Optimization Report: An orange juice production Plan

Production Planning

In an analysis of the UniCitrus production planning in order to maximize profit from producing orange juices, Standard and Dairy, there are various conditions that not only effects to the amount of production, but it also needs to concern varieties between the market demand and supply as well as a capacity of owned plant. Consequently, in the first part of this report will focus on a fundamental condition which returns an optimal solution in a case of an infinite machine capability. Alternatively, the second part will identify the limited machine capability condition and an unequal capacity in each types of juice that effect to a designing juice proportion. In the last session, it is essential to evaluate an impact of the excess fruit supply that influences to a production decision. However, all of these analytics bases on primary assumptions, therefore in some cases, the results might not be accurate or the most fitted optimization.

To begin with the optimization in production planning of the UniCitrus, a list of constraints and an objective function are shown by the first table, as a side, formulas are listed on the second point. Overview, this analysis divides products as two types; the intermediate goods (Hamlin, Pera, Valencia juices) and the final goods (Standard and Dairy juices), subsequently models in this analysis will focus on these two of production. An inventory system will preserve only the intermediate products only following an assumption that it would be flexible to change in any final productions. As a side, it is important to manipulate a production line in each single month because of a maximization of utilization in machines and labor, plus it might be insufficient in a real circumstance if managers decide to skip a manufacturing for only month in order to increase their short-term profit and ignoring real business routine. Therefore, this model will operate the factory every month. Furthermore, a unit in this analysis uses a tonne (1000kg.) to calculate the results. For example, a machine capability operates in the tonne or ordering orange stocks as the tonne. In summary, it can conclude that there are three main assumption as previously mentioned above or the table1.1.

Table1.1: Primary assumptions in this analysis

Primary assumptios
• *Separate an intermediate goods and final goods production* • *An inventory stock keeps only an intermediate goods* • *Initiating line production every month.* • *A tonne is a unit in this calculate*

Table 2.1: Variables in an environment

Variables Name	*Description*	*Type*
Decision Variables		
Supply_Standard	Total production of Standard for each month. (Tonne)	mpvar
Supply_Dairy	Total production of Dairy for each month. (Tonne)	mpvar
All_Supply	Total production of each month. (Tonne)	mpvar
Orange in uses for each type		
Hamlin_Use_Standard	A number of Hamlin that blends in Standard. (Tonne)	mpvar
Pera_Use_Standard	A number of Pera that blends in Standard. (Tonne)	mpvar
Valen_Use_Standard	A number of Valencia that blends in Standard. (Tonne)	mpvar
Hamlin_Use_Dairy	A number of Hamlin that blends in Dairy. (Tonne)	mpvar
Pera_Use_Dairy	A number of Pera that blends in Dairy. (Tonne)	mpvar
Valen_Use_Dairy	A number of Valencia that blends in Dairy. (Tonne)	mpvar
Demand		
Demand_Standard	Demand of Standard for each month. (Tonne)	Real number
Demand_Dairy	Demand of Dairy for each month. (Tonne)	Real number
Ordering size		
Hamlin_Buy	A number of Hamlin ordering size for each month. (Tonne)	mpvar
Pera_Buy	A number of Pera ordering size for each month. (Tonne)	mpvar
Valen_Buy	A number of Valencia ordering size for each month. (Tonne)	mpvar
Availability of Orange		
Hamlin_avia	A number of expected Hamlin available. (Tonne)	Real number
Pera_avia	A number of expected Pera available. (Tonne)	Real number
Valen_avia	A number of expected Valencia available. (Tonne)	Real number
Proportion constraints		
P_Hamlin_Standard	A specification for Hamlin that are produced in Standard. (Tonne)	mpvar
P_Pera_Standard	A specification for Pera that are produced in Standard. (Tonne)	mpvar
P_Valen_Standard	A specification for Valencia that are produced in Standard. (Tonne)	mpvar
P_Hamlin_Dairy	A specification for Hamlin that are produced in Diary. (Tonne)	mpvar
P_Pera_Dairy	A specification for Pera that are produced in Diary. (Tonne)	mpvar
P_Valencia_Dairy	A specification for Valencia that are produced in Diary. (Tonne)	mpvar

