Test Case Summary

Damon Raynor – CS-5260-50 – Programming Project Part 1

1 Preface

The following gives an overview of each test case that I tried for the programming project part 1.

2 Test Cases

For each test case I show the (1) initial state, (2) identify my parameter settings, (3) remind you of my resource weights and finally (4) show you the resulting five schedules. Be sure to read each schedule from bottom to top. Start from the “ROOT NODE” and work your way up to the depth bound.

2.1 Test Case 1

This was my initial test case. I used this to help me get my footing with getting the program running. My initial state resources for each country are low, I used generic parameters, and had a shallow search depth of 3.

My EUs were okay, and I observed virtually all transfers from another country to myself. It’s not shown here, but due to my initial state resources, my country could only do a total of four transforms. All of the other possible actions were a mountain of transfers. This will be the case for the remaining test cases.

2.2 Test Case 2

For test case 2, I changed the discounted reward in hopes of getting better EU results. It worked! I was going to mess around with changing my other parameters and increase the depth bound until I got to my next test case.

2.3 Test Case 3

For Test Case 3, I changed the depth bound value to 5 and my C value to -.2 (with the hopes of getting better utilities), but I ran into the issue of my computer not being able to handle the super large number of calculations.

I learned through experience that this is why depth first search isn’t a good strategy for these implicit graphs. It restricts my depth bound, my initial state resource qty and the number of countries from increasing in size and complexity.

With the current setup, my root node produces 58 children at depth 1. Through some experimentation, I realized each child after that can be said to create the same number of children. That means that at a depth of 5, a magnitude of hundreds of millions of nodes need to be calculated using this search strategy. That’s why my computer had such a hard time running it.

TEST CASE 1

Country	Population	Metallic Elements	Timber	Metallic Alloys	Electronics	Housing	Metallic AlloyWaste	Electronic Waste	Housing Waste
Damon	10	70	20	0	0	0	0	0	0
Brobdingnag	5	30	12	0	0	0	0	0	0
Carpania	25	10	30	0	0	0	0	0	0
Dinotopia	3	20	20	0	0	0	0	0	0
Erewhon	7	50	17	0	0	0	0	0	0

Resource	Weight
Population	0
MetallicElements	0.56
Timber	0.16
MetallicAlloys	0.31
Electronics	0
Housing	0.06
MetallicAlloyWaste	-0.03
ElectronicWaste	-0.03
HousingWaste	-0.03

Parameter Settings	
Your_country_name	Damon
Resource_filename	resource_weights
Initial_state_filename	initial_world_state
Output_schedule_filename	output_scheduler
Num_output_schedules	5
Depth_bound	3
Frontier_max_size	N/A
Discounted Reward	Y = .5
Probability of Success	L = 1
	K = 1
	X0 = 0
Expected Utility	C = -.5

```
[
Depth = 3 EU = 6.993058423713505 (transfer  Dinotopia  Damon ((MetallicElements  20)))
Depth = 2 EU = 11.199830285348822 (transfer  Brobdingnag  Damon ((MetallicElements  30)))
Depth = 1 EU = 13.999987942843598 (transfer  Erewhon  Damon ((MetallicElements  50)))
ROOT NODE

Depth = 3 EU = 6.992745400546924 (transfer  Brobdingnag  Damon ((MetallicElements  30)))
Depth = 2 EU = 9.799420315931572 (transfer  Dinotopia  Damon ((MetallicElements  20)))
Depth = 1 EU = 13.999987942843598 (transfer  Erewhon  Damon ((MetallicElements  50)))
ROOT NODE

Depth = 3 EU = 6.991380087812026 (transfer  Dinotopia  Damon ((MetallicElements  20)))
Depth = 2 EU = 11.197209693849498 (transfer  Erewhon  Damon ((MetallicElements  50)))
Depth = 1 EU = 8.397999130744925 (transfer  Brobdingnag  Damon ((MetallicElements  30)))
ROOT NODE

Depth = 3 EU = 6.984657402308629 (transfer  Erewhon  Damon ((MetallicElements  50)))
Depth = 2 EU = 6.991482526412149 (transfer  Dinotopia  Damon ((MetallicElements  20)))
Depth = 1 EU = 8.397999130744925 (transfer  Brobdingnag  Damon ((MetallicElements  30)))
ROOT NODE

Depth = 3 EU = 6.96514653646449 (transfer  Brobdingnag  Damon ((MetallicElements  30)))
Depth = 2 EU = 9.761483313373537 (transfer  Erewhon  Damon ((MetallicElements  50)))
Depth = 1 EU = 5.577526136613443 (transfer  Dinotopia  Damon ((MetallicElements  20)))
ROOT NODE
]
```

