An approach to computational creation of insight problems using CreaCogs principles*

Arpit Bahety and Ana-Maria Olteţeanu[1] [0000−0002−0639−7956]

Cognitive Systems Group, Human-Centered Computing, Freie Universitat Berlin, Germany ana-maria.olteteanu@fu-berlin.de

Abstract. Insight problems are used in the study of human creativity problem solving to evaluate the creativity of the solver, and the process through which creativity problem solving is cognitively deployed. However, not many such problems exist, and the factors underlying their creation are not well controlled. The framework CreaCogs proposes ways in which cognitive AI systems could be used to solve diverse such problems using a small set of processes. In this paper, a previous approach for the creation of insight problems proposed in CreaCogs is implemented computationally. The initial experiments, results, limitations, perspectives and potential are reported upon.

Keywords: insight · creativity · creativity problem solving · cognitive systems · cognitive AI · psychometrics

1 Introduction

Creativity and creativity problem solving are analysed and measured in the cognitive science and cognitive psychology literature with a set of tests – Alternative Uses Test [12], the Remote Associates Test [5], the Torrance Tests of Creative Thinking [3], the Wallach-Kogan tests [1], riddles [14], empirical insight tests [2, 4, 13] and others. Out of these types of tests, the ones most suffering from a scarcity of stimuli are insight problems with practical objects.

One psychometric limitation of insight problems is that, once the participant has solved a problem, this will most likely not produce insight anymore, as the solution path has already been trodden by the participant. Not all problems requiring creativity would produce insight, but having a bigger repository of problems that require creativity with practical objects would allow for a wider and deeper exploration of creativity processes, and for a selection of problems most likely to produce insight.

A computational approach to creating practical object insight problems was previously proposed [7], given principles of a framework for creativity problem

*The support of the German Research Foundation (Deutsche Forschungsgemeinschaft) through the grant awarded to project CreaCogs OL518/1-1 is gratefully acknowledged.

solving in cognitive AI systems – CreaCogs [11, 6]. In this paper, the approach is implemented computationally, and an initial set of experiments conducted and discussed.

The rest of this paper is organized as follows. Section 2 discusses the general approach to problem creation. Section 3 and gives a running example over all the problem creation steps using an example problem. Section 4 takes the formalization from [7] and constructs algorithms, based on the CreaCogs framework principles. Section 4 shows our computational experimentation on insight problem generation, with the help of three example problems. Section 5 lists the general results obtained for our computational experimentation. Section 6 lists some results generated from our computational experimentation. The paper concludes with a discussion in section 7.

2 Approach to Problem Creation

The approach to insight problem creation proposed by [7] and developed here proposes to start from a non-creativity problem which involves day-to-day objects and the solution of which is known. Then, that particular solution is hidden, and the problem transformed by a set of techniques further explained, in such a way that the problem requires creativity to be successfully solved.

The first step in the process of creating insight problems computationally[2] is obtaining and encoding a non-creativity problem. This encoded non-creativity problem is the input to the insight problem generator. A set of techniques are applied to this input to transform the non-creativity problem to an insight problem. The flow of steps is shown in Figure 1 and each of the steps explained further. It is important to mention that not all the steps need to be applied for each of the problems, but they rather constitute a repertoire of actions which transform the problem.

The insight problem creation process can be better understood with an example problem. The problem that will be discussed here is one manually created with this approach in [7] – *The blown away teddy problem.*

3 The blown away teddy problem

This problem presents the participant with the following task: *The wind blew your sons teddy bear from the clothesline into your neighbours garden. The neighbour is in holidays and the fence is too high to climb. How can you retrieve the teddy?* Figure 2 shows the problem.

One solution to this problem could be to construct a fishing rod (using the mop, the clothesline, and a clothes hanger attached to the clothesline); this fishing rod can then be used to attempt to fetch the teddy.

The above problem is a problem that requires creativity to solve, which is the *output* of the problem creation process. The *input* is a simple problem not

[2]This process will also be referred to as the problem transformation process.