Variable	*Description*	*Type*
Orange Keep		
K_Hamlin	A number of Hamlin inventory. (Tonne)	mpvar
K_Pera	A number of Pera inventory. (Tonne)	mpvar
K_Valen	A number of Valencia inventory. (Tonne)	mpvar
K_all	A number of an aggregate inventory. (Tonne)	mpvar
Addition		
Production_Cap	A machine production Capability. (Tonne)	Integer
Hamlin_Production	An amount of Hamlin used in a production. (Tonne)	mpvar
Pera_Production	An amount of Pear used in a production. (Tonne)	mpvar
Valen_Production	An amount of Valencia used in a production. (Tonne)	mpvar
Inventory_Hamlin	An amount of Hamlin inventory. (Tonne)	linctr
Inventory_Pera	An amount of Pera inventory. (Tonne)	linctr
Inventory_Valen	An amount of Valencia invenrory. (Tonne)	linctr
Bia	A binary value in considering initiate production	mpvar
Old_Standard_Valen	An amount of Valencia that produce Standard.	mpvar
New_Standard_Valen	An amount of new Valencia that produce Standard	mpvar
Old_Dairy_Valen	An amount of new Valencia that produce Dairy	mpvar
New_Dairy_Valen	An amount of new Valencia that produce Dairy	mpvar
Old_Valen	An amount of Valencia from the first supplier	mpvar
New_Valen	An amount of Valencia from the North supplier	mpvar

Table 2.2: Numeric Value

Variables	*Month1*	*Month2*	*Month3*
Hamlin_avia	1050	1225	350
Pera_avia	925	1225	1750
Valen_avia	0	175	350
Demand_Standard	500	1500	700
Demand_Dairy	200	100	100
Production_Cap	1850	1850	1850

Note: Production_Cap are calculated into the tonne in order to compare to a fruit supply in the similar term.
(1850 = 500000 boxes of Pera equal 1850 tonne of orange.)
Therefore, machine capability is 1850 tonne for every fruit.

	Hamlin	*Pera*	*Valencia*	*Standard Price*	*Dairy Price*
Juice (kg/box) for each variety	3.5	3.7	3.4	1000	1100

Optimization Report: An orange juice production Plan

Table 2: Summary of all constraint in the optimization

Objective Function

$$Profit = \sum_{i=1}^{3} (P_s * Supply_{Standard} + P_d * Supply_{Diary}) - (500000F_c) - (10 * K_{all}) - \lambda * All\_supply$$

Description: Note: λ is 1 in the first and second scenarios

Constraints

2.1 Proportion of each juice

Description:

For Standard: $A_i * \frac{F_{Use_{Standard}}}{Supply_{Standard}} \leq S_i$

For Diary: $A_i * \frac{F_{Use_{Standard}}}{Supply_{Standard}} \leq S_i$

Note: $F_{UseStandard}$ = Each fruit juice in uses for producing Standard Juice
A_i = Juice (kg/box) for each variety
S_i = Proportion for each product

2.2 Ordering amount for each fruit

$$Hamlin_{Buy} = HamlinUse_{Standard} + HamlinUse_{Dairy} + K_{Hamlin}$$

$$Pera_{Buy} = PeraUse_{Standard} + PeraUse_{Dairy} + K_{Pera}$$

$$Valencia_{Buy} = ValenciaUse_{Standard} + ValenciaUse_{Dairy} + K_{Valencia}$$

Description: Note: These variables describe the amount of ordering each fruit.

2.3 Comparing Order stock to Availability

Description:

$$Hamlin_{Buy} \leq Hamlin_{avia}$$

$$Pera_{Buy} \leq Pera_{avia}$$

$$Valen_{Buy} \leq Valen_{avia}$$

Note: Buying fruits responding to an availability of fruit in each month.