TEST CASE 2

Country	Population	Metallic Elements	Timber	Metallic Alloys	Electronics	Housing	Metallic AlloyWaste	Electronic Waste	Housing Waste
Damon	10	70	20	0	0	0	0	0	0
Brobdingnag	5	30	12	0	0	0	0	0	0
Carpania	25	10	30	0	0	0	0	0	0
Dinotopia	3	20	20	0	0	0	0	0	0
Erewhon	7	50	17	0	0	0	0	0	0

Resource	Weight
Population	0
MetallicElements	0.56
Timber	0.16
MetallicAlloys	0.31
Electronics	0
Housing	0.06
MetallicAlloyWaste	-0.03
ElectronicWaste	-0.03
HousingWaste	-0.03

Parameter Settings	
Your_country_name	Damon
Resource_filename	resource_weights
Initial_state_filename	initial_world_state
Output_schedule_filename	output_scheduler
Num_output_schedules	5
Depth_bound	3
Frontier_max_size	N/A
Discounted Reward	Y = .7
Probability of Success	L = 1
	K = 1
	X0 = 0
Expected Utility	C = -.5

[

Depth = 3 EU = 19.20799984394863 (transfer Dinotopia Damon ((MetallicElements 20)))
Depth = 2 EU = 21.951999924391934 (transfer Brobdingnag Damon ((MetallicElements 30)))
Depth = 1 EU = 19.599999938194912 (transfer Erewhon Damon ((MetallicElements 50)))
ROOT NODE

Depth = 3 EU = 19.2079997600326 (transfer Brobdingnag Damon ((MetallicElements 30)))
Depth = 2 EU = 19.207999849716433 (transfer Dinotopia Damon ((MetallicElements 20)))
Depth = 1 EU = 19.599999938194912 (transfer Erewhon Damon ((MetallicElements 50)))
ROOT NODE

Depth = 3 EU = 19.20784597001762 (transfer Dinotopia Damon ((MetallicElements 20)))
Depth = 2 EU = 21.951824626162033 (transfer Erewhon Damon ((MetallicElements 50)))
Depth = 1 EU = 11.75990424003673 (transfer Brobdingnag Damon ((MetallicElements 30)))
ROOT NODE

Depth = 3 EU = 19.207824292834292 (transfer Erewhon Damon ((MetallicElements 50)))
Depth = 2 EU = 13.719873285944612 (transfer Dinotopia Damon ((MetallicElements 20)))
Depth = 1 EU = 11.75990424003673 (transfer Brobdingnag Damon ((MetallicElements 30)))
ROOT NODE

Depth = 3 EU = 19.200244444301276 (transfer Brobdingnag Damon ((MetallicElements 30)))
Depth = 2 EU = 19.200244533949814 (transfer Erewhon Damon ((MetallicElements 50)))
Depth = 1 EU = 7.836718092186438 (transfer Dinotopia Damon ((MetallicElements 20)))
ROOT NODE

]

TEST CASE 3

Country	Population	Metallic Elements	Timber	Metallic Alloys	Electronics	Housing	Metallic AlloyWaste	Electronic Waste	Housing Waste
Damon	10	70	20	0	0	0	0	0	0
Brobdingnag	5	30	12	0	0	0	0	0	0
Carpania	25	10	30	0	0	0	0	0	0
Dinotopia	3	20	20	0	0	0	0	0	0
Erewhon	7	50	17	0	0	0	0	0	0

Resource	Weight
Population	0
MetallicElements	0.56
Timber	0.16
MetallicAlloys	0.31
Electronics	0
Housing	0.06
MetallicAlloyWaste	-0.03
ElectronicWaste	-0.03
HousingWaste	-0.03

Parameter Settings	
Your_country_name	Damon
Resource_filename	resource_weights
Initial_state_filename	initial_world_state
Output_schedule_filename	output_scheduler
Num_output_schedules	5
Depth_bound	5
Frontier_max_size	15000
Discounted Reward	Y = .7
Probability of Success	L = 1
	K = 1
	X0 = 0
Expected Utility	C = -.2

COMPUTATIONAL ISSUE (CPU freeze up)

COMPUTATIONAL ISSUE

(CPU freeze up)

N = Number of children for 1 Node
D = Search Depth
K = Depth Level

$$\frac{N^{(D+1)} - 1}{(N-1)} = \text{Nodes to compute}$$

$$N^{K} = \text{Nodes at “K” Depth}$$

$$\frac{58^{(5+1)} - 1}{(58-1)} = 667{,}871{,}799 = \text{Nodes to compute}$$

$$58^{5} = 656{,}356{,}768 = \text{Nodes in Solution}$$