CS7637: Knowledge-Based AI:
RPM Project: Milestone 3

Jing Gan
jgan34@gatech.edu

1 QUESTION 1

How does your agent currently function?

More examples are given in Question 3 to elaborate on methods taken by my agent.

Directions: In a 3 × 3 problem, my agent mainly considers horizontal and vertical transformations. It does not examine the diagonal relation directly as I think it could be bypassed and compensated by relations in another two directions. Horizontally, the agent will check either 3 figures in 3 ways: 1) 2 figures as a pair – (A, B), (B, C), or 2) 3 figures as a whole – (A, B, C), or 3) addition and subtraction – A+B=C, A+C=B, B+C=A, etc. Similar patterns are checked vertically as well.

Values: The agent mainly calculates multiple values including DPR (Dark Pixel Ratio), numbers of black (dark) pixels, and numbers of white pixels. In this assignment, I do not apply the IPR (Intersection Pixel Ratio) as my agent is already able to score 100/100 on Gradescope.

Methods: My agent mainly applies direct pixel comparisons. I regard DPR as a method of this type, too. There are 4 main methods: 1) just like Basic Problem C-01 where in each row the 3 figures are the same. The agent will compare the number of dark pixels pairwise to see if the figure remains the same and find an answer for Figure 9; 2) the number of dark elements in Row 2 is n times of Row 1, and the number in Row 3 is m times of Row 2. Thus, when the multipliers are found, by calculating and comparing the number of dark pixels the agent can find the answer for Figure 9; 3) each row contains 3 same figures but in a different order. In this case, the agent will calculate DPR for each row and each figure. If DPRs in Row 2 also show up in Row 1, the agent knows that Figure 9 needs a choice that has the missing DPR in Row 3. To deal with the noise, the agent allows a +/- 5% variation in ratios. Again, all such relations are checked vertically as well.

2 QUESTION 2

How well does your agent currently perform? How many problems does it get right on the Set C problems?

According to the autograder on Gradescope, my agent earns 100/100 points.

In terms of all four sets of questions, my agent's performance is shown below:

Table 1 — Milestone 3 Gradescope Result

	Basic - C	**Test - C**	**Challenge - C**	**Raven's - C**
Correct	8	6	6	5
Incorrect	4	6	6	7

As shown in Table 1, although my agent managed to earn full points, it needs more improvements. I believe that after adding more details or algorithms to my agent it would be able to perform better on all 4 question sets.

3 QUESTION 3

3.1 What problems does your agent perform well on?

For Set C, my agent performs well on questions that involve the addition and subtraction of certain patterns such as Basic Problem C-05, Basic Problem C-06, Basic Problem C-11, Challenge Problem C-02, etc.

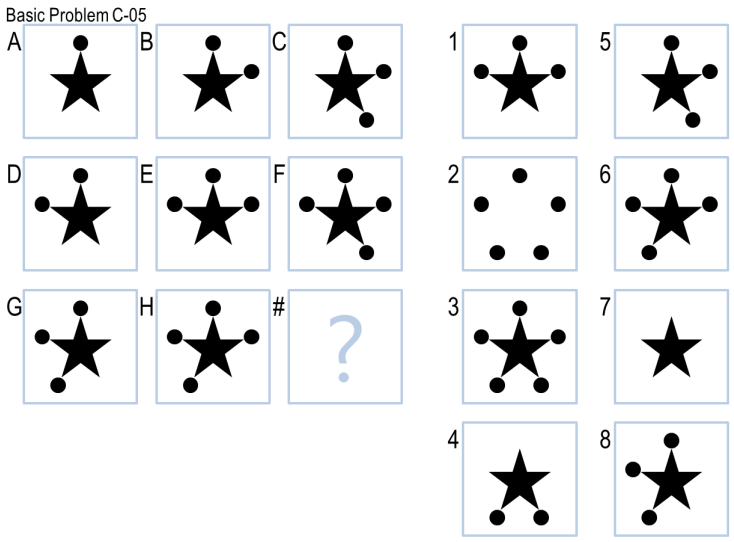

Figure 1— Basic Problem C-05

For example, in Basic Problem C-05 above, from A to C and D to F, each time a black dot is added. Therefore, by subtracting B from C we get a black dot which is the same as subtracting A from B, D from E, E from F, and G from H. Therefore, by subtracting H from each choice, if the result is of the same number of black pixels, the agent can correctly return the answer.

My agent also performs well on questions involving the same 3 figures but in different (or same) orders in each row, such as Basic Problem C-01, Challenge Problem C-01, etc.

