is to allow no production of it for the next 2 months and still meet demand, and again with Honey in Month 10 (2993), to allow no production in 11 & 12.

3.2 Part 2 - Rolling 12 Month Strategy

The code to make this section run is in Appendix B.

Running the model gives:

Total cost: 674926.3

Production plan:

	Month	1	2	3	4	5	6	7	8	9	10	11	12
1	Prod	1190	1955	1145	900	600	400	1700	0	0	1200	1955	1955
	Store	(0)	(155)	(0)	(0)	(0)	(0)	(1300)	(800)	(0)	(0)	(455)	(710)
2	Prod	75	0	600	628	222	400	0	1690	1880	755	0	0
	Store	(600)	(0)	(0)	(128)	(0)	(200)	(0)	(1390)	(2770)	(2925)	(2125)	(1225)
3	Prod	690	0	210	250	956	978	78	88	0	0	0	0
	Store	(390)	(90)	(0)	(0)	(706)	(1484)	(1362)	(1250)	(1000)	(700)	(400)	(0)

Table 6: Optimal Production Plan for part 2

Capacities used:

Month	Total	Standard	OT Band 1	OT Band 2
1	3519	3200	319	0
2	3519	3200	319	0
3	3519	3200	319	0
4	3200.4	3200	0.4	0
5	3200.4	3200	0.4	0
6	3200.4	3200	0.4	0
7	3200.4	3200	0.4	0
8	3200.4	3200	0.4	0
9	3384	3200	184	0
10	3519	3200	319	0
11	3519	3200	319	0
12	3519	3200	319	0
Total	40500	38400	2100	0

Table 7: Optimal Capacity Plan for part 2

This model is fully cyclic and repeatable, the model gives the Initial/Final stock levels of

$$i_1 = f_1 = 710, i_2 = f_2 = 1225, i_3 = f_3 = 0$$

to minimise costs the greatest. The total cost is £674926.3, only £29 less than the production plan in Part 13.1, this cost is from higher storage costs since the total hours stay constant (due to the fact that time to create 1 tonne of product maintains the same regardless of other data, therefore to meet demand never changes). If the year could be ended with no stock, the total cost could be £611041.16, however, the costs incurred in the first 3 months of the following period would be far higher than the difference. In the rolling 12 month plan, there is no Band 2 work, this is due to the extreme wage cost, which can be avoided with the better stock starting/finishing levels. It can also be noted that the stockpiling trend noted in Table 4 is also seen in Table 6. The differences are the seemingly sporadic approach to it, Natural has a spike in months 6 & 7 and produced none in 8 & 9, then in month 10 it is the largest and in 11 & 12, it is the only porridge produced. Honey has a peak of 2925 in month 10, which is a build of the 3 previous months (8,9,10) and then is not produced until the new cycle, and relies on the stored porridge for the 1225 final level as well as meeting demand. Fruit porridge sees stock prepared from month 5, relying purely on stored porridge for months (9,10,11,12).]

3.3 Part 3 - Next 24 Months

It is known that the most optimal rolling 12 month plan is £674926.30, as documented in Part 3.2. Three Bears need the most optimal way to convert their current stock levels to the rolling 12 month strategy, and that is as follows.

	Month	1	2	3	4	5	6	7	8	9	10	11	12
1	Prod	1880	1800	1300	1028	472	400	900	0	1880	1955	1483	592
	Store	(0)	(0)	(0)	(128)	(0)	(0)	(500)	(0)	(1080)	(1835)	(1818)	(710)
2	Prod	0	0	350	500	1056	0	878	1778	0	0	0	1363
	Store	(850)	(250)	(0)	(0)	(706)	(506)	(1184)	(2662)	(2162)	(1562)	(762)	(1225)
3	Prod	55	155	300	250	250	1378	0	0	0	0	472	0
	Store	(145)	(0)	(0)	(0)	(0)	(1178)	(978)	(778)	(528)	(228)	(400)	(0)
	Month	13	14	15	16	17	18	19	20	21	22	23	24
1	Prod	1190	1955	1145	900	600	400	1700	0	0	1200	1955	1955
	Store	(0)	(155)	(0)	(0)	(0)	(0)	(1300)	(800)	(0)	(0)	(455)	(710)
2	Prod	75	0	600	628	222	400	0	1690	1880	755	0	0
	Store	(600)	(0)	(0)	(128)	(0)	(200)	(0)	(1390)	(2770)	(2925)	(2125)	(1225)
3	Prod	690	0	210	250	956	978	78	88	0	0	0	0
	Store	(390)	(90)	(0)	(0)	(706)	(1484)	(1362)	(1250)	(1000)	(700)	(400)	(0)

Table 8: Optimal Production Plan for next 24 Months

The full cost of this plan is £1348982.3, the first 12 months total cost is £674056, the second 12 month period total cost is £674926.30. This production plan allows Three Bears to slowly, and cost-effectively transition onto the optimal production plan for their demand over a 24 month period.

