Operation Research Project

Stardew Valley Optimization

Submitted by

Anon	**Pongsawang**	**T08901201**
Ch-non	**Tosomsakul**	**T08902204**
Punyapa	**Poonapanont**	**T08901203**
Satsawat	**Suttawuttiwong**	**T08902202**
Tai	**Tantipiwatanaskul**	**T08902205**
Thanakorn	**Prayoonkittikul**	**T08902201**
Thunyasorn	**Pornkraisri**	**T08901204**

Presenting to

Albert Y. Chen

Associate Professor, Operation Research
Civil Engineering, National Taiwan University

16th January 2020

About the problem

Agriculture is undeniably one of the most important sectors in Thailand. About 40% of Thai's population works in agriculture-related jobs [1] and more than 43% of Thai's total area is dedicated to agriculture[2]. However, only about 8% of Thai GDP comes from the agricultural sector, which is considered small compared to the industrial sector (39%) and the service sector (52%). Moreover, the average annual salary of Thailand farmers is only about 2,460 USD[4] which is also considered small compared to Thailand's GDP per capita (6,590 USD)[5]. We now can see that there are rooms for Thailand's agricultural sector to increase their income which can then lead to a better life quality of Thai people. Therefore, we conduct a project to research on how to maximize the revenue of Thai agriculture. We will build a model that can suggest species of crops that farmers should plant in different seasons to maximize their profit. However, since the data about crop types and prices in each year are different, and there are also some limits of planting some types of crops in certain areas, we will build a farming model based on a farming simulator game called *Stardew Valley* instead. We hope that we can use our model as a base to help the Thai government to build a more complex model based on real data.

Literature review

In order to increase the revenue in agriculture, we can do both in increase the production of crops and reduce the cost of materials which can be water and fertilizer usage or even labor activities. Although there were several criteria to optimize, one of the research[6] show that an Optimization Multicriteria Mathematical Programming model can simultaneously optimize several conflict of criteria which are gross margin maximization, fertilizer minimization and labor minimization. This model tries to find the optimum farm resource allocations, which have a fixed set of crops, to optimize the three criteria. According to the optimization researches on agriculture[6, 7, 8], they have used models that we didn't learn in class and many of their variable is hard to acquire. Thus, we decided to create a model upon the Stardew Valley game which aid us in term of data collecting and this game system is similar to the real world agriculture process. While analyzing the features, we come up with the knapsack model that can be used for crop planning. After we got the model to work on, we tried to look for a research that use this model on crop production planning but we can't find it.

Objective and Assumption

Stardew Valley Optimization is the project to maximize the crop profit in Stardew Valley, one of the best farming games, where we consider it to be a simulation of the real-world farm. The assumptions related to the project are as follows;

- Not taking processed goods and the grade of plants into consideration
- Observing only in the Single Player Mode
- Assume player has unlimited stamina
- All plants will be watered simultaneously using sprinklers in the game
- Seeds will be bought at Piere's
- In Stardew Valley, some crops can be grown across seasons, we assume every crop grow and die within each season.
- No seed drop after harvesting the plants
- Assume no double plant drop (the game often allows extra crop harvest)
- We will not destroy plants manually. Each and every crop will be planted until it dies naturally.

The main challenges where the optimized system in Stardew Valley to apply to the real-world are mainly the challenges in data collecting, real-world constraints, and unexpected events. Data collecting can be difficult due to the unclear data and variables to use for optimizing, as events, for example; natural disasters and plant diseases also can not be foreseen. For a day-to-day basis farming, the built system will be able to experiment with the maximizing profit process. The goal of the project is to maximize the profit from crop growing in Stardew Valley. The cost and profit are based on the game simulation environment where the system is designed to be the most alike with real-world agriculture.

Modeling Process

Data Collection:

Most of the data collection about the game was gathered from Stardew Valley Wiki. Stardew Valley Wiki provided information such as: the crop availability for each season, its buying and selling prices, and its harvesting period.

