RPM Final Project: The Little Agent That Could

Shahzad Rizvi
srizvi43@gatech.edu

Abstract—Raven's Progressive Model (RPM) is a type of a nonverbal IQ test used to gauge abstract thinking and fluid intelligence, which is related to comprehension, problem solving, and learning[1]. In this report, we will look at an AI Agent that decomposes the images in the question and using a combination of Affine and Set Transformation Induction (ASTI)[2], Dark Pixel Ratio and custom build heuristics to determine an optimal solution.

1 HOW THE AGENT WORKS

1.1 Introduction

The previous generations of solutions for RPM focused on atomic rules to be applied to the problems and used heuristic reasoning to address the problem. In this agent, a combination of those heuristic reasoning were coupled with image decompositions to represent the problem and drive the solution. The 5 main rules that Carpenter et al, 1990 used to address the problem that were utilized in this agent were:

1. Row/Column consistency
2. Triplet division
3. Pairwise progression
4. Figures logically and together
5. Distribution of a pair of values

Recent iterations of the approach to the RPM problem added an affine component to resolving the problem. This method uses the more modern libraries to decompose the images and perform operations on individual images. Shegheva et al, 2018 identified an important common thread in these approaches: they decompose the problem images and apply specific heuristics to solve them. This agent takes these threads and weaves it into a (semi) effective solution.

1.2 Methodology

The fundamental challenge this agent tries to address is in identifying a viable relationship in the limited data presented to it. The approach that was taken in build-

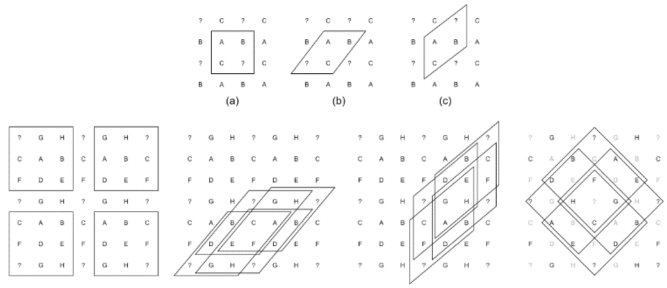


Figure 1 – Combinatory sets by varying dimensionality (Yuan Yang, 2020) (larger image in appendix)

ing this agent out was based on utilizing the 5 fundamentals affine transformations and using the combinatorial sets proposed by Yuan Yang, 2020.

Fig 1 identifies the primary sets used to generate potential relationships in the images. By utilizing these sets for the 2x2 and 3x3 problems, the agent was successfully able to identify the primary relationships.

The summary viewpoint of the agent is identified in Equation 1:

$$\exists\, x \in solutions, \varphi \in \text{heuristics}\,,\ x \equiv \sum_{i=0}^{\text{matrix size}} \varphi\left(A_i, B_{i,}, C_{i,}\right) \omega \rightarrow \mathbb{Q} \qquad \textit{Equation 1}$$

In the agent, φ is the unique heuristics that were used based on the 5 fundamental affine operations adjusted by ω weight. The agent decomposes the images in the problem set and based on the complexity of the problems, analysis the combinations and determines a most probable solution into a rational number.

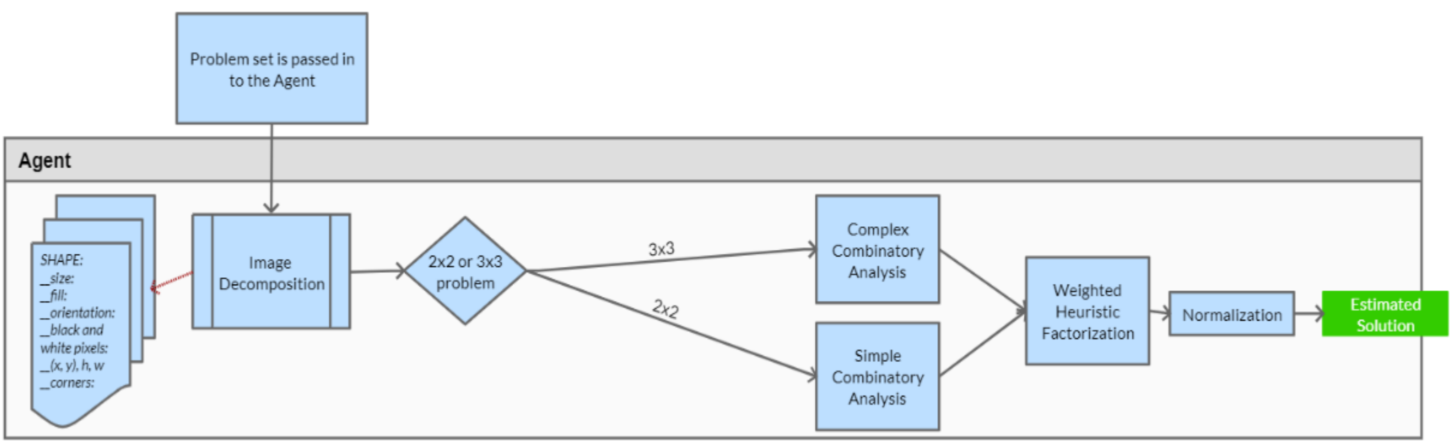


Figure 2 Agent Algorithm Design

1.2.1 *Image Decomposition*

The first major component of the agent takes in the images and decomposes each of the image into a set of values for the following: Image name: Shape, size, filled,

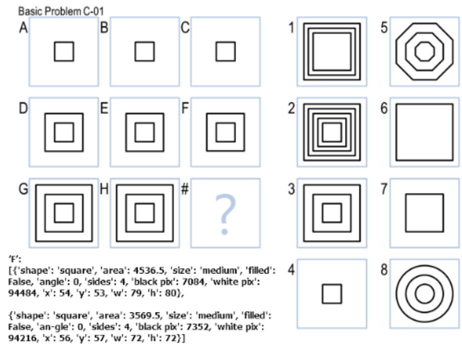


Figure 3 – Example of image decomposition from Basic Problem C-01 (larger image in appendix)

orientation, black and white pixels, (x, y) coordinates with height and width, as seen in Figure 3. This set of datapoints allows the agent to apply the heuristics according to the combinatory analysis picked.

1.2.2 *Simple Combinatory Analysis (SCA)*

For the 2x2 problems, the relationships were simpler comparatively to the 3x3 problem space. Also, as the information the agent must solve the problem with is significantly (66%) less compared to the 3x3 problems In Figure 1, images (a), (b) and (c) were the set of combinations the agent uses in determining the most accurate connections.