4 Conclusion

Three Bears have a good insight into how they should tailor their production plan to minimise costs, and therefore maximise profit. Three Bears should follow the production plan outlined in Part 3, then once this 24 month period has finished they should transition onto the production plan outlines in Part 2. All initial stock levels & final stock levels have been accounted for, including current stock.

5 Discussion

It would be interesting to additionally model machine constraints, but that is for another day :) It could also be interesting to investigate the potential of modelling the weeks in the months that are busy for Three Bears.

References

KY2 Specification (2019). Keith Yates. URL: https://moodle.mmu.ac.uk/pluginfile.php/3790597/mod_assign/introattachment/0/KY2.pdf (visited on 12/03/2019).

XPress (2019). FICO. URL: https://www.fico.com/en/products/fico-xpress-optimization (visited on 12/03/2019).

A Mosel Code - Part 1

```

model "KY2 Three Bears − Part 1"
 uses "mmxprs"

 declarations
  NT = 12                                    ! Number of Months in planning period
  MONTHS = 1..NT                             ! Set of Months
  P = 3                                      ! Number of products in plan
  PRODS = 1.. P                              ! Set of products
  W = 3                                      ! Number of wage bands in plan
  BANDS = 1..W                               ! Set of Wage Bands



  DEM: array(PRODS,MONTHS) of integer        ! Demand per product and per month
  CPROD: real                                ! Production cost per product
  CSTOCK: real                               ! Storage cost per product
  ISTOCK: array(PRODS) of integer            ! Initial stock levels
  FSTOCK: array(PRODS) of integer            ! Min. final stock levels
  TIMEC: real                                ! Worker time per unit
  CAPW: array(BANDS) of integer              ! Capacity of workers
  WAGE: array(BANDS) of real                 ! Wage for workers

   produce: array(PRODS,MONTHS) of mpvar ! Production of products per month
   store: array(PRODS,MONTHS) of mpvar   ! Amount stored at end of month
   working: array(MONTHS,BANDS) of mpvar ! Number of Hours for each band each month
   hours: array(MONTHS) of mpvar         ! Total Number of Hours
 end−declarations

 initializations from 'c2glass.dat'
  CAPW DEM CSTOCK CPROD ISTOCK FSTOCK TIMEC WAGE
 end−initializations

! Objective: sum of production and storage costs
 Cost:=
  sum(p in PRODS, t in MONTHS) (CPROD∗produce(p,t) + CSTOCK∗store(p,t)) + sum(t in
     MONTHS,b in BANDS) working(t,b)∗WAGE(b)

! Stock balances
 forall(p in PRODS, t in MONTHS)
   store(p,t) = if(t>1, store(p,t−1), ISTOCK(p)) + produce(p,t) − DEM(p,t)

! Final stock levels
```

```
forall(p in PRODS) store(p,NT) >= FSTOCK(p)

! Working Time
forall(t in MONTHS)
 sum(p in PRODS) TIMEC*produce(p,t) = hours(t)     ! Total Work Hours

! Converting Monthly Time to Wage Bands
forall(t in MONTHS)
 sum(b in BANDS) working(t,b) = hours(t)

! Capacity constraint
forall(t in MONTHS, b in BANDS)
 working(t,b) <= CAPW(b)


 ! Force Integer
forall(p in PRODS, t in MONTHS) do
 produce(p,t) is_integer
 store(p,t) is_integer
end-do

 ! Force Non-Negativity
forall(p in PRODS, t in MONTHS) do
 produce(p,t) >=0
 store(p,t) >= 0
end-do

! Solve the problem
minimize(Cost)

! Solution printing
writeln("Total cost: ",getobjval)
writeln("Production plan:")
write("      Week")
forall(t in MONTHS) write(strfmt(t,7))
writeln
forall(p in PRODS) do
 write(p,": Prod. ")
 forall(t in MONTHS) write(strfmt(getsol(produce(p,t)),7))
 writeln
 write("    Store ")
 forall(t in MONTHS) write(strfmt("("+getsol(store(p,t))+")",7))
 writeln

