ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

cognitive Architecture

TOWARDS SEQUENTIAL PROBLEM SOLVING IN ACT-R: A CASE STUDY OF TANGRAM

CANDIDATE

Giacomo Zamprogno

SUPERVISOR

Prof. Giuseppe Di Pellegrino

CO-SUPERVISOR

co-supervisor

Academic year 2021-2022

Session 1st

+retrieval>

ISA dedication

NAME tbd

REASON tbd

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Where I describe cognitive modelling and its applications, and the specific case study of the tangram in the context manufacturing and tutoring

What is cognitive modelling? Why how does it relate to AI and why is it of interest at airbus? MAYBE the reserch group at airbus The idea of the project

1.1 Tangram

The tangram is an ancient Chinese puzzle in which seven pieces, also called *tans* are obtained from an original square.
The most common tans, also used in this work, consist of 5 square triangles (2 small, 1 medium and 2 large), 1 square and 1 parallelogram, their dimension is shown in Figure 1.1 .
Usually players are presented with an homogeneous silhouette, likely in the shape of some stylized figure, and are asked to reproduce the pattern by using all the tans, without overlap.
Besides the interest among puzzle-solvers, a number of works has been studying its applications in the teaching fields, where its nature as a fairly complex game can help children to develop geometric and communicative skills[citation to some papers might be needed here].

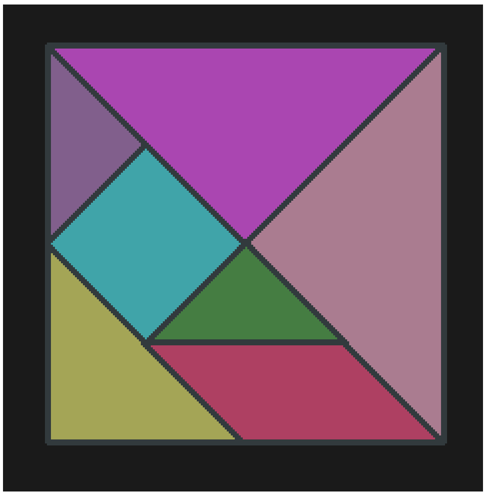

Figure 1.1: Most commond tans and starting position

In the context of cognitive modelling, the tangram can be seen as an abstraction for a set of different sequential problem solving tasks. The fact that the field is still at the early stages of development, studying a simpler puzzle and how humans approach its solution might provide initial insights about various types of assembling processes, in an attempt to create machines more capable to interpret and adapt to human actions in an explainable and rule-based way.

Chapter 2

Related Works

Where I quickly go throught the available Literature, describe ACT-R and its functioning and analyze the various approaches for tangram solving

2.1 Cogntive modelling of puzzles

Despite their nature and potential as an abstraction for more sophisticated sequential problem solving tasks, the applicaions of cognitive modelling to puzzles are still at an early stage.
Rosenberg et al. [5] coupled cognitive architectures and the tangram puzzle in order to model the curiosity aspect of a social robot, but the actual solution of the puzzle was implemented with a connectionist approach and the cognitive aspect was focused on the social interaction and artificial curiosity modelling.
Gentile and Lieto [2] instead used ACT-R in order to model the role of mental rotation applied to the task of the Tetris™ video-game, based on the previous work of Shepard and Metzler [6], providing introductory results and a functioning model for the mental rotation process.

2.2 Tangram

As mentioned, cognitive-modelling specific literature regarding the tangram is extremely limited. Nonetheless, previous works in the fields of computer science, neuroscience and cognitive psychology can be seen as an interesting starting baseline for the modelling process.

Among the computational approaches, Deutsch and Hayes [1] originally suggested a heuristic solution based on extending the lines inside the silhouette and matching the edges of the tans with the prolonged lines; they intersetingly suggest an heuristic approach based on different types of matching, distinguishing between a direct matching type, where all the sides perfectly match, and a partial matching type, where only part of the edges match with the lines or their extensions. While possibly tangent to some cognitive processes at a higher level, the strictly geometrical nature of their solving algorithm and especially the complex evaluation of the lines extensions reduce its applicability in the current architecture.

During the advent of machine learning, Oflazer [4] used a Boltzmann Machine, where each neuron represented a possible positioning of one piece, and was fed with excitatory input from the outline constraints and inhibitory input from the conflicting configurations. The main limit of the approach is that the authors only tackle *grid tangrams*, where each corner of a placed tan must be coincident with a point in an equally spaced grid. This in turn largely limits the allowed rotations for the pieces and thus its applicability, even at a knowledge representation level, when dealing with less constrained puzzles as the irrational length of edges would be impossible torepresent with precision by using the grid.