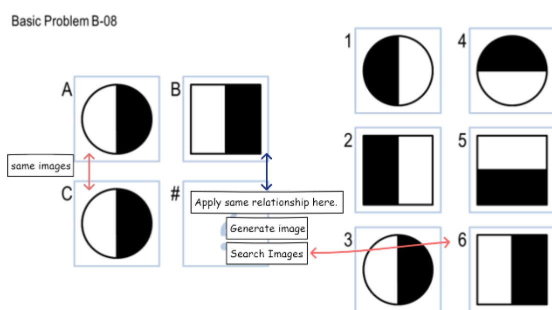


Figure 4- 2x2 matrix with (a) combination, using similar image heuristic

Figure 4 exemplifies how the images A and C would be recognized as similar and the agent would 'generate' (which is the same image as image B in this case) and test against the solution images.

1.2.3 *Complex Combinatory Analysis (CCA)*

For the 3x3 problems, the agent uses the other combinations to determine relationships. Figure 5 shows an application of the combination sets used for Basic Problem

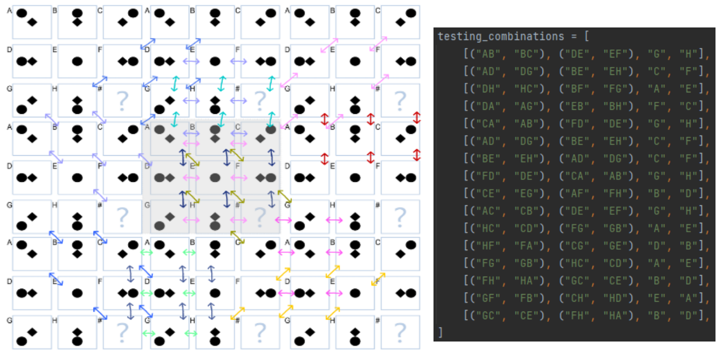


Figure 5- Combinatory pattern and list used in 3x3 problems (larger image in appendix)

C-07. Using the basic problem space of A-F (the greyed area in Figure 5), there were 5 directions of checks applied to 16 different sets (as shown in Figure 1). The duplicate test sets were reduced. This allowed the agent to check for expanded relationships.

1.2.4 *Heuristic Factorization*

The agent takes the decomposed images and iterates through the combination sets and creates the sum in Equation 1. This sum is based on the heuristics in described in Table 1.

Table 1- Heuristic Descriptions

Heuristic Name	Description
Same Images	Check if images in the problem are all the same
Same Row	Check if images in the problem are all the same in the row
Same Column	Check if images in the problem are all the same in the column
Rotated	Are the images rotated? Rotates the images 0° to 360° by 45° increments
Flipped Upside down	Are images flipped horizontally?
Flipped sideways	Are images flipped vertically?
Dark pixel ratio comparison	Are the black pixels increasing in a consistent ratio?
Image sum	Add the image pixels on top of one another to create a sum of the images
Image difference	Remove pixels from one image based on another image
XOR	Perform logical XOR to check this relationship in the combination sets
AND	Perform logical XOR to check this relationship in the combination sets
Corner comparisons	Find edges and corners and identify a relationship in corner changes
Size growth	Are images growing/shrinking across the combination sets?
Sudoku approach[6]	Is any of the solution images already in the problem set?

Based on the data set provided, these heuristics created a sum of probabilities, that were weighted to produce the probabilities for all the solution images. Figure 2 summarizes all the components and how they are interconnected to estimate a solution. This algorithm of the agent shows how it uses a combination of generate/test and means end analysis to drive the probabilities.

1.2.5 *Similarity Metrics*

One additional component the agent uses in the algorithm that is of interest is the concept of similarity metrics. It uses a combination of 3 factors to determine how 'similar' two images are. This is one of the main workhorses of the agent as it allows the agent to determine how to interpretate the problem and solution space. The three approaches it takes are:

- Root Mean Square deviation[7]
- ImageChops Image difference – pairwise differences in the images
- Black pixel difference – Sum of all the black pixels in the images/total pixels

2 PERFORMANCE ACROSS ALL SETS OF PROBLEMS

Overall, the agent can successfully pick **88 of the 96** solutions for the Basic and Test problems. The breakdown is as follows:

Problem Set	Score	Problem Set	Score
Basic B	12/12	**Test B**	11/12
Basic C	12/12	**Test C**	9/12
Basic D	12/12	**Test D**	12/12
Basic E	12/12	**Test E**	8/12
Totals	***48/48***		***40/48***
Challenge Set	*28/48*		
Raven's Set	*24*		

3 AGENT SUCCESSFUL

3.1 Problem 1 – Image differences

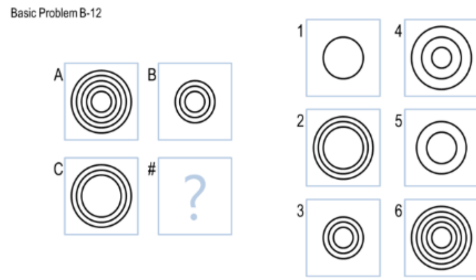


Figure 6 - Basic Problem B-12 successfully solved

One of the problems the agent can successfully solve is Basic B-12. The first step it takes is to decompose the images. In this case, the decomposition is not really needed as using the combinations in Fig 1 (a), the agent is able to identify when applying the heuristic ImageDifference it can determine from image A to B that the image in Figure 7 was removed to get to B. Applying the same difference, it is able to generate an image in Figure 8. In the Weight Heuristic Factorization, it matches the image generated and returns the following probabilities: {**1: 25**, 2: 0, 3: 21.397, 4: 0, 5: 0, 6: 0}

Figure 8 - Generated Image

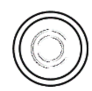


Figure 7- Subtracting factor from Image A in B

Solution image 1 is the correct selection. (The reason image 3 is such a close match is due to the noise of the traces left in the transformations, as can be seen in Figure 8)

3.2 Problem 2 – Complex Combinatory Analysis

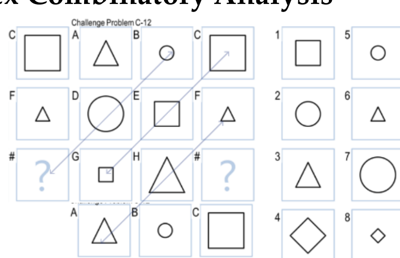


Figure 9 – Successfully solved problem (larger image in appendix)

3.2.1 *Image decomposition*

The agent decomposes the images in Figure 9 to their attributes. The key ones to note are the **shape** and **size**.

3.2.2 *Complex Combinatory Analysis*

In this case, the combination that fruits results is C:E:G::F:H:A ≡ BD#

3.2.3 *Weighted Heuristic Factorization*

The two main heuristics that come into play here are the Size growth and Sudoku approach. With the sudoku approach, the agent returns a negative value for $\varphi(A_i, B_{i,}, C_{i,})$ solutions 1, 3, 5, 6 and 7. The Size growth recognizes the image size is the key in that relationship using the shape decompositions.