2.4 Supply function

$$Supply_{Standard} = A_i * HamlinUse_{Standard} + A_i * PeraUse_{Standard} + A_i * ValenciaUse_{Standard}$$

$$Supply_{Dairy} = A_i * HamlinUse_{Dairy} + A_i * PeraUse_{Dairy} + A_i * ValenciaUse_{Dairy}$$

2.5 Limitation on selling

Description:

$$Supply_{Standard} \leq Demand_{Standard}$$

$$Supply_{Dairy} \leq Demand_{Dairy}$$

Note: it is a condition that production must follow a demand

Optimization Report: An orange juice production Plan

2.6 Smoothing Optimization

$$Hamlin_{Buy} \leq 1$$

Description:

Note: control a model uses Hamlin at least 1 tonne. However, it will buy a max Hamlin at 35.7 tonne, apart from that it is unbounded.

2.7 Production Occurrence

$$Hamlin_{Production} = HamlinUse_{Standard} + HamlinUse_{Dairy}$$

$$Pera_{Production} = PeraUse_{Standard} + PeraUse_{Dairy}$$

$$Valencia_{Production} = ValenciaUse_{Standard} + ValenciaUse_{Dairy}$$

2.8 Production Constraints

$$1.1 * Hamlin_{Production} \leq Production_{Cap}$$

$$Pera_{Production} \leq Production_{Cap}$$

$$0.9 * Valen_{Production} \leq Production_{Cap}$$

Description:

Note: Adjustment of the Hamlin that consumes more 10 percent of machine capability and less 10 percent for blending Valencia.

2.9 Storing Stock

$$K_{all} = K_{Hamlin} + K_{Pera} + K_{Valen}$$

$$Inventory_{Fi}(i) := Fi_{Buy}(i) - Fi_{Production}(i) = K_{Fi}(i)$$

$$Inventory_{Fi}(i) := Fi_{Buy}(i) - Fi_{Production}(i) + K_{Fi(i+1)} = K_{Fi}(i)$$

2.10 Model in considering open the production line

$$Hamlin_{Production} + Pera_{Production} + Valen_{Production} \leq 1850 * Bia$$

$$Bia <= 1$$

$$Fi_{Production} \geq or \leq 0.1 * 1850 * Bia$$

$$Hamlin_{Production} + Pera_{Production} + Valen_{Production} \leq m * Bia$$

Description:

Note: All these constraints are set up for answering Question2.
Firstly, the first formula means production decision if a total production from the intermediate goods should not exceed machine capability. Bia means binary value (0 = not produce,1 = produce). Fi means “each fruit type in production” following condition Hamlin spends more capacity and Valencia spend less capacity.
m = 1000000, by m means “Value that determine whether design to run a production or not.

Table 4: Summary of an unlimited machine capability

Scenario: An unlimited machine capability

Production/month	Month: 1	Month: 2	Month: 3
Standard_Product	500.00	1500.00	700.00
Dairy_Product	0.00	100.00	100.00
All Production	500.00	1600.00	800.00

Orange Order	*Month1*	*Month2*	*Month3*
Hamlin	1.000	1.000	1.000
Pera	134.189	427.432	127.000
Valencia	0.000	4.412	96.029

Orange Uses	*Month: 1*	*Month: 2*	*Month: 3*
For Standard			
Hamlin	1.000	1.000	1.000
Pera	134.189	404.460	113.514
Valencia	0.000	0.000	81.323
For Dairy			
Hamlin	0.000	0.000	8.571
Pera	0.000	22.973	13.513
Valencia	0.000	4.412	14.705

Summary Production	*Total*
Net Profit	£ 1,417,100.00
Revenue	£ 2,920,000.00
Fixed Cost	£ 1,500,000.00
Operating Cost	£ 2,900.00
Production Detail	
Machine Capacity	Infinite
Total Production	2900.000

In the scenario of an unlimited machine capability, this production can fulfil demand for last second month, however the first month there is no Valencia selling in the market. As the Dairy needs Valencia in blending at least 15 percent, Dairy is subsequently no longer produced.