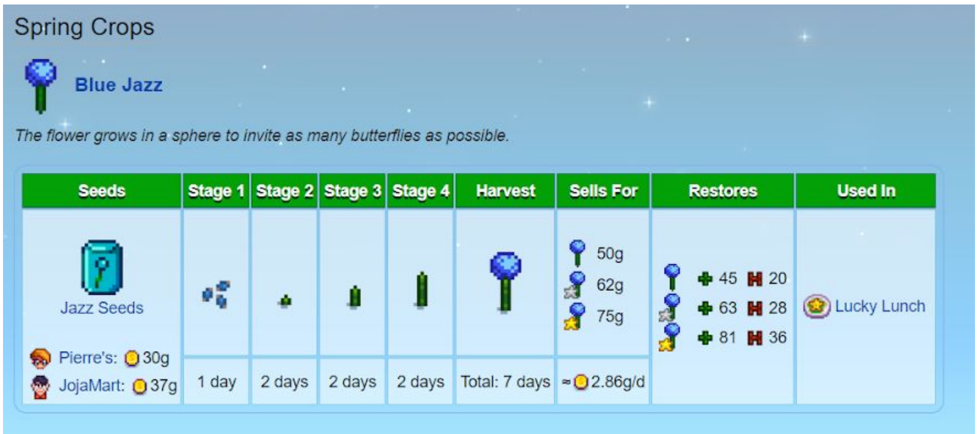

Fig.1 Example of data on Stardew Valley Wiki Page

The information of each plant was inserted into an excel file. From the given information the column 'Growing Days' and 'Profit per Growing Days' was generated. Growing days is the number of days the plant takes to grow. For single type plant (Type: S), the growing days is the same as days to mature. On the other hand, continuous type of plants (Type: C) are plants where they only need to be planted once but are able to continuously produce crops until the end of the season. The growing days of continuous plants are calculated from the following formula:

$$Growing\ Days\ =\ Days\ to\ Mature\ +\ \left[floor\left(\frac{28 - Days\ to\ Mature}{Days\ to\ Regrow}\right) \times Days\ to\ Regrow\right]$$

Profit per Growing Days represents the amount of profit the player would earn by planting and harvesting the crop at the end of its growing period.

Season	Name	Plant type	Days to Maturity	Days to Regrow	Number of Harvest per seed	Growing Days	Seed Price	Sell Price for one item	Crops per Harvest	Sell Price per Harvest	Profit per Growing Days
Spring	Blue Jazz	S	7	999	1	7	30	50	1	50	20
Spring	Cauliflower	S	12	999	1	12	80	175	1	175	95
Spring	Coffee Bean	C	10	2	10	28	2,500	15	4	60	-1,900
Spring	Garlic	S	4	999	1	4	40	60	1	60	20
Spring	Green Bean	C	10	3	7	28	60	40	1	40	220
Spring	Kale	S	6	999	1	6	70	110	1	110	40
Spring	Parsnip	S	4	999	1	4	20	35	1	35	15
Spring	Potato	S	6	999	1	6	50	80	1	80	30
Spring	Rhubarb	S	13	999	1	13	100	220	1	220	120
Spring	Strawberry	C	8	4	6	28	100	120	1	120	620
Spring	Tulip	S	6	999	1	6	20	30	1	30	10
Spring	Unmilled Rice	S	7	999	1	7	40	30	1	30	-10
				value							
				weight							