3.2.4 *Image Search*

Based on the weighted heuristics, this problem returns the following probability list: {1: 10.74, **2: 31.6356**, 3: 0, 4: 12.8472, 5: 15.3701, 6: 0, 7: 15.5599, 8: 12.8472}

3.3 Problem 3 – Corner Detection

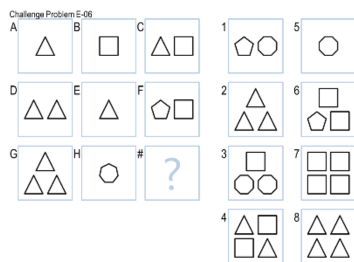


Figure 10 - Problem #3 successfully solved

3.3.1 *Image decomposition*

The agent decomposes the image to its attributes. The key ones to note are the **corners** in each image. The corners returned in the decomposition are {'A': 3, 'B': 4, 'C': 7, 'D': 6, 'E': 3, 'F': 9, 'G': 9, 'H': 7, '1': 13, '2': 9, '3': 20, '4': 14, '5': 8, '6': 13**, '7': 16**, '8': 12}

3.3.2 *Complex Combinatory Analysis*

In this case, the combination that fruits results is A:B:C::D:E:F ≡ GH#

3.3.3 *Weighted Heuristic Factorization*

The main heuristics that come into play here is the Corner comparison. The agent recognizes that the corners are being added.

3.3.4 *Image Search*

Based on the weighted heuristics, this problem returns the following probability list: {1: 5.814, 2: 0, 3: 14.8256, 4: 5.814, 5: 7.5581, 6: 14.8256, **7: 36.3372**, 8: 14.8256}

4 AGENT STRUGGLING

4.1 Problem 1 – The birds

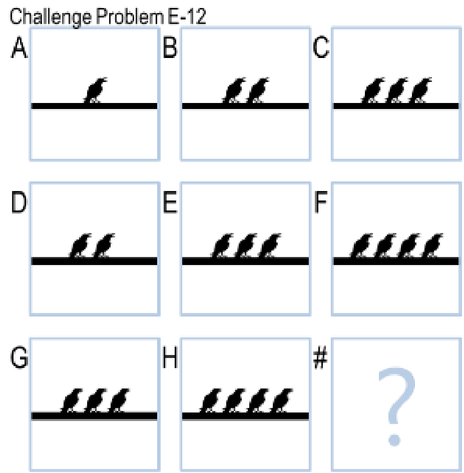


Figure 11 - Challenge problem #1

One of the main problems the agent struggles with is Challenge E12 as seen in Figure 11.

4.1.1 *Image decomposition*

The key area where the agent starts to derail is in the decompositions. One of the primary flaws it has is it is overfitted with analyzing geometric shapes. Images with birds (or with multiple lines used for shading), almost seems to confuse the agent.

4.1.2 *Complex Combinatory Analysis*

In this case, the agent is not able to identify any significant pattern. The surprising aspect is the dark pixel ratio should be able to identify the difference of one bird equivalent from images A:B:C::D:E:F:::G:H:# but it is unable to do so.

4.1.3 *Weighted Heuristic Factorization*

The main heuristics that should come into effect here is the dark pixel ratio increasing by a standard deviation. However, the solution images are too similar and too similarly unusual for the agent to be able to meaningfully reduce the problem using a heuristic.

4.1.4 *Image Search*

Based on the weighted heuristics, this problem returns the following probability list: {1: 16.3959, 2: 16.5286, 3: 21.6245, 4: 16.3836, 5: 12.89, 6: 3.2874, 7: 12.6491, 8: 0.2409}. The key observation here is the lowest probability images are 6 (the bird has flown away) and 8 (the bird is taking off). The other images in the solution are similar enough to each other and the correct solution to 'confuse' the agent.

4.2 Problem 2

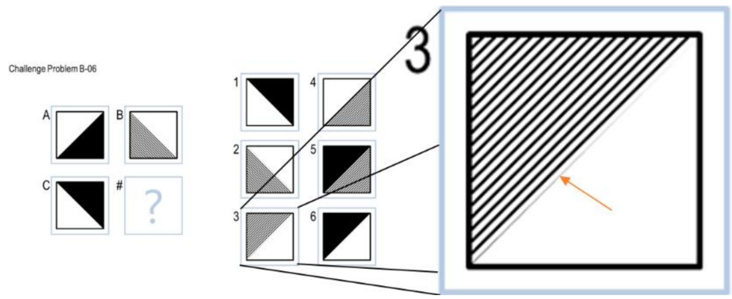


Figure 12 - Challenge Problem #2

The primary issue the agent has with Challenge B06 is the diagonal lines. When the images are processed through the decomposition phase, these lines are interpreted as separate shapes. For example, in Image 3, there is a grey fine line in the diagonal of the square. These nuances allow it to not be recognized as the generated solution. The agent does indeed recognize the simple 90 degree rotational relationship in A:C::B:#, however as that creates a negative result, the agent gravitates towards pixel similarity. As image 2 has the same issue, the agent consistently lands on image 4 with the probability distribution of {1: 0, 2: 93.607, **3: 0**, 4: 0, 5: 0, 6: 6.393}. It does recognize 6 as a close match to the right answer as the shape is like what was expected in the above relationship.

5 APPROACH TO DESIGN

When the RPM project began, the agent used a simplistic version of the heuristics. It was severely limited to just comparing the dark pixel ratios. The design went

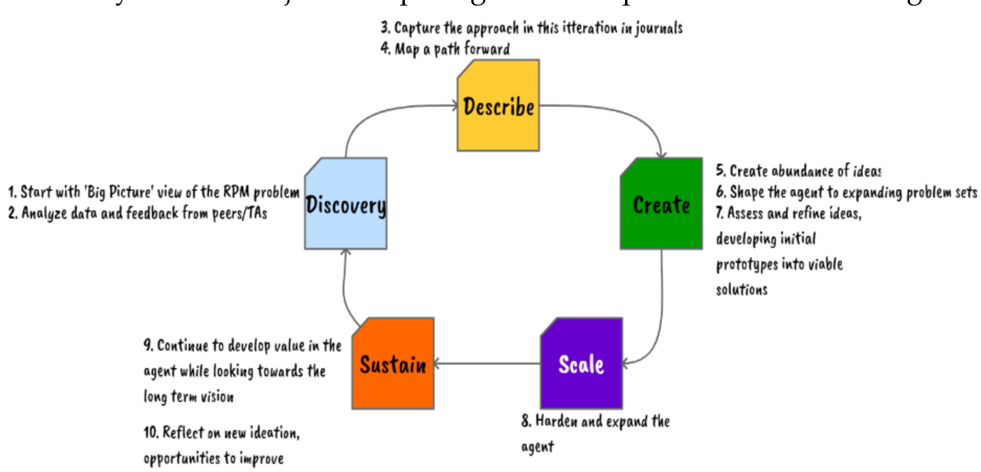


Figure 13- Design paradigm (larger image in appendix)

through a discovery where the framework was explored while looking towards the big picture view.

The Milestone descriptions and feedback from peers allowed the design to incorporate multiple viewpoints and forge a path forward.