As the second table shown, it shows an amount of fruit orders by month. The highest order is Pera. The reason why this model employs Pera as the main proportion because of the constraint specification which require at least 60 percent in Standard and 50 percent for Dairy. As a side one box of Pera can produce 3.7 kg. which is the highest value among others.

It can be seen that Standard is a major product as the high demand. Therefore, almost quantity from the stock uses for build Standard.

In summary, with an infinite machine capacity, the line production can run in responding demand, whereas with a limited market fruit availability, it cannot produce Dairy to respond demand in the first month

As it can be seen from the Orange buy table, Hamlin is not eligible to buy. This can be assumed that the constraints may not consider Hamlin as an important ingredient. Others fruit can substitute Hamlin in order to maximize profit because production can mix Pera and Valencia which is more sufficient. As the Dairy requires Valencia at least 15 percent, ordering only Valencia to mix might be more effective. In other words, Hamlin is not essential ingredient due to no requirement. In an applicable situation, it can reduce Hamlin operation cost. However, this report might require Hamlin for 1 tonne to show that Hamlin is able to use but not necessary.

Table 5: A limited machine Capability

Scenario: A limited machine capability = 1850

Production/month	Month: 1	Month: 2	Month: 3
Standard_Product	500.00	1500.00	700.00
Dairy_Product	0.00	100.00	100.00
All Production	500.00	1600.00	800.00

Orange Order	*Month1*	*Month2*	*Month3*
Hamlin	35.714	43.514	21.775
Pera	101.350	387.216	191.564
Valencia	0.000	4.412	4.412

Orange Uses	*Month: 1*	*Month: 2*	*Month: 3*
For Standard			
Hamlin	35.714	43.514	13.203
Pera	101.350	364.243	176.699
Valencia	0.000	0.000	0.000
For Dairy			
Hamlin	0.000	0.000	8.571
Pera	0.000	22.973	14.864
Valencia	0.000	4.412	4.412

Summary Production		*Total*
Net Profit	£	1,417,100.00
Revenue	£	2,920,000.00
Fixed Cost	£	1,500,000.00
Operating Cost	£	2,900.00
		Production Detail
Machine Capacity		1850
Total Production		2900.000

In the scenario of a limited machine capability, the production is likely to be similar the previously mentioned circumstance due to no Valencia in the first month.

To calculate a machine capacity, we assume that Pera 500,000 boxes can produce 1850 tonne of blending, subsequently every orange type can process 1850 tonne a month.
((3.7*500000)/1000)
Although there is a limited capacity of devices, it can respond to the market demand because the maximum demand is 1600 tonne.

The way this model worked is based on the description on constraint 2.10 above. We think of how to create machine capacity at first, then think of productive consumption in Hamlin and Valencia.

As the second table shown, it shows an amount of fruit orders by month. The highest order is also Pera, but this model there is a significant order from Hamlin. In this model Hamlin becomes a minor ingredient instead of Valencia because Hamlin might utilize a machine rather than Valencia. However, Valencia is still important to maintain Dairy production as it requires at least 15 percent.

In summary, with a finite machine capacity, the line production can run in responding demand, whereas there is a major change in a fruit order. Hamlin becomes a major choice of the production. Furthermore, there is no change in a profit and cost.

Table 6: An excess supply of Valencia oranges

Scenario: An excess supply of Valencia oranges

Production/month	Month: 1	Month: 2	Month: 3
Standard_Product	500.000	1500.000	700.000
Dairy_Product	200.000	100.000	100.000
All Production	700.000	1600.000	800.000

Orange Order	*Month1*	*Month2*	*Month3*
Hamlin	1.000	0.000	1.000
Pera	162.162	417.973	202.703
Valencia	14.191	7.353	6.838