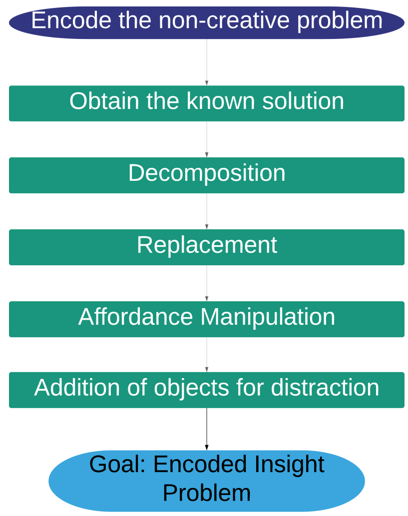

Fig. 1. From non-creative to insight and/or creativity problem

requiring creativity. A non-creative version of this problem would be one in which the solver can simply step in the neighbour's garden. Assuming extra constraints (the solver is not allowed in the garden, and doesn't have a key), a version of this problem could require a fishing rod to catch the teddy. Showing the fishing rod as part of the problem would be, in our example, the less creative version of the problem (the input problem). Using this, we use the problem creation process to obtain a problem requiring creativity, and to showcase the approach.

The process starts with the encoding of the non-creativity problem.

3.1 Encoding

The encoding module can be broken down into three sub-parts:

1. Why are we encoding?
   The non-creativity problem, which is the input to the problem transformation process is in words and images. To make the input in a machine understandable format, we have the encoding module.

2. How does the encoding look like?
   The encoding [6] of the input has two parts:
   (a) Problem encoding:
   The input problem is transformed in a set of `concepts`(C_i), `relations`(R_i), `actions`(H_i), `goals`(G_i) and `constraints`(K_i), avoiding NLP issues. In this encoding: `Concepts` are everyday objects with properties such as shape, material, size etc.

Fig. 2. The blown away teddy problem

`Relations` are an association between two or more concepts.
`Actions` are the action performed on the concepts which may or may not lead to a change in relation.
`Goals` describe the final state to be reached. It could be a set of concepts and relations.
`Constraints` describe the resource or action limits of the task.

According to the above scheme, the blown away teddy problem could be encoded as follows:
$C_1 - Fence$
$C_2 - Teddy$
$C_3 - Fishing\_rod$
$C_4 - Person$
$R_1 - far(C_1, C_2)$
$R_2 - between(C_1, C_4, C_2)$
$G_{solution} - \{hold(C_4, C_2)\}$
$K_1 - cannot\_climb(C_1, C_4)$
$C_{problem} = \{C_1, C_2, C_3, C_4\}$
$R_{problem} = \{R_1, R_2\}$
$H_{problem} = \{\}$
$G_{problem} = \{G_{solution}\}$
$K_{problem} = \{K_1\}$
$I = \{C_{problem}, R_{problem}, H_{problem}, G_{problem}, K_{problem}\}$

(b) Encoding of the solution of non-creativity problem:
The solution of the input problem is transformed in a set of

   – solution objects
   – solution affordance(s) of the i^{th} solution object
   – solution affordance of the problem.

   This is further discussed in section 3.2
3. How would the encoding be generated?
   The present implementation has the encoding generated manually. Generating the encoding computationally is an interesting problem and is involved in our future work.

3.2 Knowing the solution:

The solution object(s), needs to be known prior to beginning the problem transformation process; this allows us to conceal or transform the solution related objects and/or their solution related affordances.

The solution object(s) is represented by set C_{sol}. The solution affordance(s) of the i^{th} solution object is denoted by $A_{c_i sol}$. The solution affordance of problem is represented by A_{sol}.

For the example problem, the solution object is a fishing rod. Thus, the transformation steps are applied to this object. The solution related affordance of fishing rod is to fetch far away objects. In this case, the solution affordance of the example problem is also to fetch far away objects. The aim of the insight problem creation process is to hide both the fishing rod and its solution affordance.

$C_{sol} = \{C_3\}$
$A_{c_3 sol} = fetch\_far\_away\_object$
$A_{sol} = fetch\_far\_away\_object$

3.3 Decomposition:

In this step, solution object(s) are decomposed or broken down into different parts and then each part may or may not be re-represented in a different structure or object. This process requires the knowledge of what parts the object consist of. The process outputs more concepts.

For the example problem, the $Fishing\_rod$ can be decomposed into it's constituent parts - $Rod, String, Hook$. This decomposition step decomposes the concept, C_3 and gives us three new concepts.

$C_5 - Rod$
$C_6 - String$
$C_7 - Hook$

Decomposition essentially breaks down the solution objects so that the human solver needs to reconstruct it.