The (re)creation of the agent with a backlog of a pool of ideas allowed the agent to be shaped into addressing expanding heuristics and solving additional problem sets. The agent was expanded and refined to scale up to start answering unseen problem sets. With feedback from the Test Sets, it was being sustained by looking towards the long-term vision and incorporating new ideas. For example, one of the last ideas to be incorporated was (x, y) coordinates in the shape decomposition.

6 THE HUMANITY OF THE AGENT

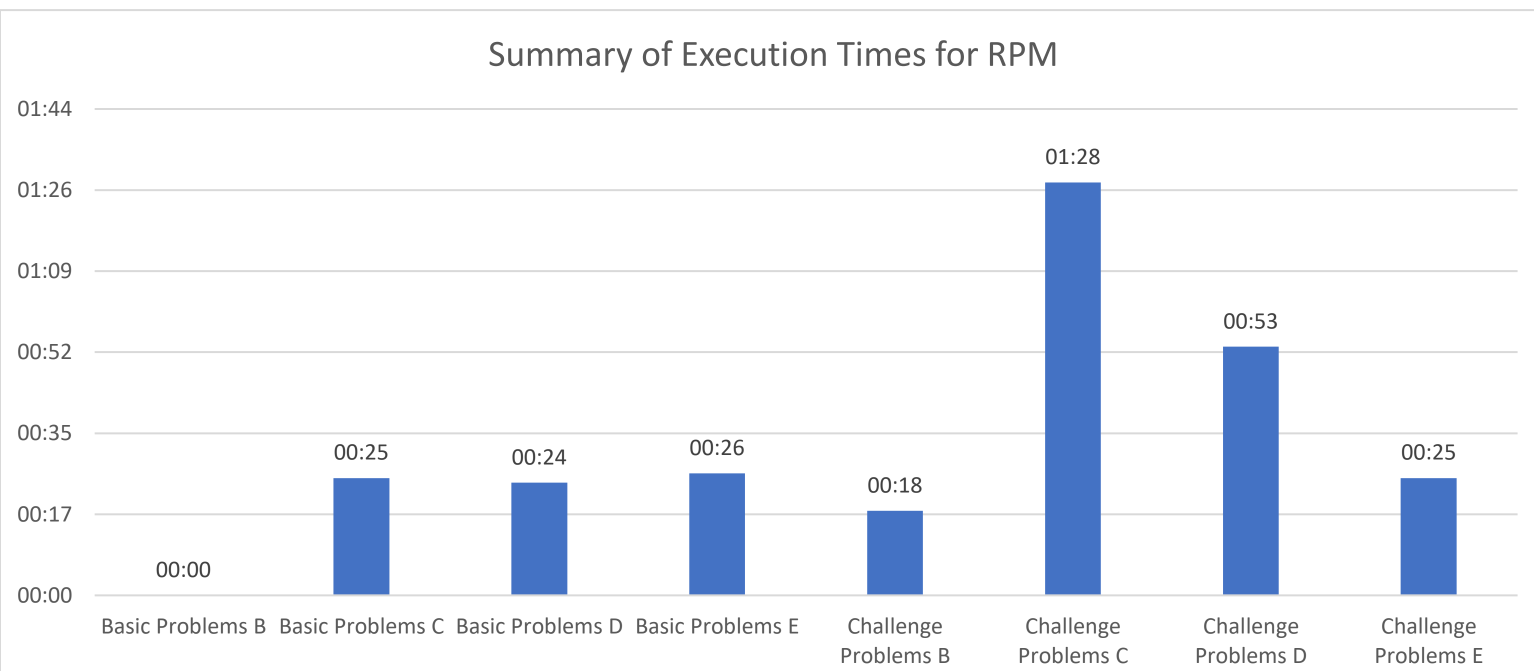
This Agent uses techniques rooted in how a human would search for a solution. A human agent would start the process by mentally assembling analogies row by row and then column by column. An interesting aspect might be the background of the human agent – a native English speaker would start from ABC:DEF:GH# pattern. While a Arabic or Urdu native speaker would mentally be used to reading from right to left and just as likely to start CBA:FED:#HG combinatory pattern. Be that as it may, the human agent would take a 'guess' and try then try to prove it one way or the other – they would plug in the potential solution image in the matrix and try to apply the same relationship(s). The agent tries to take the same approach but using a lot more data points. For example, using dark pixel ratio is a piece of information human agent would not have access to (unless they have a magnifying glass and some extra time to count the pixels manually). Alternatively, the human agent would be able to recognize gaps and holistic patterns by collectively looking at the matrix.

One of the problems that the agent was successfully at was Challenge D-11. This problem, at first blush, seems too abstract for the agent to resolve however it showed almost something akin to intuitiveness in deducing this correctly. Even though the agent was not successful in answering all the problem sets in the grader, the little engine that could still manages to surprise and perform beyond the scope of its original programming.

7 REFERENCES

1. Unsworth, N., Fukuda, K., Awh, E., & Vogel, E. K. (2014). Working memory and fluid intelligence: Capacity, attention control, and secondary memory retrieval. Cognitive Psychology, 71, 1–26. https://doi.org/10.1016/j.cogpsych.2014.01.003
2. Yang, Y., McGreggor, K., & Kunda, M. (2020). Not Quite Any Way You Slice It: How Different Analogical Constructions Affect Raven's Matrices Performance.
3. Joyner, D., Bedwell, D., Graham, C., Lemmon, W., Martinez, O., & Goel, A. (2015). Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems on Intelligence Tests. In Proceedings of the Sixth International Conference on Computational Creativity. Provo, Utah.
4. Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: a theoretical account of the processing in the Raven Progressive Matrices Test. Psychological review, 97(3), 404–431.
5. Shegheva, S., & Goel, A. (2018). The Structural Affinity Method for Solving the Raven's Progressive Matrices Test for Intelligence. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/11323
6. Vasilisa Bashlovkina, Sudoku Approach, https://edstem.org/us/courses/8279/discussion/866620?answer=1985333
7. Wikipedia contributors. (2021b, August 6). *Root-mean-square deviation*. Wikipedia. https://en.wikipedia.org/wiki/Root-mean-square_deviation