Fig. 2 Spring Crop Information

	A	B	C	D	E	F	G	H	I	J	K	L
1	Season	Name	Plant type	Days to Maturity	Days to Regrow	Numbers of Harvest per Seed	Growing Days	Seed Price	Sell Price for one item	Crops per Harvest	Sell Price per Harvest	Profit per Total Growing Days
2	Summer	Blueberry	C	13	4	4	25	80	50	3	150	520
3	Summer	Corn	C	14	4	4	26	150	50	1	50	50
4	Summer	Hops	C	11	1	18	28	60	25	1	25	390
5	Summer	Hot pepper	C	5	3	8	26	40	40	1	40	280
6	Summer	Melon	S	12	9999	1	12	80	250	1	250	170
7	Summer	Poppy	S	7	9999	1	7	100	140	1	140	40
8	Summer	Radish	S	6	9999	1	6	40	90	1	90	50
9	Summer	Red Cabbage	S	9	9999	1	9	100	260	1	260	160
10	Summer	Starfruit	S	13	9999	1	13	400	750	1	750	350
11	Summer	Summer Spangle	S	8	9999	1	8	50	90	1	90	40
12	Summer	Sunflower	S	8	9999	1	8	200	80	1	80	-120
13	Summer	Tomato	C	11	4	5	27	50	60	1	60	250
14	Summer	Wheat	S	4	9999	1	4	10	25	1	25	15
15												
16												
17					value							
18					weight							
19												

Fig. 3 Summer Crop Information

	A	B	C	D	E	F	G	H	I	J	K	L
1	Season	Name	Plant type	Days to Maturity	Days to Regrow	Numbers of Harvest per Seed	Growing Days	Seed Price	Sell Price for one item	Crops per Harvest	Sell Price per Harvest	Profit per Growing Days
2	Fall	Amarant	S	7	999	1	7	70	150	1	150	80
3	Fall	Artichoke	S	8	999	1	8	30	160	1	160	130
4	Fall	Beet	S	6	999	1	6	20	100	1	100	80
5	Fall	Bok Choy	S	4	999	1	4	50	80	1	80	30
6	Fall	Cron	C	14	4	4	26	150	50	1	50	50
7	Fall	Cranberries	C	7	5	5	27	240	75	2	150	510
8	Fall	Eggplant	C	5	5	5	25	20	60	1	60	280
9	Fall	Fairy Rose	S	12	999	1	12	200	290	1	290	90
10	Fall	Grape	C	10	3	7	28	60	80	1	80	500
11	Fall	Pumpkin	S	13	999	1	13	100	320	1	320	220
12	Fall	Yam	S	10	999	1	10	60	160	1	160	100
13												
14												
15												
16												
17					value							
18					weight							

Fig. 4 Fall Crop Information

	A	B	C	D	E	F	G	H	I	J	K	L
1	Season	Name	Plant type	Days to Maturity	Days to Regrow	Numbers of Harvest per Seed	Growing Days	Seed Price	Sell Price for one item	Crops per Harvest	Sell Price per Harvest	Profit per Growing Days
2	Winter	Crocus	S	6	999	1	6	30	120	1	120	90
3	Winter	Snow Yam	S	4	999	1	4	30	100	1	100	70
4	Winter	Winter Root	S	5	999	1	5	30	105	1	105	75
5	Winter	Crystal Fruit	C	9	5	4	24	30	150	1	150	570
6												
7					value							
8					weight							
9												

Fig. 5 Winter Crop Information

Model Formulation:

The objective of this simulation is to find what crop should be planted and harvested to maximize the profit for each of the seasons. Each season has a 28 days cycle, however, each plant has its own unique profit and harvesting time. This problem setup shares many similarities with the knapsack problem, where the capacity is 28 days, the value is the profit of each plant, and the weight being the growing days. Therefore, to find the maximum profit for each season is just performing the knapsack 4 times for Spring, Summer, Fall, and Winter.