The work from Hu, Lam and Yuan [3] on neural activity during the tangram task suggests a possible direction to take in order to reproduce human behaviour. The authors monitored the fronto-paretal area of the brain during the puzzle solution, and propose that trial and error strategies might be prevalent instead of inductive reasoning. In this case, the need for forward reasoning mechanisms in the model might be reduced in favour of an insight based approach.

2.3 ACT-R

Chapter 3

Experimental scenario

In collaboration with the Technical University of Berlin (TUB) an experiment was developed and performed in order to obtain the training and test sets.
The participants were recruited among the students of the TUB, and two experiment sessions were held at different dates, with different participants, for a total of [waiting for test set]. Participants signed an informed consent for the usage of the gathered data and were compensated with [which model?] miniatures provided by the company. Each participant was presented with a virtual implementation of the tangram game, including 4 puzzles [cite figure]. After an inital explanation of the controls, each participant was required to tackle the solution of the puzzles, which were always shown in the same order. There was no time limit, and at any moment the *NEXT* button was available, so that the player could give up on the specific tangram and move to the next one. It must be noted that while backtracking was possible during the solution of the single task, once the button was pressed the solution was submitted and no option to go back was available.
Between each problem, the participants were asked to complete a NASA evaluation form, and at the end of the trial they provided a general feedback.
In addition to this, the screen recording and application logs, including data regarding times, piece action types (rotation, movement) and piece positioning were stored for the analysis.

During the first trial set, 31 participants (17 males, 14 females) took part to the experiment; they all had an academic background, a large majority of them being from engineering or human factors studies.
These first players composed the training set.

Chapter 4

Data Analysis and Hypothesis

[very initial draft]

Methods

Watching videos

What is considered a step

Heatmap results

Individual sequence analysis

What is observed is that participants tend to reproduce ”similar” sequences at the beginning and at the end of the tasks. The order ususally is not strictly relevant, but common configurations are present in the solutions of different players.

Landmarks

The frequency of repeated patterns in the solution sequences hints at the presence of particular features of the silhouette that are often noticed and exploited by the participants during the task.

In order to model such observation, the concept of **landmark** was introduced, so defined:

A landmark is feature of the task silhouette that strongly hints at the possibility of placement of a given tan

The definition as it stands is still somehow general, nonetheless it allows for a clear approach in the implementation of the agent: provide a specific definition of landmark, extract whatever landmarks are available at the current state, act on one or more of such landmarks according to hypotesized strategies.

Best fit

A common and somewhat natural characteristic of most solution processes is the preference to place tans along the edges of the silhouettes, seemingly by exploiting information about particular corners or corner-edge sequences. Paricularly, in a similar fashion to the direct matching suggested by [1], moves that would allow an almost perfect match between the tan and a sub-part of the silhouette (the chimney in tangram 2, the left sail in tangram 3, the foot,arm or hat in tangram 4) are usually performed within the first 3 or 4 steps. It must be noted that this is not strictly limited to clear and mandatory placements: in tangram 3, the sail can be otherwise created by using other tans, and in tangram 4 the placement of the parallelogram in the arm is not clearly hinted at the beginning. This suggests a somewhat relaxed and noisy interpretation of such strategy, thus defined as *best fit* instead of *perfect* or *direct*. In any case, due to its conisistency in the data, the *best fit* concept will be the main design choice for the agent: landmarks will be defined as possibly imprecise placements of the tans, extracted via a pattern matching detection on the edges of the silhouette.