8 APPENDICES



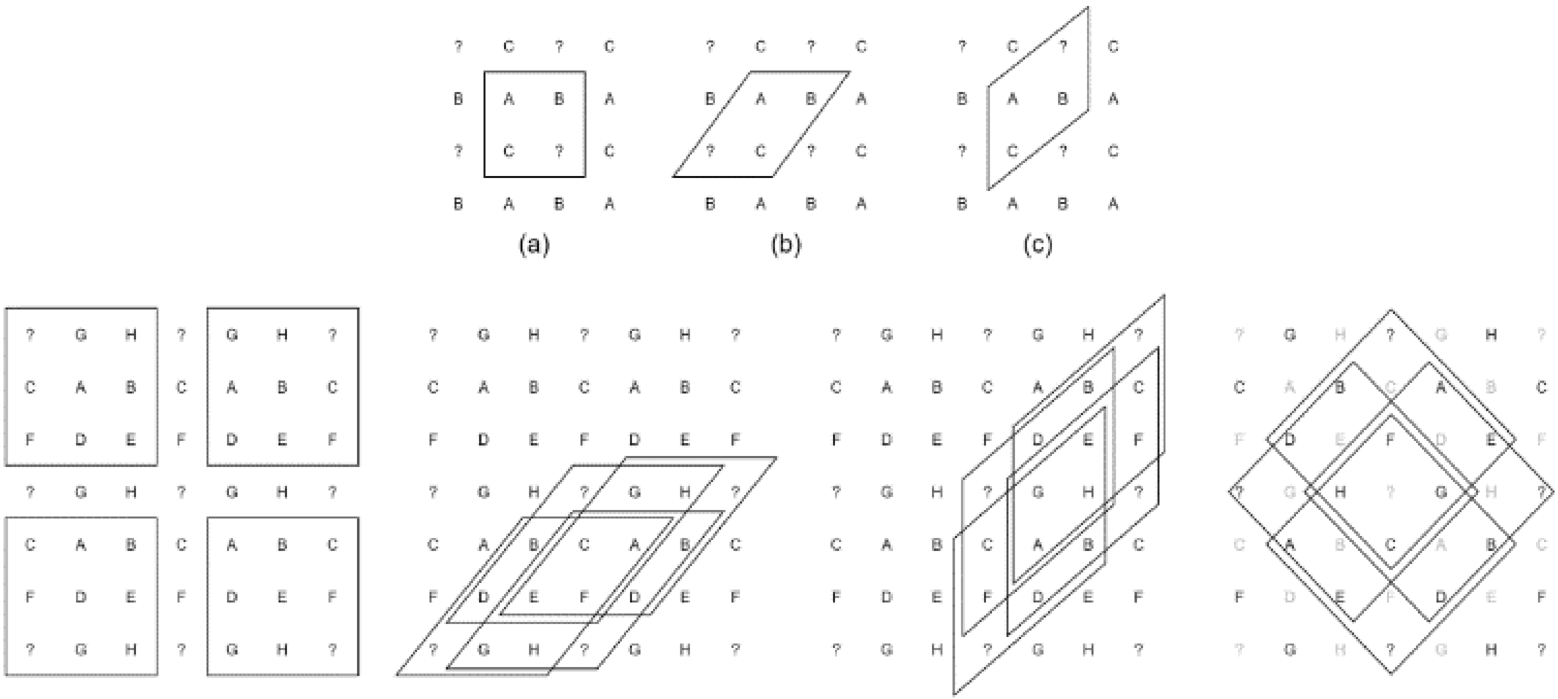


Figure 14 – Combinatory sets by varying dimensionality (Yuan Yang, 2020)