Orange Uses	*Month: 1*	*Month: 2*	*Month: 3*
For Standard			
Hamlin	0.000	0.000	0.000
Pera	135.135	404.459	189.189
Valencia	0.000	0.000	0.000
For Dairy			
Hamlin	1.000	0.000	1.000
Pera	27.027	13.514	13.513
Valencia	14.191	7.353	6.838

Summary Production	*Total*
Net Profit	£ 1,640,000.00
Revenue	£ 3,140,000.00
Fixed Cost	£ 1,500,000.00
Operating Cost	λ

	Production Detail
Machine Capacity	Unlimited
Total Production	3100.000

In a situation of excess Valencia fruit, it allows manufacturer to produce Dairy in the first month, which increases a company profit t0 1.64 million pound.

This model based on the first model and it is applied a new Valencia from the norther of Brazil as a new supplier. Consequently, an array in a production is added one column (New Valencia buy). In order to add the new Valencia buy variable into decision variables, we separate an amount of order into an use for Standard and Dairy. Next we sum them into a total and apply it with decision variable.

As the result, New Valencia order substitute all the old Valencia order. And this helps to evaluate lambda with a former cost (former cost=1)

In the analysis of cost changes, lambda which is the cost, can apply in a decision of buying Valencia. If lambda cost* a quantity from New supplier less than 220000 (200*1100), we can use the new supplier. (This idea is like comparing marginal revenue and marginal cost.)

Further suggestions

Orange Store	*Month: 1*	*Month: 2*	*Month: 3*
K_Hamlin	0.000	0.000	0.000
K_Pera	0.000	0.000	0.000
K_Valen	0.000	0.000	0.000

From all of these algorithms, the first particular suggestion is there is an inventory for intermediate products. It can be improved by storing fruit in order to reduce an operating cost in reality. Secondly, it can be developed by maximize a machine capability corresponding to the inventory.

Optimization Report: An orange juice production Plan

Appendix

Code Block

```
model Tour

 uses "mmxprs";

 declarations
  !Objecyive function and month
  month = 1..3

  !Decision Variable to objective, Supply
  Supply_Standard : array(month) of mpvar
  Supply_Dairy : array(month) of mpvar
  All_Supply : array(month) of mpvar

  !Orange in uses for each type, minor decision var
  Hamlin_Use_Standard : array(month) of mpvar
  Pera_Use_Standard : array(month) of mpvar
  Valen_Use_Standard : array(month) of mpvar

  Hamlin_Use_Dairy : array(month) of mpvar
  Pera_Use_Dairy : array(month) OF mpvar
  Valen_Use_Dairy : array(month) of mpvar

  !Demand
  Demand_Standard : array(month) of integer
  Demand_Dairy : array(month) of integer

  !Var to check with aviability, buy for each month
  Hamlin_Buy : array(month) of mpvar
  Pera_Buy : array(month) of mpvar
  Valen_Buy : array(month) of mpvar

  !Aviability in each month
  Hamlin_avia : array(month) of integer
  Pera_avia : array(month) of integer
  Valen_avia : array(month) of integer

  !Proportion constraints
  P_Hamlin_Standard : array(month) of mpvar
  P_Pera_Standard : array(month) of mpvar
  P_Valen_Standard : array(month) of mpvar
  P_Hamlin_Dairy : array(month) of mpvar
  P_Valen_Dairy : array(month) of mpvar
  P_Pera_Dairy : array(month) of mpvar

  !Orange keep
  K_Hamlin : array(month) of mpvar
  K_Pera : array(month) of mpvar
  K_Valen : array(month) of mpvar
```