3.4 Replacement:

In this step solution object(s) or their parts are replaced by other object(s) which are similar by properties and affordance to the solution object but for which the said affordance is not as salient as for the solution object. The approach to the process of object replacement is discussed in [8]. This process requires the knowledge of object properties such as shape, size, material etc. This process does not change the number of concepts or relations, but replaces some concepts with others.

For the example problem, the replacement could take place as follows:
$creative\ replacement(Rod) = Mop\_handle$
$creative\ replacement(String) = Clothes\_line$
$creative\ replacement(Hook) = Clothes\_hanger$
Thus, the new concepts are:
$C_5 - Mop\_handle$
$C_6 - Clothes\_line$
$C_7 - Clothes\_hanger$

Replacement will essentially force the human solver to re-represent the replacement object (or object parts) with the original object. Thus, giving the affordance of the original object to the replacement object.

3.5 Affordance Manipulation:

This step reduces the salient affordance of the solution object(s) by showing the object(s) (or object parts) in different contexts of affordance. The salient affordance can also be concealed by having that affordance as already taken up or in use.

Knowledge of alternative uses [9] of solution objects and knowledge of problem templates [10] is required for this step. There are two ways to perform affordance manipulation.

1. **Knowledge of alternative uses** can be used to show the object in different contexts of affordance. This alternative affordances can only be shown if they are not the same as the solution affordance(A_{c_i}) of the object.
   For example, suppose that for a creativity problem the solution affordance of a solution object, *clothespin* is *to_clip_clothes_on_string*. An alternative affordance of *clothespin* could be $clip\_flower\_and\_stick$. Since, this alternative affordance is different from the solution affordance of *clothespin*, the *clothespin* can be shown in the context of the alternative affordance.

2. **Knowledge of problem templates** – To explain this method, we first explain what problem templates stand for in CreaCogs. A problem template is a sequence of states where actions help in transitioning from one state to the other. a State is a set of concepts, relations and goals that will lead to a particular solution or affordance. Figure 3 explains the structure of problem templates.; problem templates are part of the knowledge of the solver.
   For example, a problem template, PT_{clean_floor} can be described as:
   $C_1 - Mop$
   $C_2 - Bucket$
   $C_3 - water$
   $C_4 - Person$

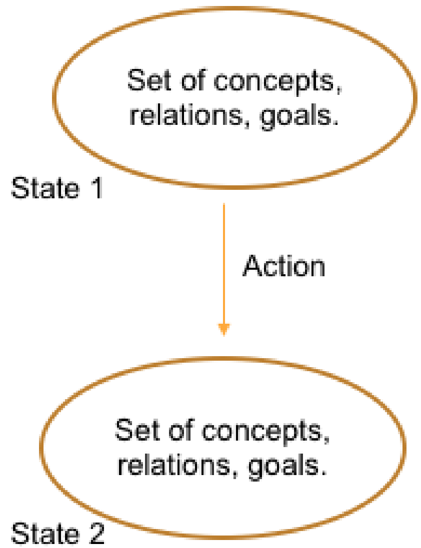

Fig. 3. Problem templates

$R_1 - besides(mop, bucket)$
$R_2 - inside(bucket, water)$

$\downarrow H_1 - grab(person, mop)$

$C = C_1, C_2, C_3, C_4$
$R_3 - hold(person, mop)$
$R = R_2, R_3$

$\downarrow H_1 - mops(person, mop, water, floor)$

$G_{solution} = aff(PT_x) - clean\_floor$

The second method checks if a problem template, PT_x, exists, such that the solution object belongs to the problem template and the affordance of the problem template ($aff(PT_x)$) is not the same as the solution affordance of the problem (A_{sol}). If such a problem template exists, then the affordance of the object in the context of this problem template is shown.
For example, for the example problem, an alternative affordance for the object *mop* can be obtained through this process. The object *mop* belongs to the problem template of $clean\_floor$. The affordance of this problem template, $clean\_floor$ is different from the solution affordance of the example problem (which is $fetch\_far\_away\_object$). Thus, elements of this problem

template, such as the relations belonging to this problem template can be added to $R_{problem}$.
$besides(mop, bucket)$
$inside(bucket, water)$
Thus, the affordance of *mop* is shown in the context of this problem template and its solution affordance of $fetch\_far\_away\_object$ is hidden.