Combination issues and rotation resistance

The approach based purely on the *best fit* strategy seems be an acceptable hypothesis when considering the initial steps, but by itself it fails to account for some of the difficulties encountered by the participants: among these, of particular note is the tendency to miss the possibility to combine two smaller triangle into a larger one [add picture]. Both in tangram 2 and tangram 4, the correct solution involves such combination, but quite consistently [stats to be added] participants tend to first orient a single triangle with an to match the larger one, instead of noticing the possibility of combination, see figure ...
More intersetingly, when backtracking or trying different strategies it might happen that such move is repeated, even if already discarded; this seems to be in accordance to the hypothesis from [3] that the solution is mainly performed via random search instead of inductive reasoning.
In order to integrate such observations with the *best fit* strategy, an additional component is suggested: a certain placement is not only decided by how closely the interested piece matches with the silhouette, but also by how strongly it draws the agent's attention.
The concept of *landmark strength* is not too unusual and can find a direct correspondence with the baseline activation of chuncks in ACT-R. The main concern is how to define such strength: as research of tangram puzzle is relaitvely limited, few suggestions have been made on what could best draw attention, [cite missing paper] suggests that there is a preference on size of pieces, the observations mentioned above hint that $90°$ angles might be somehow involved in rotation resistance, and indivudal biases should likely be taken into account.
In order to overcome such uncertainty, the descibed agent takes a data driven approach: how strong a certain landmark is (and thus how it is likely to be chosen) can be inferred by the number of people that indeed acted upon such landmark, the baseline activation will thus be derived from the frequency of participants that exploited such landmarks at the current phase of the solution.

Chapter 5

Model description

The agent model consists of multiple sub-components and a coordinator which handles their interaction. The components are relatively distinct and provide the various functionalities of the system: the *application* module creates and updates the experimental window, the *landmark extractor* module implements the visual system and extracts ”plausible” placements of the tans in the current state, the ACT-R module performs the main reasoning task and decides the next action.

In principle, it would be possible to create and run an experimental setting entirely within the ACT-R context. In the specific case of the project, however, the available data were strictly dependent on the implementation of the experimental window: the coordinates for the placements of the tans were expressed with respect to the application window, and the tans themselves were defined as shapes within the framework. Reproducing the setting purely in ACT-R would likely have caused loss of precision in the shapes and coordinates definition, besides an additional overhead due to the lack of established methods for complex image manipulation: the creation an interfacing system was then preferred.

5.1 Experiment window

The applicaion window during the experiment was derived by an existing Tangram demo application [shall i cite? it comes from ”tangram4kids”] and included interactive features which allowed the users to manipulate and move the tans. While such features were not required in the developed model, as no simulation of motor functions was planned, most of the data gathered during the experimental sessions was framed in the context of the application. As a result, a similar window was mantained for better corrspondence, and the model runs on a lighter adaptation of the code, providing just two main functionalities: it updates the window to represent the current state, and it captures the window screen so that it can be forwared to the visual system for processing.

5.2 Visual system

The agent’s visual system was implemented in order to be ”cognitively inspired” from the hierarchical structure of the human’s visual system. Given the strictly geometrical nature of the puzzle, and the fact that *second layer neurons* [need the proper name and citation] fire when matching certain oriented lines in the visual field, the algorithm simulates an hypothetical higher level construct able to identify certain line patterns representing the shapes. This is done via a pattern matching function applied on the edges of the current state and the template, using the sum of squared differences as similarity function. Considering the binary nature of the images, a sum of absolute differences or a custom function might have been possible alternatives, but due to the amount of templates to match (for each shape and for each of its available rotations) the faster opencv implementation was preferred, even though limited in the similarity functions options.

For each template, 7 candidate placements are extracted. These are then filtered in two successive steps. First, it is checked whether the placement would intersect with other pieces currently placed, which can happen as the similarity function on edges might still tolerate limited intersection of corners. This is done by imposing a threshold on the following similarity function:

$$\sum_{x,y}((part \ \wedge \ template) \oplus template) \tag{5.1}$$

Where $\wedge$ denotes bitwise-and, $\oplus$ bitwise-xor, and *part* is the part of the target image in the candidate bounding box.