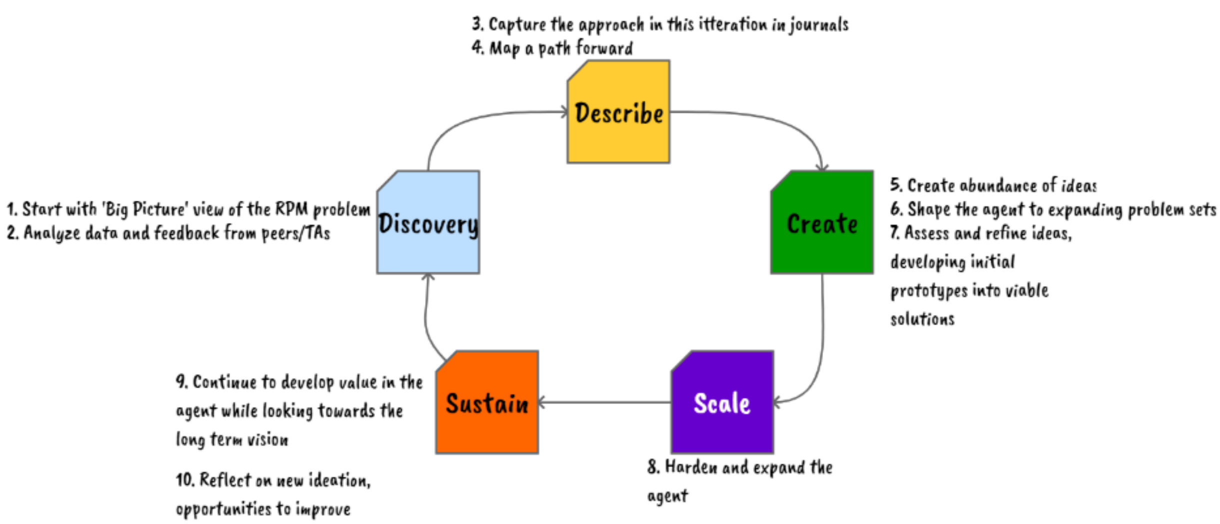


Figure 15- Design paradigm

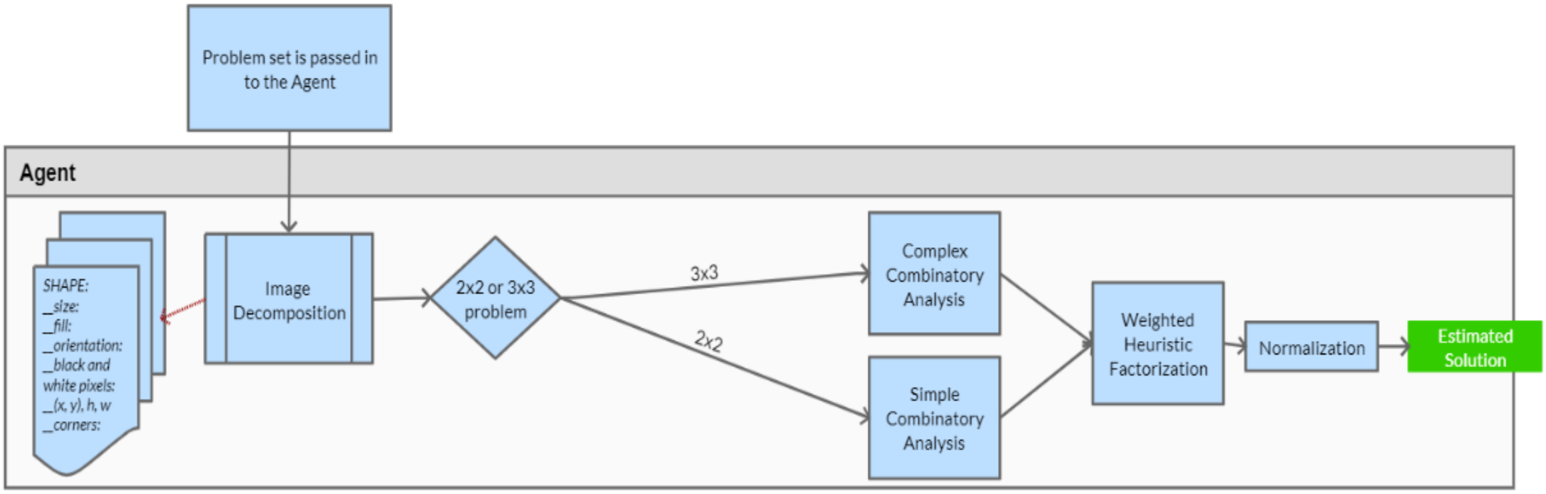
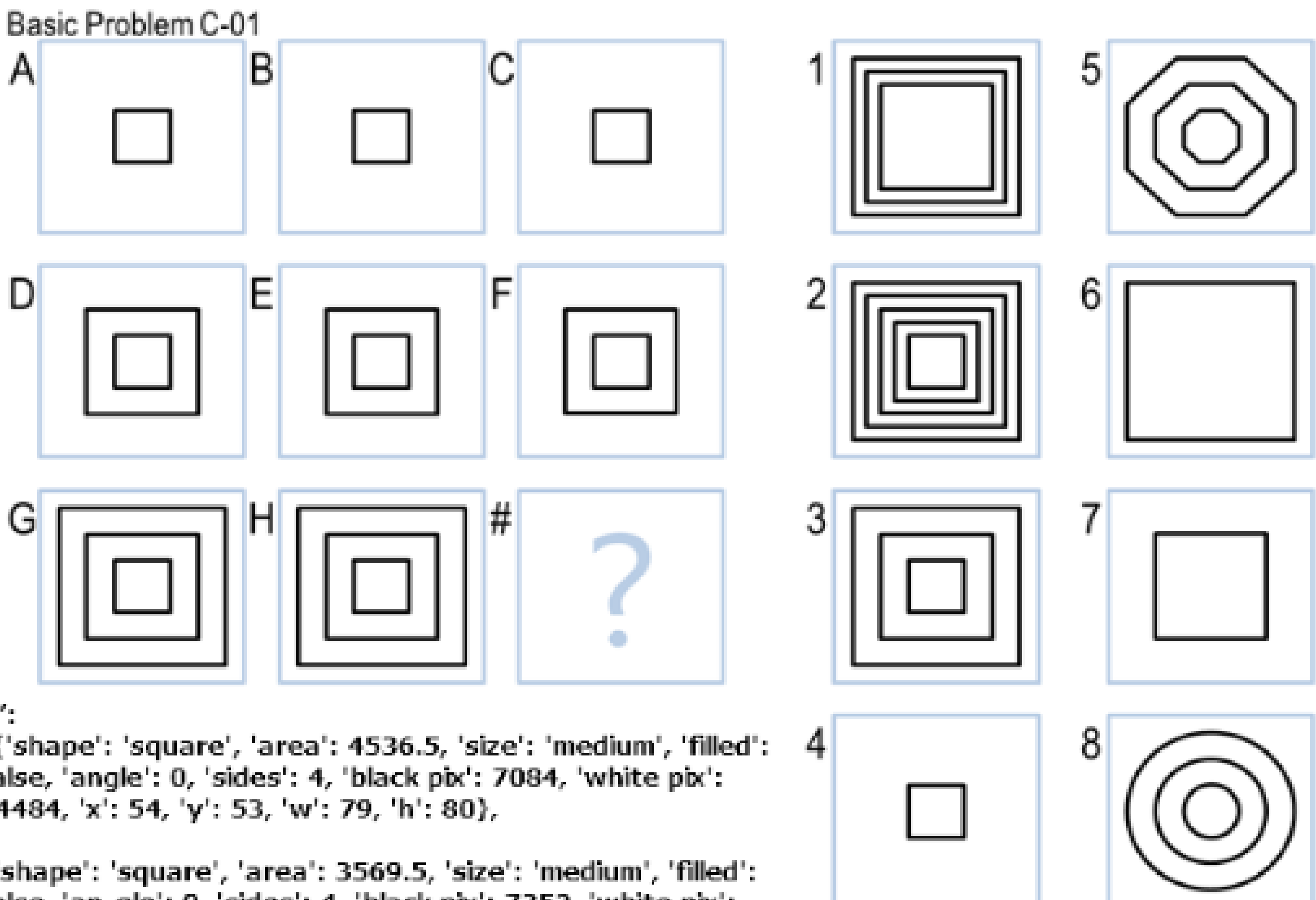


Figure 16 Agent Algorithm Design



'F':
[{'shape': 'square', 'area': 4536.5, 'size': 'medium', 'filled': False, 'angle': 0, 'sides': 4, 'black pix': 7084, 'white pix': 94484, 'x': 54, 'y': 53, 'w': 79, 'h': 80},

{'shape': 'square', 'area': 3569.5, 'size': 'medium', 'filled': False, 'an-gle': 0, 'sides': 4, 'black pix': 7352, 'white pix': 94216, 'x': 56, 'y': 57, 'w': 72, 'h': 72}]