The mathematical formulation of this problem:

$$
\begin{aligned}
& max\ z = \sum_{i=1}^{n} v_i * m_i \\
& s.t. \\
& \sum_{i=1}^{n} w_i * m_i \leq C \\
& m_i \geq 0 \ and\ integer,\ i = 1, 2, 3, ..., n
\end{aligned}
$$

v_i: profit of plant i
m_i: the amount of time that plant i will be planted
w_i: growing days

Note: The python code are separated into two (2) parts, the knapsack class and the execution part

```
class KnapsackDPResult(object):
    def __init__(self, f, result):
        # instance variables
        # optimal objective value
        self.f = f
        # optimal combination of goods in knapsack
        self.result = result

def knapsackDP(C,W,V):
    # C is the capacity, which is a constant
    # W is the weight vector
    # V is the value vector

    # fill in the logic for Dynamic Programming here:
    items = len(W)

    f = numpy.zeros((items+1, C+1))

    # For storing optimal m
    optM = numpy.zeros((items+1, C+1))
    # For storing optimal f
    result = numpy.zeros(items)

    iSet = range(items)
    iSet.reverse()

    for i in iSet:
        for j in range(C+1):
            m = numpy.zeros(C+1)
            for k in range(int(math.floor(C/W[i]))+1):
                if(j-W[i]*k >= 0):
                    m[k] = V[i]*k + f[i+1][j-W[i]*k]
            indexM = numpy.argmax(m)
            f[i][j] = m[indexM]
            optM[i][j] = indexM

    optIndex = numpy.unravel_index(numpy.argmax(f, axis=None), f.shape)
    optf = f[optIndex]

    x = 0
    for i in range(items):
        if(i==0):
            x = optIndex[1]
            result[i] = int(optM[i][x])
        else:
            x = int(x-W[i-1]*result[i-1])
            result[i] = int(optM[i][x])

    return KnapsackDPResult(optf, result)
```

Fig.6 Codes of knapsack class

```python
import knapsackDP as knapsackDP

reload(knapsackDP)

days = 28 # There is 28 days in a season

def optimizeCrop(season, items, weights, values):
    result = knapsackDP.knapsackDP(days, weights , values)
    print '\n'+season+' :'
    for i in range(0,len(items)):
        if(result.result[i] != 0):
            print(items[i]+' : '+str(int(result.result[i]))+' time(s)')
    print('Net Profit : '+str(result.f))


# Spring
items_spring = ['Blue Jazz', 'Cauliflower', 'Coffee Bean', 'Garlic', 'Green Bean', 'Kale', 'Parnsip', 'Potato',
                'Rhubarb', 'Strawberry', 'Tulip', 'Unmilled Rice']
weights_spring = [7, 12, 28, 4, 28, 6, 4, 6, 13, 28, 6, 7]
values_spring = [20, 95, -1900, 20, 220, 40, 15, 30, 120, 620, 10, -10]
optimizeCrop('Spring', items_spring, weights_spring, values_spring)

# Summer
items_summer = ['Blueberry', 'Corn', 'Hops', 'Hot Pepper', 'Melon', 'Poppy', 'Radish', 'Red Cabbage', 'Starfruit',
                'Summer Spangle', 'Sunflower', 'Tomato', 'Wheat']
weights_summer = [25, 26, 28, 26, 12, 7, 6, 9, 13, 8, 8, 27, 4]
values_summer = [520, 50, 390, 280, 170, 40, 50, 160, 350, 40, -120, 250, 15]
optimizeCrop('Summer', items_summer, weights_summer, values_summer)

# Fall
items_fall = ['Amarant', 'Artichoke', 'Beet', 'Bok Choy', 'Cranberries', 'Eggplant', 'Fairy Rose', 'Grape',
              'Pumpkin', 'Yam']
weights_fall = [7,8,6,4,27,25,12,28,13,10]
values_fall = [80,130,80,30,510,280,90,500,220,100]
optimizeCrop('Fall', items_fall, weights_fall, values_fall)

# Winter
items_winter = ['Crocus', 'Snow Yam', 'Winter Root', 'Crystal Fruit']
weights_winter = [6 , 4, 5, 24]
values_winter = [90, 70, 75, 570]
optimizeCrop('Winter', items_winter, weights_winter, values_winter)
```