```

```
 end-do

 writeln("\nCapacities used:")
 writeln("Week",strfmt("Total",6),strfmt("Standard",10),strfmt("OT Band 1",10),strfmt("OT Band 2",10))
 forall(t in MONTHS)
  writeln(strfmt(t,3),
   strfmt(getsol(hours(t)),8),
   strfmt(getsol(working(t,1)),8),
   strfmt(getsol(working(t,2)),10),
   strfmt(getsol(working(t,3)),10))

end-model
```

A Mosel Code - Part 2

```
model "KY2 Three Bears - Part 2"
uses "mmxprs"

 declarations
  NT = 12                              ! Number of Months in planning period
  MONTHS = 1..NT                       ! Set of Months
  P = 3                                ! Number of products in plan
  PRODS = 1.. P                        ! Set of products
  W = 3                                ! Number of wage bands in plan
  BANDS = 1..W                         ! Set of Wage Bands


  DEM: array(PRODS,MONTHS) of integer   ! Demand per product and per month
  CPROD: real        ! Production cost per product
  CSTOCK: real        ! Storage cost per product
  ISTOCK: array(PRODS) of integer       ! Initial stock levels
  FSTOCK: array(PRODS) of integer       ! Min. final stock levels
  TIMEC: real ! Worker time per unit
  CAPW: array(BANDS) of integer         ! Capacity of workers
  WAGE: array(BANDS) of real ! Wage for workers


   produce: array(PRODS,MONTHS) of mpvar ! Production of products per month
   store: array(PRODS,MONTHS) of mpvar   ! Amount stored at end of month
   working: array(MONTHS,BANDS) of mpvar ! Number of Hours for each band each month
   hours: array(MONTHS) of mpvar ! Total Number of Hours
   ifstock: array(PRODS) of mpvar ! Initial/Final Stock Levels
 end-declarations

 initializations from 'KY2.2.dat'
  CAPW DEM CSTOCK CPROD ISTOCK FSTOCK TIMEC WAGE
 end-initializations

! Objective: sum of production and storage costs
 Cost:=
  sum(p in PRODS, t in MONTHS) (CPROD*produce(p,t) + CSTOCK*store(p,t)) + sum(t in
    MONTHS,b in BANDS) WAGE(b)*working(t,b)

! Stock balances
 forall(p in PRODS, t in MONTHS)
   store(p,t) = if(t>1, store(p,t-1), ifstock(p)) + produce(p,t) - DEM(p,t)

! Intial/Final stock levels
```

```
 forall(p in PRODS) do
  store(p,NT) = ifstock(p)
 end-do



! Working Time
 forall(t in MONTHS)
  sum(p in PRODS) TIMEC*produce(p,t) = hours(t)      ! Total Work Hours

! Converting Monthly Time to Wage Bands
 forall(t in MONTHS)
  sum(b in BANDS) working(t,b) = hours(t)

! Capacity constraint
 forall(t in MONTHS, b in BANDS)
  working(t,b) <= CAPW(b)


 ! Force Integer
 forall(p in PRODS, t in MONTHS) do
  produce(p,t) is_integer
  store(p,t) is_integer
 end-do

 ! Force Non-Negativity
 forall(p in PRODS, t in MONTHS) do
  produce(p,t) >=0
  store(p,t) >= 0
 end-do

! Solve the problem
 minimize(Cost)

! Solution printing
 writeln("Total cost: ",getobjval)
 writeln("Production plan:")
 write("      Week")
 forall(t in MONTHS) write(strfmt(t,7))
 writeln
 forall(p in PRODS) do
  write(p,": Prod. ")
  forall(t in MONTHS) write(strfmt(getsol(produce(p,t)),7))
  writeln
```

```
  write("    Store ")
  forall(t in MONTHS) write(strfmt("("+getsol(store(p,t))+")",7))
  writeln

 end-do

 writeln("\nCapacities used:")
 writeln("Week",strfmt("Total",6),strfmt("Standard",10),strfmt("OT Band 1",10),strfmt("
   OT Band 2",10))
 forall(t in MONTHS)
  writeln(strfmt(t,3),
   strfmt(getsol(hours(t)),8),
   strfmt(getsol(working(t,1)),8),
   strfmt(getsol(working(t,2)),10),
   strfmt(getsol(working(t,3)),10))

end-model
```