Sometimes, to show the objects or parts in different contexts of affordance, additional objects may be needed. In this case, new objects will be added to the problem. The affordance manipulation step changes the relations between objects, the actions in the problem description, or both.

For the example problem, this step will attempt to change the affordance of the solution objects - *clothesline, mop, clothes_hanger*. One of the outputs of this step could be:
(a) Attaching the clothesline to the pole and hanging clothes on it. This shows the affordance of clothesline as to hang clothes for drying and hides the possibility of using string to make a fishing rod. This leads to new relations being formed:
$R_3 - attached(C_6, pole)$
$R_4 - on(C_6, clothes))$
(b) Displaying the mop next to a bucket filled with water to show that the affordance of mop is to clean.
$R_5 - besides(C_5, bucket)$
$R_6 - inside(bucket, water)$
(c) Hanging the clothes on the clothes hanger and hanging the clothe hanger on the clothesline.
$R_7 - on(C_7, clothes)$
$R_8 - on(C_6, C_7)$
As mentioned above, apart from new relations and actions being formed, this step could also lead to the need to bring new concepts in the problem – specifically $Bucket, Water, Clothes, Pole$ in this example.

Changing the affordance of the object might mislead the solver because they will perceive the object in the context of the new affordance and the solution affordance will be hidden. The problem would thus require more creativity to solve, requiring the solver to exit the 'box' of context affordances

3.6 Addition of objects for distraction:

This step involves addition of objects or templates whose affordance might interfere with the solution. For example, in the above example problem we could show an object, *ladder* in the problem. This object would trigger a problem template of climbing over the fence using the *ladder*. However, the constraint does not allow the solver to set foot in the other garden.
This modules has not been implemented yet. It is a matter of future work.

4 From formalization to algorithms

In this section, the formalization from the previous work [7] is turned into algorithms for the various steps. The following steps are presented as algorithms: Decomposition (Algorithm 1), Replacement (Algorithm 2) and Affordance Manipulation (Algorithm 3). The Encoding step is currently done manually.

Let:
KB denote the Knowledge Base of objects with their shapes and materials
$C_{problem}$ be a set of concepts for the problem
C_{sol} be a set of solution concepts for the problem
Input : $\langle C_{problem}, C_{sol}, KB \rangle$
for c *in* C_{sol} **do**
  object_parts = find object parts of object c from the KB
  if *size(object_parts) > 1* **then**
    for p *in object_parts* **do**
      $C_{problem}$.add(p)
      C_{sol}.add(p)
    end
    $C_{problem}$.remove(c)
    C_{sol}.remove(c)
  else
end

Algorithm 1: Decomposition

Let:
KB denote the Knowledge Base of objects with their shapes and materials
$C_{problem}$ be a set of concepts for the problem
C_{sol} be a set of solution concepts for the problem
p denote the probability of replacing an object, p < 1
SOM_{shape} denote a self organized map for shapes
$SOM_{material}$ denote a self organized map for materials

Input : $\langle C_{problem}, C_{sol}, KB, p, SOM_{shape}, SOM_{material} \rangle$
for c *in* C_{sol} **do**
  if *replacement_to_be_done(p)* **then**
    $c\_new \leftarrow find\_replacement$(c)
    $C_{problem}$.add(c_new)
    C_{sol}.add(c_new)
    $C_{problem}$.remove(c)
    C_{sol}.remove(c)
  else
end
Function `FIND_REPLACEMENT`(c):
  find *c.shape* and *c.material* from KB
  $potential\_shapes \leftarrow SOM_{shape}(c.shape)$
  $x \leftarrow random.randint(0, size(potential\_shapes))$
  $chosen\_shape \leftarrow potential\_shapes[x]$
  $potential\_materials \leftarrow SOM_{material}(c.material)$
  $y \leftarrow random.randint(0, size(potential\_materials))$
  $chosen\_material \leftarrow potential\_materials[y]$
  $c\_new \leftarrow$ an object with *chosen_shape* and *chosen_material* or an object having a part with *chosen_shape* and *chosen_material* from KB
  return *c_new*
end