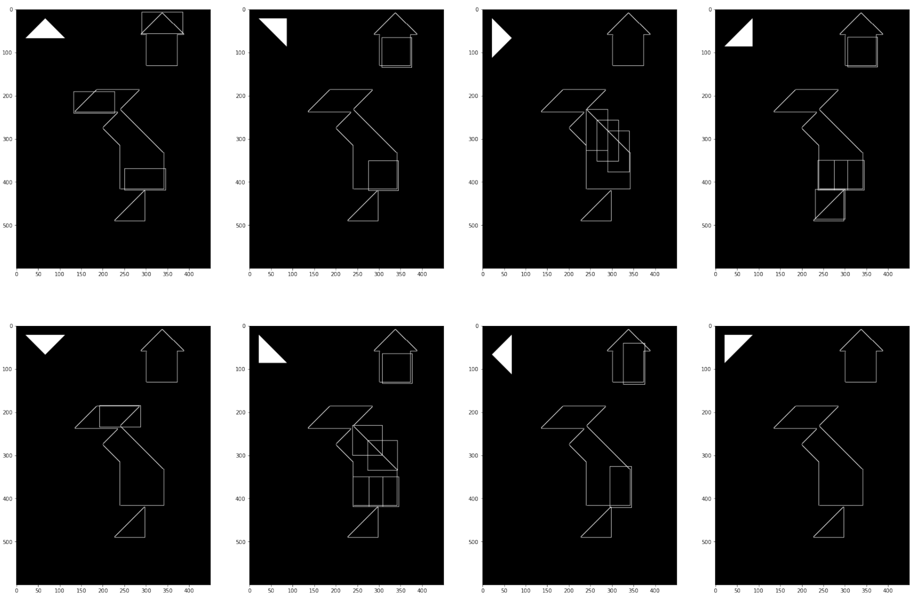

Figure 5.1: Extracted landmarks before intersecting with human data

The second screening checks for feasibility: the template matching method applied aims to simulate a possible cognitive mechanism, but is in no way plausible per-se and might find patterns that are unseen or hardly seen by the humans. As a consequence, only the placements that have some representation in the training data at the current phase of the solution are kept. This also tackles an additional aspect: once a landmark is found its strength must be defined, which in turn will determine the baseline activation of the landmark chunk itself in the model. While a possible solution would be basing the strength on the similarity value, this would not have any cognitive foundation. On the other hand, the data provide a clear indicator of its likelihood: if a large number of participants acted on the landmark in the current phase of the game, it should in turn mean that such landmark should be a strong landmark. The frequency count is thus used not only for screening, but also for defining the baseline activation of the chunk once loaded into the declarative memory.

As the human's visual (working?) memory is limited [cite maybe 7+-2, or something more relevant], only the 6 strongest landmarks will define the imaginal buffer, with the rest being merely loaded into the declarative memory with their respective strength.

A final task performed by the visual system, is to recognize the presence of problematic regions (UNF-REGIONS): in principle, such regions are parts of the silhouette in which no tan can be placed respecting the rules. They cannot be matched directly as they come in various different shapes and sizes, but a property of the task can be exploited to identify some of them: the silhouette area must eventually be fully covered and all available pieces used. Considering this, if at any point in time no available placement is found for a piece,it means that its area is split between two or more separate regions that cannot accommodate it. In such cases, the presence of a problematic region is found and the last action is marked as uncertain.

5.3 Coordinator

The coordinator is the main python application, providing the interfacing features between the other modules. It initializes the various sub-components, keeps an internal representation of the puzzle state and updates it accordance to the chosen ACT-R action.

5.3.1 State representation

The tangram puzzle currently being solved is represented by the eponymous class *Puzzle*. Its fields are mainly used for listing the free tans, the currently used tans, their respective positioning and the landmarks they are involved in. While in principle the definition of the landmark already includes the involved piece, the presence of multiple tans of the same type requires such distinction, in order to avoid acting on already placed landmarks. A given action is thus defined as a tuple (Piece, Landmark). A list of the overall sequence of actions is finally stored for the performance evaluation.

5.3.2 Updating actions

While the ACT-R module will provide the reasoning and eventually choose the action performed at each step, the implementation of such action is delegated to the coordinator. The main methods that provide such functionalities are described in the following:

- **update**: the update method represents the main operation for the task solution. Accepting a landmark definition (piece type, position, orientation) it picks the first available piece corresponding to such type, creates the (Piece, Landmark) match, and updates the state accordingly.
- **region_backtrack**: this method ideally represents the attempt for backtracking when a region that cannot fit any available piece is identified. Humans (and thus the agent) do not necessarily act on such problematic regions in the moment they produce them, but only once they are noticed.
  In order to simulate such behaviour, the visual system will flag such occurrence as soon as it appears, by creating an ”UNF-REGION” landmark, but the method will be called only once the agent actually matches it via firing the backtracking production. All actions performed while

such a landmark exists will be flagged as potentially problematic, and every time the method is called, the first such action will be backtracked, possibly clearing the queue if the issue is not present anymore.

- **piece_backtrack**: backtracking can happen even if no problematic region is matched: participants can realize that a current placement is leading to no solution, or fail to identify a possible landmark. Both of these cases are represented by a retrieval failure, which in turn triggers the piece_backtrack method.
  In this case no information about problematic placements is available, so the decision on which action to backtrack is based on the landmark strength, as determined by the data. For each of the active landmark the function extracts its frequency count at the current game state, and the weakest one is chosen for backtracking.