Optimization Report: An orange juice production Plan

```
  K_all : array(month) of mpvar
 !Addition
 Production_Cap : array(month) of integer
 Hamlin_Production :array(month) of mpvar
 Pera_Production :array(month) of mpvar
 Valen_Production:array(month) of mpvar
 m : array(month) of integer

 Inventory_Hamlin :array(month) of linctr
 Inventory_Pera :array(month) of linctr
 Inventory_Valen :array(month) of linctr

 New_Standard_Valen :array(month) of mpvar
 Old_Standard_Valen :array(month) of mpvar
 Old_Dairy_Valen :array(month) of mpvar
 New_Dairy_Valen :array(month) of mpvar
 Old_Valen :array(month) of mpvar
 New_Valen :array(month) of mpvar
 New_Valen_avai:array(month) of mpvar
 lambda : real
 New_Valen_Buy :array(month) of mpvar
 Valen_Buy_Total:array(month) of mpvar
 end-declarations


 !List numeric array
 !Aviability Constaints Value
 Hamlin_avia :: [1050,1225,350]
 Pera_avia :: [925, 1225,1750]
 Valen_avia :: [0,175,350]
 !Demand Constrain Value
 Demand_Standard :: [500,1500,700]
 Demand_Dairy :: [200,100,100]
 m ::[1,1,1]
```

Question 1

```
 !Objective function
  profit := (1000*(SUM(i in month) Supply_Standard(i))) + (1100*(SUM(i in month) Supply_Dairy(i))) -
(500000*(sum(i in month) m(i))) - (10*(SUM(i in month) K_all(i))) -((SUM(i in month) All_Supply(i)))
 !!!!!!Contraints!!!!!!!!
 !!!!!!!!!!Create supply!!!!!!!!!!

  !Proportion created and specification for each products!
 forall(i in month)do
  3.5*Hamlin_Use_Standard(i) <= 0.25*Supply_Standard(i)
  3.7*Pera_Use_Standard(i) >= 0.6*Supply_Standard(i)
  3.4*Valen_Use_Standard(i) <= 0.4*Supply_Standard(i)
 end-do
```

```
 forall(i in month) do
   3.5*Hamlin_Use_Dairy(i) <= 0.3*Supply_Dairy(i)
   3.7*Pera_Use_Dairy(i) >= 0.5*Supply_Dairy(i)
   3.4*Valen_Use_Dairy(i) <= 0.5*Supply_Dairy(i)
   3.4*Valen_Use_Dairy(i) >= 0.15*Supply_Dairy(i)
   end-do

 !Buy fruit check
 forall(i in month) do
   Hamlin_Use_Standard(i) + Hamlin_Use_Dairy(i) + K_Hamlin(i) + K_Hamlin(i)*3.5 + K_Pera(i)*3.7 +
K_Valen(i)*3.4 = Hamlin_Buy(i)
   Pera_Use_Standard(i) + Pera_Use_Dairy(i) + K_Pera(i) + K_Hamlin(i)*3.5 + K_Pera(i)*3.7 + K_Valen(i)*3.4
= Pera_Buy(i)
   Valen_Use_Standard(i) + Valen_Use_Dairy(i)+ K_Valen(i) + K_Hamlin(i)*3.5 + K_Pera(i)*3.7 +
K_Valen(i)*3.4 = Valen_Buy(i)
 end-do

  !Compare Buy and Aviability in mrk
 forall(i in month) do
   Hamlin_Buy(i) <= Hamlin_avia(i)
   Pera_Buy(i) <= Pera_avia(i)
   Valen_Buy(i) <= Valen_avia(i)
 end-do

 !Supply function
 forall(i in month) do
   Hamlin_Use_Standard(i)*3.5 + Pera_Use_Standard(i)*3.7 + Valen_Use_Standard(i)*3.4 =
Supply_Standard(i)
   Hamlin_Use_Dairy(i)*3.5 + Pera_Use_Dairy(i)*3.7 + Valen_Use_Dairy(i)*3.4 = Supply_Dairy(i)
   Supply_Standard(i)+Supply_Dairy(i) = All_Supply(i)
 end-do

 !Demand = Supply constraints
 forall(i in month) do
   Supply_Standard(i) <= Demand_Standard(i)
   Supply_Dairy(i) <= Demand_Dairy(i)
 end-do

 !Production Condition
 !Set for smoothing optimization
 forall(i in month)do
    Hamlin_Buy(i) >= 1
 end-do

 !Keep stock
 forall(i in month)do
   K_Hamlin(i)+K_Pera(i)+K_Valen(i) = K_all(i)
 end-do

 !No inventory last month
 K_Hamlin(3) = 0
```

```
 K_Pera(3) = 0
 K_Valen(3) = 0
 Supply_Standard(3) = Demand_Standard(3)
 Supply_Dairy(3) = Demand_Dairy(3)

!Improve production
!Create Var by each orange type to store how many tonne used in a production
 forall(i in month)do
   Hamlin_Production(i) = Hamlin_Use_Standard(i) + Hamlin_Use_Dairy(i)
   Pera_Production(i) = Pera_Use_Standard(i) + Pera_Use_Dairy(i)
   Valen_Production(i) = Valen_Use_Standard(i) + Valen_Use_Dairy(i)
 end-do
!To keep Hamlin, Pera, Valencia
 Inventory_Hamlin(1) := Hamlin_Buy(1) - Hamlin_Production(1) = K_Hamlin(1)
 Inventory_Hamlin(2) := Hamlin_Buy(2) - Hamlin_Production(2) + K_Hamlin(1) = K_Hamlin(2)
 Inventory_Hamlin(3) := Hamlin_Buy(3) - Hamlin_Production(3) + K_Hamlin(2) = K_Hamlin(3)

 Inventory_Pera(1) := Pera_Buy(1) - Pera_Production(1) = K_Pera(1)
 Inventory_Pera(2) := Pera_Buy(2) - Pera_Production(2) + K_Pera(1) = K_Pera(2)
 Inventory_Pera(3) := Pera_Buy(3) - Pera_Production(3) + K_Pera(2) = K_Valen(3)

 Inventory_Valen(1) := Valen_Buy(1) - Valen_Production(1) = K_Valen(1)
 Inventory_Valen(2) := Valen_Buy(2) - Valen_Production(2) + K_Valen(1) = K_Valen(2)
 Inventory_Valen(3) := Valen_Buy(3) - Valen_Production(3) + K_Valen(2) = K_Valen(3)
```