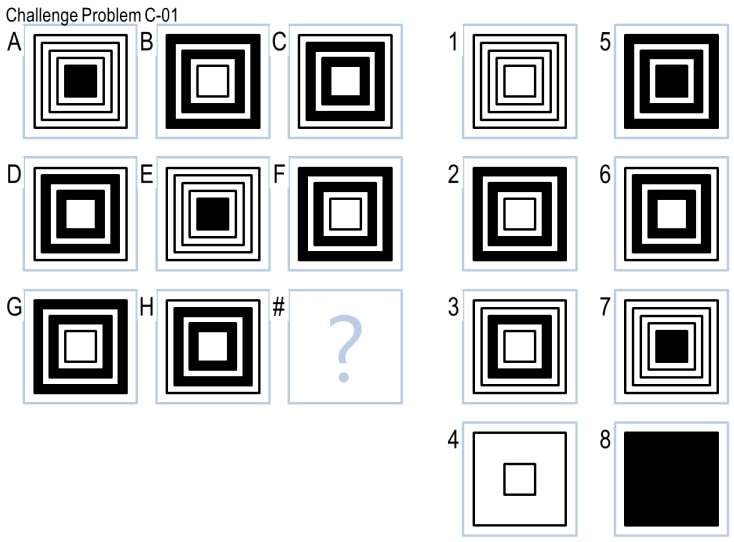

Figure 2— **Challenge Problem C-01**

For example, in Challenge Problem C-01 above, each row contains the same 3 figures but in different orders. Therefore, the agent can use the DPR (Dark Pixel Ratio) of each figure to detect such patterns. In other words, if the agent finds that DPRs in Row 2 are the same as those in Row 1 (regardless of the order), it knows that Row 3 should have the same DPRs. Once the agent excludes DPRs of G and H, it uses the remaining DPR to test each choice and find the answer. Since the DPRs are not the exact same, I use a +/- 5% lower and upper bound to allow any small differences. For example, if the DPR of E is within the range of (DPR_A * 0.95, DPR_A * 1.05), they are the same.

My agent also performs well on questions involving the multiplication of elements, such as Challenge Problem C-07, Challenge Problem C-11, etc.

For example, in Challenge Problem C-11 below, the number of black elements in Row 2 is 2 times of those in Row 1, and the number in Row 3 is 3 times of Row 1. This time, my agent calculates the number of black pixels in each figure and compares the figures horizontally and vertically. If it finds the vertical multipliers the same, it calculates the expected number of black pixels in the last figure based on Figure C. Then, it compares the number to all choices and finds the answer is 4.

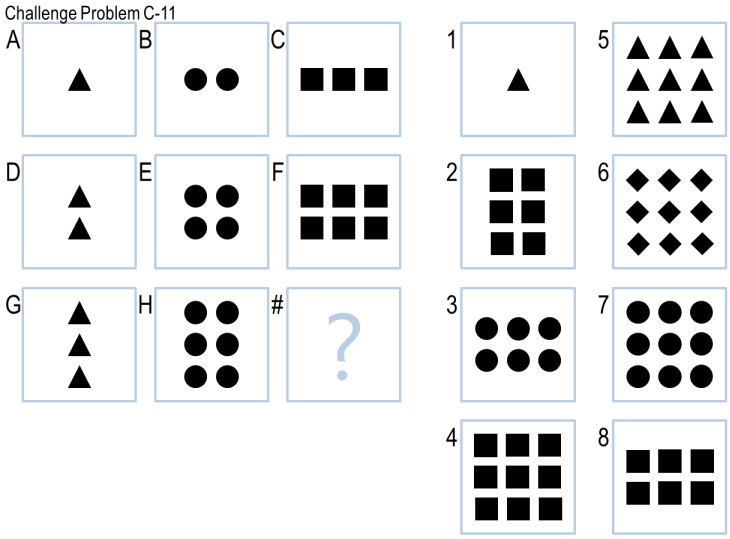

Figure 3— Challenge Problem C-11

3.2 What problems (if any) does it struggle on? Why does it struggle?

It struggles with problems such as Basic Problem C-09, Basic Problem C-10, etc.