5.4 ACT-R model

The ACT-R model is tasked with choosing the next action to take at any given position, and it is mostly based on the baseline activation and spreading activation mechanisms; three modules are involved in the process: the imaginal module, the declarative module and the retrieval module.
As previously described, the visual system extracts the current landmarks, adds them to the declarative memory according to their strength, and subsequently loads the 6 strongest into the imaginal buffer. In this context, the imaginal buffer represents the ”most noticeable” landmarks, and helps their retrieval by the spreading activation mechanism.
A description of the main production rules follow:

- **retrieve-landmark**: as the name suggests, this production calls for a retrieval of any given landmark. It matches when there are still pieces available and at least one noticed landmark is in the imaginal buffer.It poses no restrictions on the retrieval request, other than asking for a landmark chunk which has not been recently retrieved.

The retrieval process is then guided by three components: the baseline activation, which is provided at chunk definition (by the visual system) and depends on the landmark strength, the spreading activation, depending on the landmark chunks present in the imaginal buffer, and a noise component, the distribution of which is an hyperparameter of the model.

- **act-and-update**: this production is triggered whenever the retrieval from **retrieve-landmark** is successful. Its task is to extract the relevant fields from the landmark chunk and to provide them to the coordinator, by calling the **update** method.

- **fail-to-retrieve**: the experiments additionally show that a participant might fail to retrieve certain landmarks even if they are visible and representing a near "perfect-match". In order to simulate such behaviour, a retrieval threshold value is introduced: if, due to noise or weakness of activation, no landmark's activation is above the threshold a buffer failure will happen in the retrieval request. Such failure is matched by the production **fail-to-retrieve**, its purpose being to trigger the **piece_backtrack** procedure.

- **notice-problem**: this production is in direct competition with **retrieve-landmark** whenever the special landmark *UNF-REG* is present in the imaginal buffer. This represents the participant noticing a problematic region in the puzzle and thus starting the backtracking process of solving such region.

- **solve-problem**: this production directy follows **notice-problem** and, in a similar fashion to **fail-to-retrieve**, its task is to interact with the coordinator in order to call the correct backtracking method.

The interaction between the two backtracking procedures should be noted: in some cases, a given incorrect placement of a piece does not create any recognizable unfeasible-region, but will cause all the following actions to be problematic. This will not be noticed by any of the two individual productions, but it will eventually be possible to backtrack by their combination. Due to the constraint on *:recently-retrieved nil*, after various iterations of the **region_backtracking** with no solution, no new landmark will be available. At this point the **piece_backtracking** method will be triggered, which will remove the weakest landmark. At later phases of the solution, the problematic landmarks will be less frequent, as more participants will actually have solved the puzzle, and its strength in the model will thus be lowered.

Chapter 6

Results and evaluation

Where I compare the model to the expected data and try to discuss whether the hypothesis are funded and whether there are possible applications

Chapter 7

Conclusions and future developments

Bibliography

[1] E. S. Deutsch and K. C. H. Jr. “a heuristic solution to the tangram puzzle”. *Machine Intelligence*, 7, 1972.

[2] M. Gentile and A. Lieto. The role of mental rotation in tetristm gameplay: an act-r computational cognitive model. *Cognitive Systems Research*, 73:1–11, 2022. ISSN: 1389-0417. DOI: https://doi.org/10.1016/j.cogsys.2021.12.005. URL: https://www.sciencedirect.com/science/article/pii/S1389041721000991.

[3] Z. Hu, K.-F. Lam, and Z. Yuan. Effective connectivity of the fronto-parietal network during the tangram task in a natural environment. *Neuroscience*, 422:202–211, 2019.

[4] K. Oflazer. Solving tangram puzzles: a connectionist approach. *International Journal of Intelligent Systems*, 8, 1993.

[5] M. Rosenberg, H. W. Park, R. Rosenberg-Kima, S. Ali, A. K. Ostrowski, C. Breazeal, and G. Gordon. Expressive cognitive architecture for a curious social robot. *ACM Trans. Interact. Intell. Syst.*, 11(2), 2021. ISSN: 2160-6455. DOI: 10.1145/3451531. URL: https://doi.org/10.1145/3451531.

[6] R. N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971.

Acknowledgements