Algorithm 2: Replacement

Let:
KB_{uses} denotes the Knowledge Base of normal and creative uses of objects
KB_{PT} denotes the Knowledge Base of problem templates
PT_{sol} denotes the solution problem template
$C_{problem}$ be a set of concepts for the problem
C_{sol} be a set of solution concepts for the problem
$R_{problem}$ be a set of relations
$H_{problem}$ be a set of actions
A_{sol} be the solution affordance of the problem

```
Input  : ⟨KB_uses, KB_PT, PT_sol, C_problem, C_sol, R_problem, H_problem, A_sol⟩
for i in length(C_sol) do
    c_i ← C_sol[i]
    aff_c_i ← find all affordances of c_i from KB_uses and choose one
    if aff_c_i ≠ H_{c_i sol} then
        H_problem.add(aff_c_i)
    [a]else
        for PT in KB_PT do
            if c_i ∈ KB_PT and aff(PT) ≠ A_sol then
                r ← relations involving c_i in PT
                R_problem.add(r)
                h ← actions involving c_i in PT
                H_problem.add(h)
        end
end
```

Algorithm 3: Affordance Manipulation

[a]The current implementation of this process does not perform the 'else' part of the above algorithm. This is because the knowledge base of problem templates has not been gathered yet.

5 Computational Experimentation

Each of the steps for turning a non-creativity problem into a creativity requiring one is now a process which can have multiple outcomes. In the following section, we will showcase our computational experimentation with generating problems which require insight and possibly creativity. To be able to maintain a linear progression, one potential outcome will be chosen after each step, before producing the next step. The multiplicity of outcomes is described in Section 6.

For each problem we manually make the non-creative version of the corresponding classical creativity problem. The classical creativity problems are:

Problem 1 - The two strings problem [4]
Problem 2 - The cardboard problem [2]
Problem 3 - The candle problem [2]

The non-creative version of these problems is made by showing the solution object(s) in the problem description. In our computational experimentation we input the non-creative version of each of these problems and try to reach the corresponding classical creativity problem by applying the transformation process. In this process, we obtain various other creativity problems. For each problem, we have described the non-creativity problem first. Then each step of the transformation process is explained step by step. The output of this process is a creativity problem.

5.1 Problem 1 - Creating The Two Strings Problem

The non-creative version of the two strings problem is stated as follows: *A person is put in a room that has a string and a pendulum hanging from the ceiling. The task is to tie the string and the pendulum together, but it is impossible to reach one while holding the other.*

The solution to this problem is to swing the pendulum and then hold the string and wait for the pendulum to swing within your reach. Similar to the previous problem, to make this an insight problem, we apply the steps of the transformation process with the non-creativity problem as the input.

1. The first step is to encode this problem.
   $C_1 - Pendulum$
   $C_2 - String$
   $C_3 - Ceiling$
   $C_4 - Person$
   $R_1 - hang(Ceiling, Pendulum)$
   $R_2 - hang(Ceiling, String)$
   $R_3 - hold(Person, String)$
   K_1–If $hold(Person, String)$ then $cannot\_hold(Person, Pendulum)$
   $G_{solution} - \{tied(C_1, C_2)\}$
   $C_{problem} = \{C_1, C_2, C_3, C_4\}$
   $R_{problem} = \{R_1, R_2, R_3\}$
   $H_{problem} = \{\}$
   $K_{problem} = \{K_1\}$
   $G_{problem} = \{G_{solution}\}$
   $I = \{C_{problem}, R_{problem}, H_{problem}, G_{problem}, K_{problem}\}$
2. Next we encode the solution of the non-creativity problem.
   $C_{sol} = \{C_1\}$
   $A_{c_1 sol} = swing(Pendulum)$
   $A_{sol} = swing(Pendulum)$
3. The next step is to apply the decomposition step to the solution objects. For this problem, the decomposition step decomposed the object *pendulum* and output two new concepts:
   $C_5 - String$
   $C_6 - Weights$
   $C_{problem} = \{C_2, C_3, C_4, C_5, C_6\}$
   $C_{sol} = \{C_5, C_6\}$
4. We move on to the next step which is replacement. The following results were obtained:
   Possible replacements for C_5: *Shirt, Scarf, Mitten, Rag, Tie, T-shirt, Drapes, Satchel, String*
   Possible replacements for C_6: *Soda_can, Battery, Lock, Spool, Luggage, Screw-driver, Horseshoe, Weights, Bottle*
   We chose *Horseshoe* as the replacement of *Weights* and retained the object *String*. After this step,

