CS486 A4

Charlie Wu

Q1.1

U_1	1	2	3	4
1	-0.05	-0.05	-0.05	-0.05
2	-0.05	X	-0.05	-1
3	-0.05	-0.05	0.75	+1

	1	2	3	4
1	↓,↑,←,→	↓,↑,←,→	↓,↑,←,→	↑
2	↓,↑,←,→	X	←	-1
3	↓,↑,←,→	↓,↑,←,→	→	+1

U_2	1	2	3	4
1	-0.1	-0.1	-0.1	-0.1
2	-0.1	X	0.445	-1
3	-0.1	0.54	0.82	+1

	1	2	3	4
1	↓,↑,←,→	↓,↑,←,→	↓,↑,←,→	↑
2	↓,↑,←,→	X	↓	-1
3	↓,↑,←,→	→	→	+1

U_3	1	2	3	4
1	-0.15	-0.15	0.286	-0.15
2	-0.15	X	0.5505	-1
3	0.362	0.714	0.8765	+1

	1	2	3	4
1	↓,↑,←,→	↓,↑,←,→	↓	↑
2	↓,↑,←,→	X	↓	-1
3	→	→	→	+1

U_4	1	2	3	4
1	-0.2	0.1488	0.3604	0.0638
2	0.2096	X	0.60625	-1
3	0.5424	0.794	0.8927	+1

	1	2	3	4
1	↓,↑,←,→	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

U_5	1	2	3	4
1	0.11256	0.26808	0.45626	0.1447
2	0.42584	X	0.624785	-1
3	0.6604	0.82296	0.899895	+1

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

U_6	1	2	3	4
1	0.328736	0.368624	0.491106	0.229478
2	0.563488	X	0.632394	-1
3	0.716992	0.834508	0.902468	+1

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

U_7	1	2	3	4
1	0.470526	0.41661	0.555726	0.265833
2	0.636291	X	0.635214	-1
3	0.745654	0.838876	0.903486	+1

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

	1	2	3	4
1	0.547947	0.445903	0.526415	0.289164
2	0.673782	X	0.63631	−1
3	0.759295	0.840564	0.90387	+1

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	−1
3	→	→	→	+1

U_8

	1	2	3	4
1	0.58839	0.477378	0.532555	0.300049
2	0.692193	X	0.636729	−1
3	0.765759	0.841209	0.904018	+1

	1	2	3	4
1	↓	←	↓	←
2	↓	X	↓	−1
3	→	→	→	+1

U_9

	1	2	3	4
1	0.610331	0.516188	0.537124	0.306049
2	0.701046	X	0.636887	−1
3	0.768762	0.841456	0.904075	+1

	1	2	3	4
1	↓	←	↓	←
2	↓	X	↓	−1
3	→	→	→	+1

U_{10}

Please see the above for the true utility $U(S)$ for each state S and the optimal policy for each state. The left diagram for true utility value and right for optimal policy. If the optimal policy changes from one iteration to the next iteration, it is in orange with circle (ex: ↑). The calculation for $U_3(S_{13})$ is the following.

$$U_3(S_{13}) = -0.05 + \max[(0.8)(0.1)+(0.1)(-0.1)+(0.1)(-0.1), (0.8)(-0.1)+(0.1)(-0.1)+(0.1)(0.445), (0.8)(0.445)+(0.1)(-0.1)+(0.1)(-0.1), (0.8)(-0.1)+(0.1)(-0.1)+(0.1)(0.445)]$$

$$= -0.05 + \max[-0.1, -0.0455, 0.336, -0.0455] = -0.05 + 0.376 = 0.286.$$

Q1.2

	1	2	3	4
1	−0.1	−0.1	−0.1	−0.1
2	−0.1	X	−0.1	−1
3	−0.1	−0.1	0.7	+1

	1	2	3	4
1	↓,↑,←,→	↓,↑,←,→	↓,↑,←,→	↑
2	↓,↑,←,→	X	←	−1
3	↓,↑,←,→	↓,↑,←,→	→	+1

U_1

	1	2	3	4
1	−0.2	−0.2	−0.2	−0.2
2	−0.2	X	0.35	−1
3	−0.2	0.44	0.76	+1

	1	2	3	4
1	↓,↑,←,→	↓,↑,←,→	↓,↑,←,→	↑
2	↓,↑,←,→	X	↓	−1
3	↓,↑,←,→	→	→	+1

U_2

CS486 A4 [illegible]