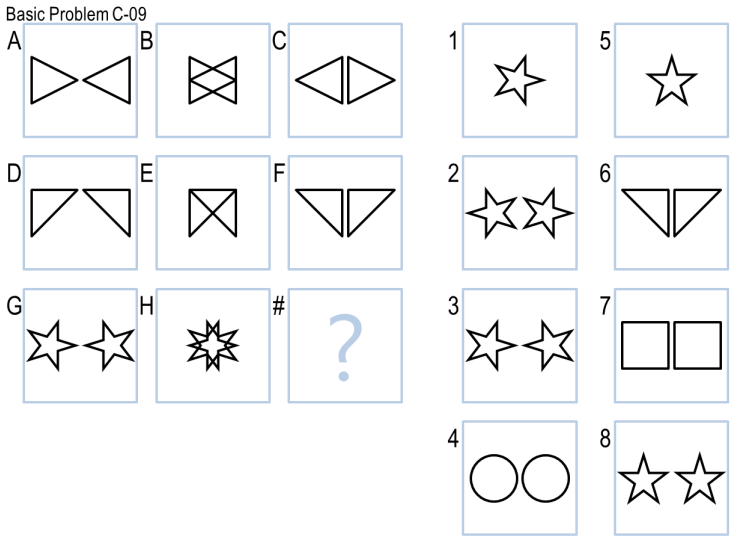

Figure 4— **Basic Problem C-10**

In such questions, although the transformations look straightforward to humans, they are hard for the agent to detect as they move elements within the figure continuously. At this stage, I fail to come up with an algorithm to deal with such

patterns. My agent cannot replicate such movements (transformations) and generalize a specific rule to be applied on the last row. And this is what I will be working on in future assignments.

4 QUESTION 4

How efficient is your agent? Does it take a long time to run? Does it slow down significantly on certain kinds of problems?

Since I have limited knowledge of Big O notation, I will describe the efficiency in terms of correctness and running time.

It is very efficient in terms of solving questions in a short time. When I test my agent locally, it can provide answers in a few seconds. And on Gradescope when it is faced with more questions of more complex transformations, the running time increases obviously. However, there is no timeout warning.

Although I have no access to the test questions on Gradescope, I expect that as mentioned it struggles with questions involving frames, in-figure movements, and intersections. And as I wrote in Milestone report 2, "there may be other problems that involve even more complex transformations such as filling the frames that my agent can't solve."

5 QUESTION 5

How do you plan to improve your agent's performance on these problems before the final project submission?

The plan is simple. I want to focus on the questions that my agent fails as shown in Table 1. I need to polish my DPR method and consider introducing the IPR (Intersection Pixel Ratio) method. I also need to consider the PIL package as I only use OpenCV this time. I may find other useful tools in PIL.

What's more, I will closely follow Ed Discussion and learn from the ideas of my classmates!

6 QUESTION 6

Looking ahead to Sets D and E, which problems do you think your agent will be able to solve at present? Which problems will it struggle with?

At present, since the transformations are similar, my agent should be able to solve questions such as Basic Problem D-01, Basic Problem D-02, Basic Problem D-03, Basic Problem E-02, etc.

And it will struggle with questions such as Basic Problem D-05, Basic Problem D-07, etc. The reason is that multiple transformations, such as frames, zooming, and fillings, take place at the same time:

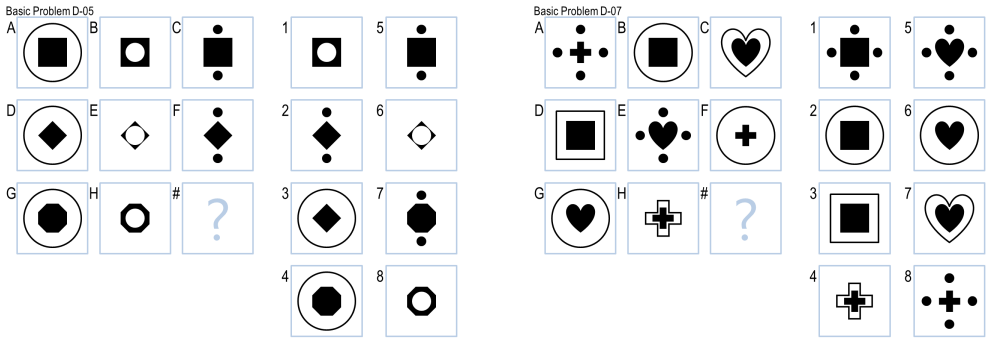

Figure 5 — Basic Problem D-05, Basic Problem D-07

7 QUESTION 7

What feedback would you hope to get from classmates about how your agent could do better? What challenges do you think could benefit from someone else's feedback?

As I mentioned in Milestone 2 report, "in fact, any suggestions and feedback are greatly welcomed. If you have more ideas on dealing with frames problems, please let me know!"