Fig.7 Codes of the main execution part

Results

```
In [9]: runfile('/Users/bingo/Documents/NTU 108-1/Operation Research/Stardew Project/
stardew.py', wdir='/Users/bingo/Documents/NTU 108-1/Operation Research/Stardew
Project')
Reloaded modules: knapsackDP
Spring :
Strawberry : 1 time(s)
Net Profit : 620.0

Summer:
Starfruit : 2 time(s)
Net Profit 700.0

Fall:
Cranberries : 1 time(s)
Net Profit : 510.0

Winter:
Snow Yam : 1 time(s)
Crystal Fruit : 1 time(s)
Net Profit : 640.0
```

Fig.8 Test Run Result

Our final results can be seen at fig.6, and from the following:
In spring: plant strawberries 1 time. (C-type plant)
In summer: plant starfruit 2 times. (S-type plant)
In fall: plant cranberries 1 time. (C-type plant)
In winter: plant snow yam 1 time (S-type plant) followed by crystal fruit 1 time (S-type plant)

C-type plant: need to be planted only once and continuously produce crops
S-type plant: need to be planted again after the harvest

We are very sure that our input values are correct since it is derived from the game. We are also very confident in our outputs since it makes a lot of sense when we actually substitute our output variables and compute the profit. For example, in the fall season, we can see that cranberries have significantly higher profit compared to the rest and it's impossible to come up with a combination of other crops that can out-profit the cranberries. In addition, our model is based on a knapsack algorithm, which is already well-proven about its correctness.

Our thoughts

The project benefits us in many ways. First, as what we have learned so far in class is all about the theory, the project gave us a chance to try to use what we have learned in class and apply it into real-world problems which lead us to a deeper understanding of all the concepts. The project also provides an opportunity for us to refresh and sharpen our skills. In the project, we use dynamic programming to solve our problem which is very similar to the Knapsack problem we learned from the class. What we have to do is to revise the Knapsack problem and do some modification to make it fits with the data we provided and the outcome we wanted. Lastly, another benefit we gain while working on the project is an improvement in creativity, problem-solving, critical thinking skills, and working in teams since the project lets us come up with our own problem that we want to solve and solve it as a team.

Work distribution

Anon Pongsawang	14.3%
Ch-non Tosomsakul	14.3%
Punyapa Poonapanont	14.3%
Satsawat Suttawuttiwong	14.3%
Tai Tantipiwatanaskul	14.3%
Thanakorn Prayoonkittikul	14.3%
Thunyasorn Pornkraisri	14.3%

References

[1] Agriculture in Thailand - https://en.wikipedia.org/wiki/Agriculture_in_Thailand
[2] Thailand Agricultural Land
https://tradingeconomics.com/thailand/agricultural-land-percent-of-land-area-wb-data.html
[3] Economy of Thailand - https://en.wikipedia.org/wiki/Economy_of_Thailand
[4] Thai farmers income -
https://www.thailand-business-news.com/economics/73550-thai-farmers-income-on-the-rise.html
[5] GDP per capita (Thailand) -
https://data.worldbank.org/indicator/NY.GDP.PCAP.CD?locations=TH
[6] Manos, B. & Chatzinikolaou, P. & Kiomourtzi, F. "Sustainable Optimization of Agriculture Production". 19-20, Jan 2013.[Online]. Available:

https://www.sciencedirect.com/science/article/pii/S2212670813000729.[Accessed Jan. 12, 2020].
[7] Pap, Z. “Crop rotation constraints in agricultural production planning”. [Online]. Available: https://www.researchgate.net/publication/224345826_Crop_rotation_constraints_in_agricultural_production_planning.[Accessed Jan. 14, 2020]
[8] Razali. & Nasution, Z. & Rahmawaty. “Optimization Model On The Use Of Agriculture Land In The Catchment Area Of Lake Toba”. 11 Nov, 2014. [Online]. Available: https://www.researchgate.net/publication/270583079_Optimization_Model_On_The_Use_Of_Agriculture_Land_In_The_Catchment_Area_Of_Lake_Toba. [Accessed Jan. 14, 2020]