U_3

	1	2	3	4
1	-0.3	-0.3	0.14	-0.3
2	-0.3	X	0.443	-1
3	0.212	0.596	0.811	+1

	1	2	3	4
1	↓,↑,←,→	↓,↑,←,→	↓	↑
2	↓,↑,←,→	X	↓	-1
3	→	→	→	+1

U_4

	1	2	3	4
1	-0.4	-0.048	0.1944	-0.118
2	0.0096	X	0.4931	-1
3	0.368	0.668	0.8254	+1

	1	2	3	4
1	↓,↑,←,→	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

U_5

	1	2	3	4
1	-0.13712	0.04592	0.27788	-0.05628
2	0.19632	X	0.50963	-1
3	0.47216	0.69392	0.83185	+1

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

U_6

	1	2	3	4
1	0.047936	0.131488	0.306668	0.016676
2	0.316992	X	0.516443	-1
3	0.521984	0.704264	0.834148	+1

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

U_7

	1	2	3	4
1	0.171536	0.171632	0.327971	0.047002
2	0.380986	X	0.518963	-1
3	0.547309	0.708171	0.835059	+1

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

U_8

	1	2	3	4
1	0.239105	0.196703	0.337034	0.0670768
2	0.414044	X	0.519944	-1
3	0.559366	0.709682	0.835402	+1

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

U_9

	1	2	3	4
1	0.274816	0.208967	0.342333	0.0763345
2	0.430302	X	0.520316	-1
3	0.565086	0.710258	0.835535	+1

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

	1	2	3	4
1	0.29262	0.21566	0.344783	0.0814997
2	0.438129	X	0.520459	-1
3	0.567745	0.710499	0.835585	+1

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

U_{10}

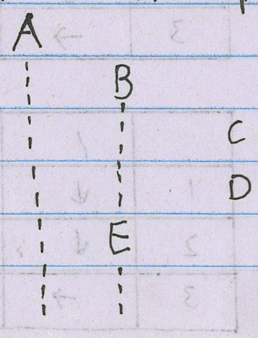
Please see the above for the true utility $U_i(S)$ for each state S and the optimal policy for each state. The left diagram for true utility value and right for optimal policy. If the optimal policy changes from one iteration to the next iteration, it is in orange with circle (ex: ↑). The calculation for $U_3(S_{13})$ is the following

$$U_3(S_{13}) = -0.1 + \max[(0.8)(-0.2) + (0.1)(-0.2) + (0.1)(-0.2), (0.8)(-0.2) + (0.1)(-0.2) + (0.1)(0.35),$$
$$(0.8)(0.35) + (0.1)(-0.2) + (0.1)(-0.2), (0.8)(-0.2) + (0.1)(-0.2) + (0.1)(0.35)]$$
$$= -0.1 + \max[-0.2, -0.145, 0.24, -0.145] = -0.1 + 0.24 = 0.14$$

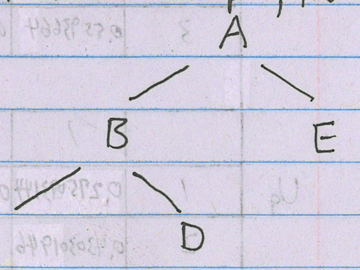
Q1.3 The state that the optimal policy is different is state S_{12}. With $R(S) = -0.05$, the cost for each step is small enough that if we start from the state S_{12}, we still prefer to go the long way to avoid the higher possibility of ending up in the -1 end state. However with $R(S) = -0.1$, the cost for each step is too big to take the long way from state S_{12}. Thus, only the state S_{12} has a different optimal policy - one going the long way and one going the short way.

Q1.4 Please refer to the end of this assignment. It will be at the end if it is done.

Q2.1 Please build id3_algo.cpp using gcc (g++ id3_algo.cpp) and run the output program (a.out). The program will then ask you to input the file for question 2.3. Please simply enter N and the program will try to read set_b.csv for question 2.3. Otherwise, please input the file name to use for question 2.3. The program, before the prompt, will also generate a decision tree with set_a.csv data and print the tree out. The tree will be printed in the format that every level is recognized by the number of index followed by the node. The left node will be printed first then the right node. For example, the following:

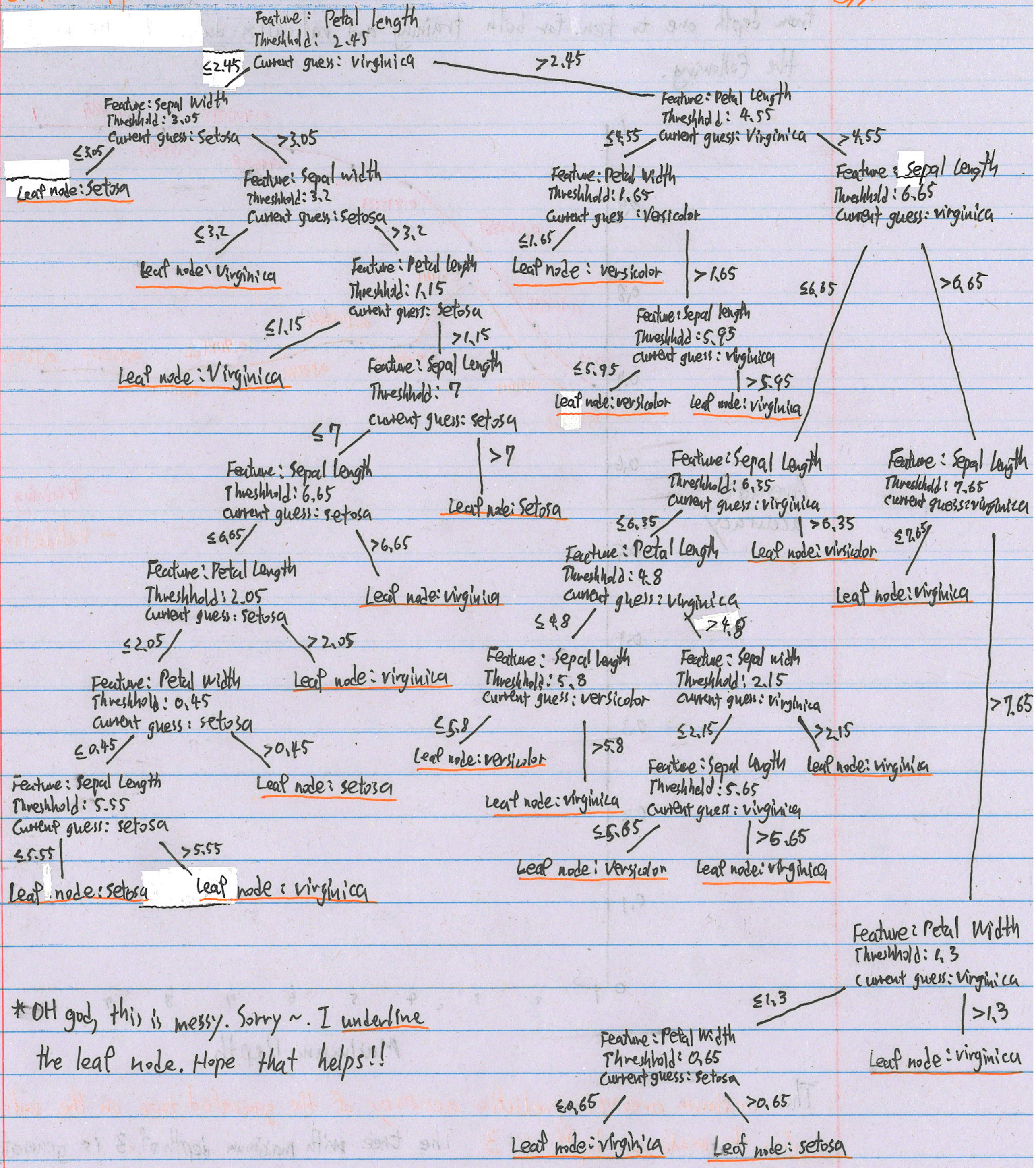


Translate to



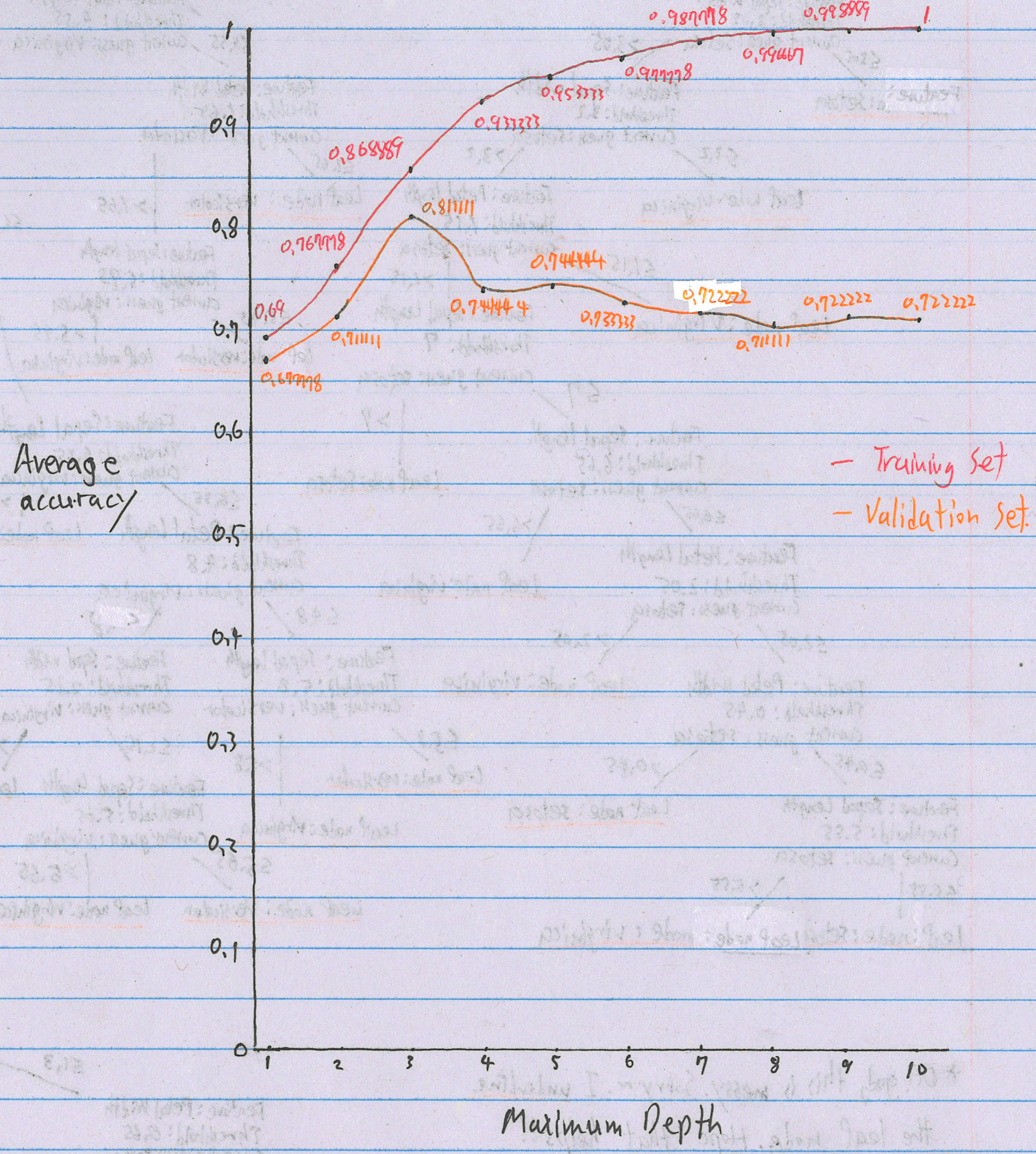
* The tree is on next page. Each node will show the feature, threshold, and the closest guess of the decision. Otherwise, decision only if it a leaf.

CS 486 A4 Charlie Uhu

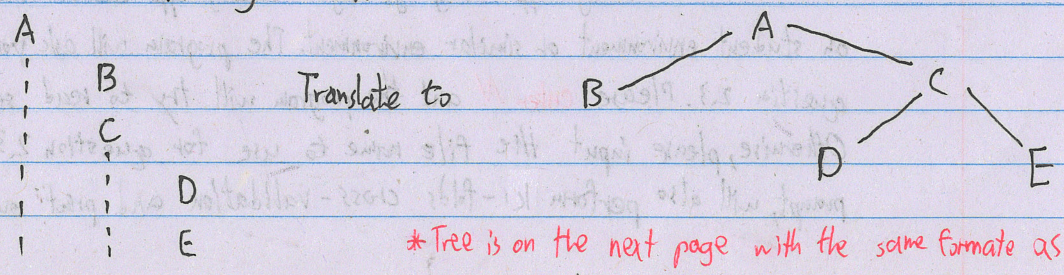


*OH god, this is messy. Sorry ~. I underline the leaf node. Hope that helps!!