Question: 2

```
!!!!!!!!Question2!!!!!!!!
   Production_Cap :: [1850,1850,1850] !! in each month, capacity = 1850 tonne
 !Create Var by each orange type to store how many tonne used in a production
 forall(i in month)do
   Hamlin_Production(i) = Hamlin_Use_Standard(i) + Hamlin_Use_Dairy(i)
   Pera_Production(i) = Pera_Use_Standard(i) + Pera_Use_Dairy(i)
   Valen_Production(i) = Valen_Use_Standard(i) + Valen_Use_Dairy(i)
 end-do

 !Production Constraints
 !Machine constraints
 forall(i in month) do
   1.1*Hamlin_Production(i) <= Production_Cap(i)
   Pera_Production(i) <= Production_Cap(i)
   0.9*Valen_Production(i) <= Production_Cap(i)
 end-do

 !Binary to check if opening the production line or not
 forall(i in month)do
   Hamlin_Production(i) + Pera_Production(i) + Valen_Production(i) <= 1850*Bia(i)
   Bia(i) <= 1
   Hamlin_Production(i) >= 0.1*1850*Bia(i)
   Valen_Production(i) <= 0.1*1850*Bia(i)
   Hamlin_Production(i) + Pera_Production(i) + Valen_Production(i) <= 10000000*Bia(i)
 end-do
```

Optimization Report: An orange juice production Plan

```
!!!!!!!!!!!!END Q2!!!!!!!!!!!!!
```