Figure 17 – Example of image decomposition from Basic Problem C-01

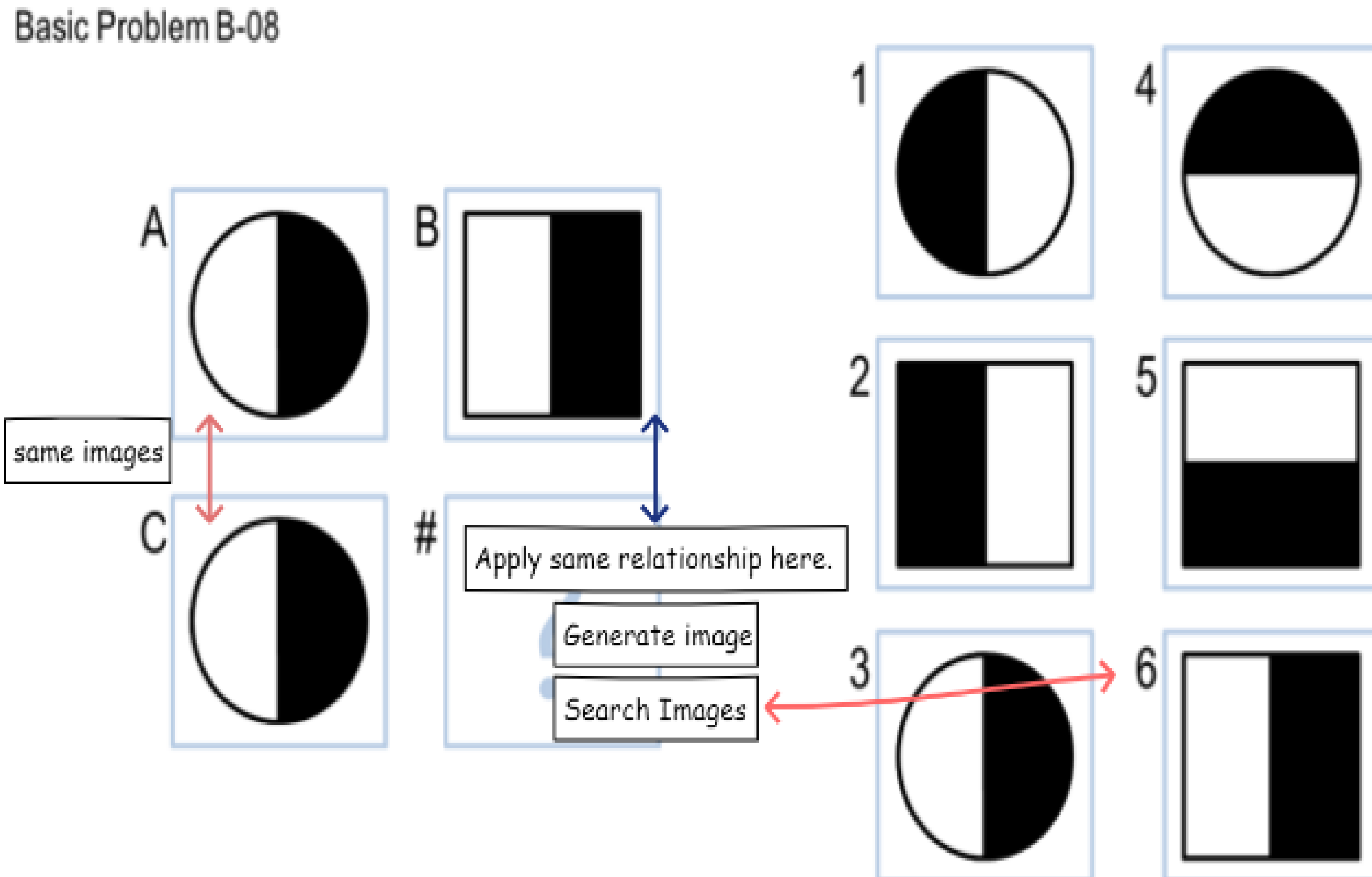
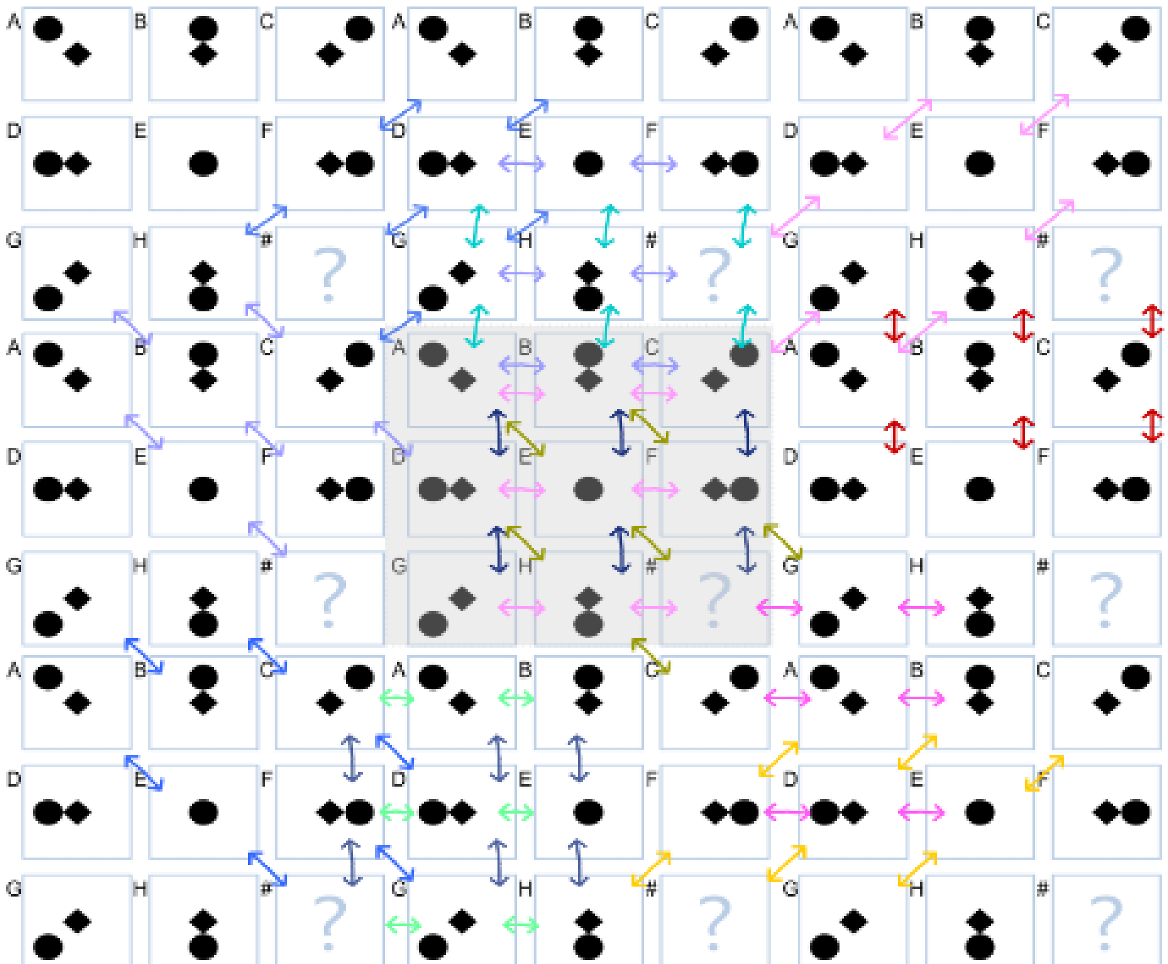


Figure 18- 2x2 matrix with (a) combination, using similar image heuristic




```
testing_combinations = [
    [("AB", "BC"), ("DE", "EF"), "G", "H"],
    [("AD", "DG"), ("BE", "EH"), "C", "F"],
    [("DH", "HC"), ("BF", "FG"), "A", "E"],
    [("DA", "AG"), ("EB", "BH"), "F", "C"],
    [("CA", "AB"), ("FD", "DE"), "G", "H"],
    [("AD", "DG"), ("BE", "EH"), "C", "F"],
    [("BE", "EH"), ("AD", "DG"), "C", "F"],
    [("FD", "DE"), ("CA", "AB"), "G", "H"],
    [("CE", "EG"), ("AF", "FH"), "B", "D"],
    [("AC", "CB"), ("DE", "EF"), "G", "H"],
    [("HC", "CD"), ("FG", "GB"), "A", "E"],
    [("HF", "FA"), ("CG", "GE"), "D", "B"],
    [("FG", "GB"), ("HC", "CD"), "A", "E"],
    [("FH", "HA"), ("GC", "CE"), "B", "D"],
    [("GF", "FB"), ("CH", "HD"), "E", "A"],
    [("GC", "CE"), ("FH", "HA"), "B", "D"],
]
```