Q2.2 Please build id3_algo.cpp using gcc (g++ id3_algo.cpp) and run the output program (./a.out) on student environment or similar environment. The program will ask you to input the file for question 2.3. Please enter N and the program will try to read set_6.csv for question 2.3. Otherwise, please input the file name to use for question 2.3. The program, before the prompt, will also perform 10-folds cross-validation and print out the average accuracy from depth one to ten for both training and validation data set. The result is plotted as the following.

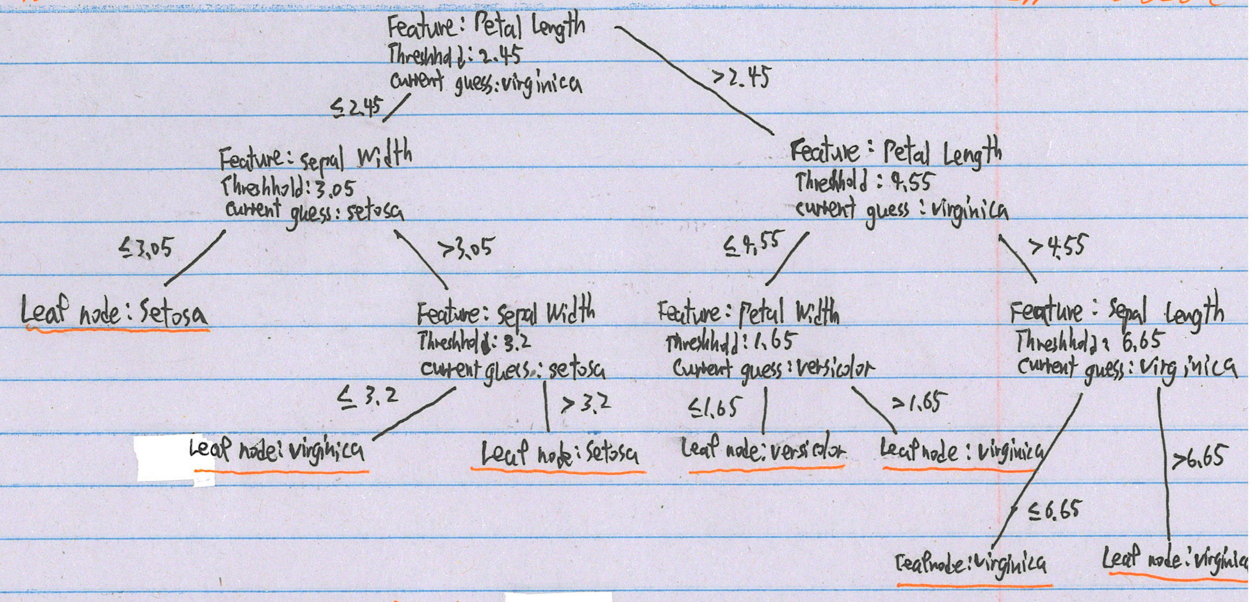


The maximum average prediction accuracy of the generated tree on the validation set is when the maximum depth is 3. The tree with maximum depth of 3 is generated with the whole data set a is on the next page. Again, when running the program, the tree is printed out in the following format.



*Tree is on the next page with the same formate as Q2.1

CS 486 A4 Charli [illegible]



Again, I under line the leaf node.

The decision at the leaf node is made based on majority and any tie is break randomly.

The prediction accuracy is 0.87.

Q3 Again, please build id3_algo.cpp and run the program as mention in Q 2.1 and Q2.2. Please make sure you enter the correct file name or N which the program will use set_b.csv as input file. The tree will be generated on background (it will not get printed) and the accuracy will be printed out.

Q1.4 Bonus Continue on the question 1.1 until the iteration 20 (Please build and run id3_algo.cpp which now also do the iteration for question 1.1 and 1.2 for 20 iteration), we get the following

	1	2	3	4
1	0.638459	0.575911	0.549326	0.321594
2	0.708791	X	0.636986	-1
3	0.771296	0.84161	0.90411	+1

	1	2	3	4
1	↓	←	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

Compare the above with the course note, the optimal policy differ in state S_{13}. With $R(S) = -0.04$ as in course note, the policy prefer the longer route, where the above prefer the shorter but riskier route. Thus, with $R(S) = -0.05$, it is just too expensive to take the longer route from state S_{13}.