Question: 3

```
!!!!!!Q3!!!!!!!!!!
!Condition for Question 3
 forall(i in month)do
 New_Standard_Valen(i) + Old_Standard_Valen(i) = Valen_Use_Standard(i)
 New_Dairy_Valen(i) + Old_Dairy_Valen(i) = Valen_Use_Dairy(i)
 Old_Standard_Valen(i) + Old_Dairy_Valen(i) = Old_Valen(i)
 New_Standard_Valen(i) + New_Dairy_Valen(i) = New_Valen(i)
 Old_Valen(i) + New_Valen(i) = Valen_Use_Standard(i)+Valen_Use_Dairy(i)
 Valen_Use_Standard(i)+Valen_Use_Dairy(i) <= Valen_avia(i) + New_Valen_avai(i)
 New_Valen(i) <= New_Valen_avai(i)
 Old_Valen(i) <= Valen_avia(i)
 end-do
   !Supply function, Production function
 forall(i in month) do
   Hamlin_Use_Standard(i)*3.5 + Pera_Use_Standard(i)*3.7 + Valen_Use_Standard(i)*3.4 +
New_Standard_Valen(i)*3.4 = Supply_Standard(i)
   Hamlin_Use_Dairy(i)*3.5 + Pera_Use_Dairy(i)*3.7 + Valen_Use_Dairy(i)*3.4 + New_Dairy_Valen(i)*3.4  =
Supply_Dairy(i)
   Supply_Standard(i)+Supply_Dairy(i) = All_Supply(i)
 end-do

 !Buy fruit check, to make New Valencia
 forall(i in month) do
   Hamlin_Use_Standard(i) + Hamlin_Use_Dairy(i) + K_Hamlin(i) = Hamlin_Buy(i)
   Pera_Use_Standard(i) + Pera_Use_Dairy(i) + K_Pera(i) = Pera_Buy(i)
   Valen_Use_Standard(i) + Valen_Use_Dairy(i)+ K_Valen(i)+ New_Valen_Buy(i) = Valen_Buy_Total(i)
   New_Standard_Valen(i) + New_Dairy_Valen(i) = New_Valen_Buy(i)
 end-do
!Lambda value
 lambda := 1100
 forall(i in month) do
 Valen_Buy(i) + New_Valen_Buy(i) =Valen_Buy_Total(i)
 end-do
 !Compare Buy and Aviability in mrk, unlock problem of month 1
 forall(i in month) do
   Hamlin_Buy(i) <= Hamlin_avia(i)
   Pera_Buy(i) <= Pera_avia(i)
 end-do
 Valen_Buy_Total(1) >= Valen_avia(1)
 Valen_Buy_Total(2) <= Valen_avia(2)
 Valen_Buy_Total(3) <= Valen_avia(3)
!!!!!!END Q3!!!!!!

 maximize(profit)
        writeln("Profit is: ", getobjval, " GBP")

FORALL(i in month) DO
```

```
        writeln("-----------------:")
        writeln("Month: ",i)
        writeln("Machine Capacity: ",getsol(Production_Cap(i)))
                writeln("Supply_Standard= ", getsol(Supply_Standard(i)))
                writeln("Supply_Dairy= ", getsol(Supply_Dairy(i)))
                writeln("Production Capacity= ", getsol(All_Supply(i)))

                writeln("Hamlin_Use_Standard is: ", getsol(Hamlin_Use_Standard(i)))
                writeln("Hamlin_Use_Dairy is: ", getsol(Hamlin_Use_Dairy(i)))
                writeln("K_Hamlin is: ", getsol(K_Hamlin(i)))
   writeln("Hamlin_Buy is: ", getsol(Hamlin_Buy(i)))

                writeln("Pera_Use_Standard is: ", getsol(Pera_Use_Standard(i)))
                writeln("Pera_Use_Dairy is: ", getsol(Pera_Use_Dairy(i)))
                writeln("K_Pera is: ", getsol(K_Pera(i)))
                writeln("Pera_Buy is: ", getsol(Pera_Buy(i)))

   writeln("Valen_Use_Standard is: ", getsol(Valen_Use_Standard(i)))
                writeln("Valen_Use_Dairy is: ", getsol(Valen_Use_Dairy(i)))
                writeln("K_Valen is: ", getsol(K_Valen(i)))
   writeln("Valen_Buy is: ", getsol(Valen_Buy(i)))
   writeln("------")


        END-DO

end-model
```