Figure 19- Combinatory pattern and list used in 3x3 problems

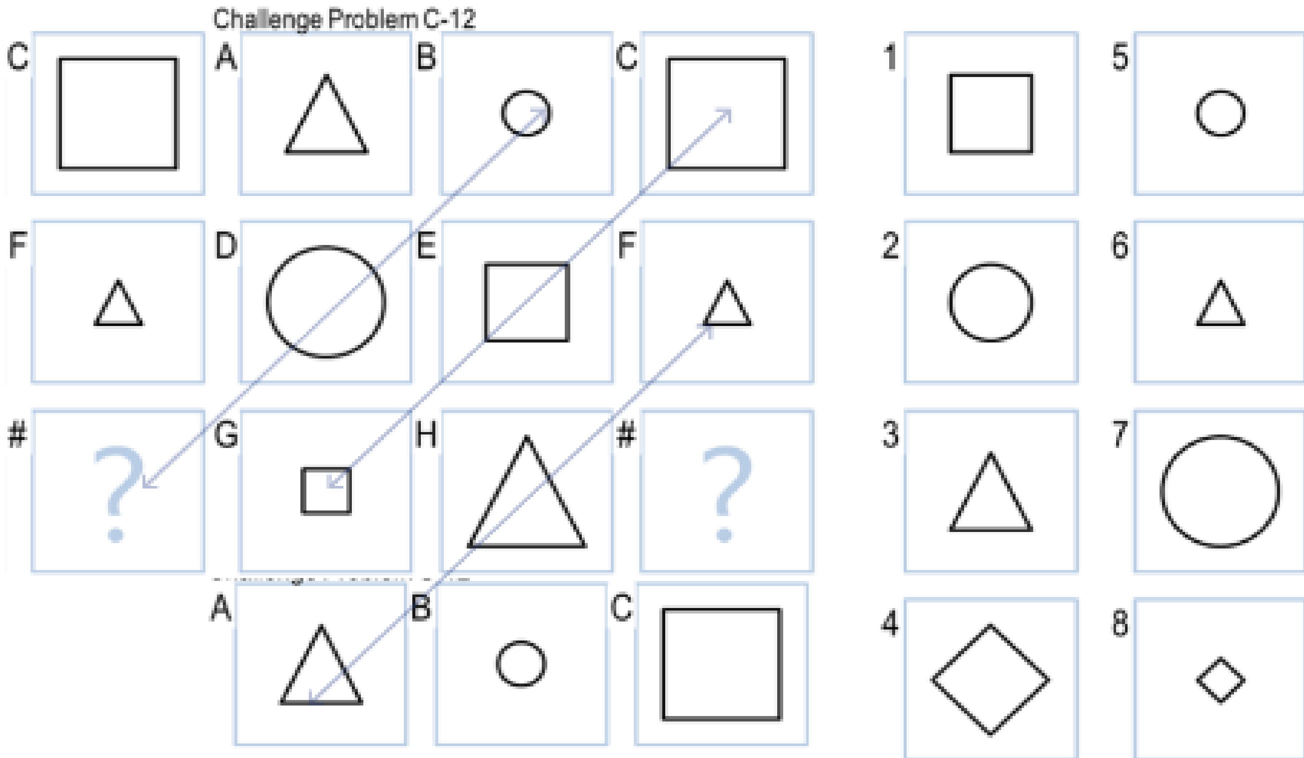


Figure 21 - Successfully Problem solved

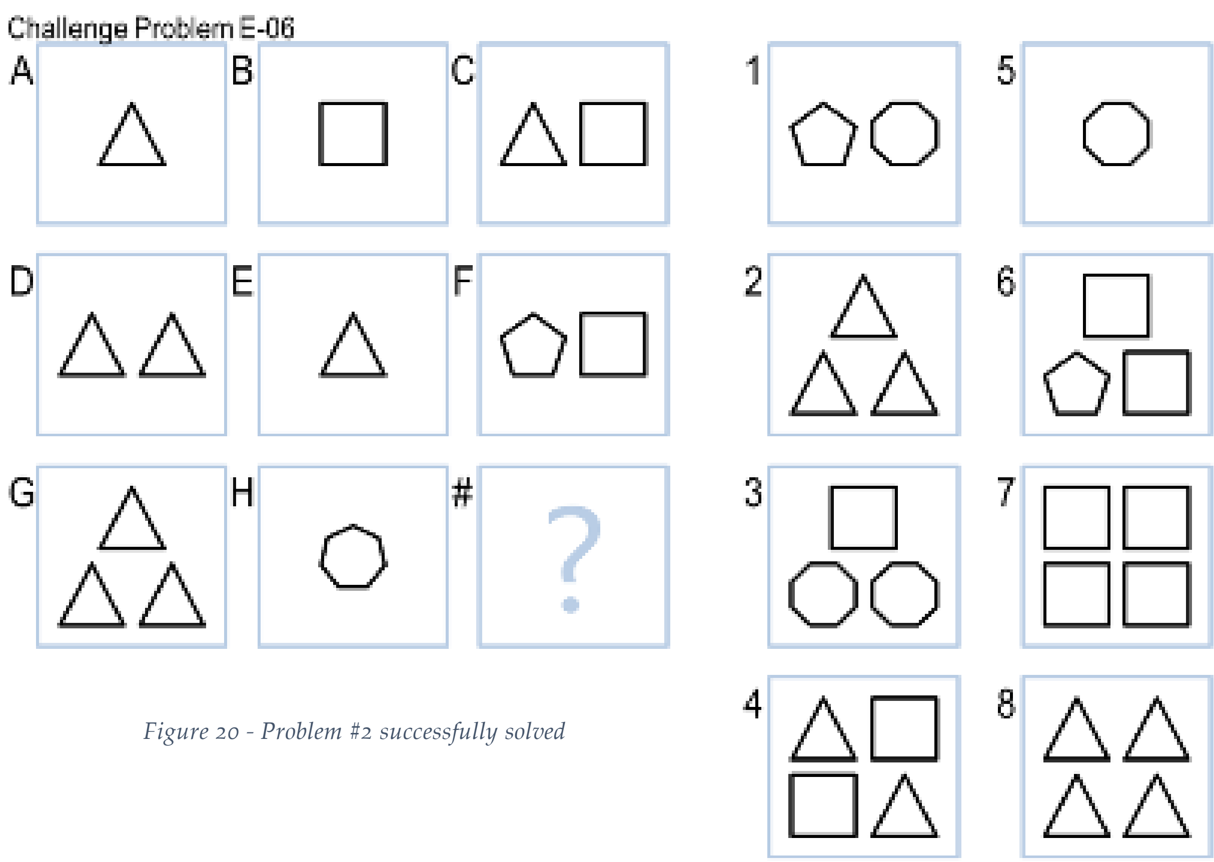


Figure 20 - Problem #2 successfully solved