$C_5 - String$
$C_6 - Horseshoe$

5. The next step is affordance manipulation. The following results were obtained for this step:
Possible affordances for *String*: *string wrapped around spool, string hanging from ceiling*
Possible affordances for *Horseshoe*: *horse wear horseshoe, horseshoe near forge*
We chose *horseshoe near forge* as the affordance to be shown for *Horseshoe* and *string hanging from ceiling* for *String*. Thus, $Horseshoe$ is shown in this context of affordance and it's affordance to act as a weight for a pendulum is hidden.
The following new concepts and relations are obtained:
$C_7 - Forge$
$R_4 - near(C_7, C_6)$

Now we have a set of concepts, relations, actions, constraints and goal. This is the encoded insight problem. The conversion of this encoded insight problem to text is currently done manually. The creativity variant of the problem will show - *a person in a room with two strings hanging from the ceiling. A horseshoe will be kept near a forge. The task will be to tie the two strings together with the constraint that it is impossible to reach one while holding the other.*

5.2 Problem 2 - Creating The Cardboard problem

The non-creative version of the cardboard problem is stated as follows: *You are asked to attach a piece of cardboard to the loop on the ceiling. There is a hook placed on the table. How do you proceed?*

The solution to this problem is to use the hook to attach the piece of cardboard to the loop. Again, this is not an insight problem. To make this an insight problem, we follow a similar procedure to the previous problems.

1. The encoding of this problem is as follows:
$C_1 - Cardboard$
$C_2 - Hook$
$C_3 - Ceiling$
$C_4 - Loop$
$C_5 - Person$
$R_1 - hang(Ceiling, Loop)$
$G_{solution} - \{hang(Loop, Cardboard)\}$
$C_{problem} = \{C_1, C_2, C_3, C_4, C_5\}$
$R_{problem} = \{R_1\}$
$H_{problem} = \{\}$
$K_{problem} = \{\}$
$G_{problem} = \{G_{solution}\}$
$I = \{C_{problem}, R_{problem}, H_{problem}, G_{problem}, K_{problem}\}$

2. Next we encode the solution of the non-creativity problem.
   $C_{sol} = \{C_2\}$
   $A_{c_1 sol} = attach\_object\_to\_loop$
   $A_{sol} = attach\_object\_to\_loop$
3. The next step is to apply the decomposition step to the solution objects. For this problem, this step gives no new concepts.
4. The following results were obtained from the replacement process:
   Possible replacements for C_2: *Bobby pin, U-Shaped magnet, Belt, Padlock*
   We chose *Belt* as the replacement of *Hook*. After this step,
   $C_2 - Belt$
5. The next step is affordance manipulation. The following results were obtained for this step:
   Possible affordances for *Belt*: *belt inside closet, person wears belt*
   We chose *person wears belt* as the affordance to be shown. Thus, *belt* is shown in this context of affordance and it's affordance to act as a hook is hidden. The following new relation is obtained:
   $R_2 - wear(C_5, C_2)$

The creativity variant of this problem will show - *A person wearing a belt in a room with a loop on the ceiling and a piece of cardboard on a table. The task will be to attach this piece of cardboard to the loop.*

5.3 Problem 3 - Creating The Candle Problem

The non-creative version of the candle problem is stated as follows: *You are given a candle, candle holder, nails, hammer and a box of matches. You are supposed to fix the lit candle unto the wall in a way that does not allow the wax to drip below.* The solution to this problem is to use a candle holder to prevent wax dripping below. The nail is hammered into the wall and is used to fix the candle holder on the wall. You do not need insight to solve this problem. To make this an insight problem, we apply the steps of the transformation process with the non-creativity problem as the input.

1. The encoding of this problem is as follows:
   $C_1 - Candle$
   $C_2 - Candle\_holder$
   $C_3 - Nails$
   $C_4 - Matchbox$
   $C_5 - Matches$
   $C_6 - Wax$
   $C_7 - Wall$
   $C_8 - Table$
   $C_9 - Hammer$
   $R_1 - contains(C_4, C_5)$
   $R_2 - on(C_8, \{C_1, C_2, C_3, C_4\})$
   $G_{solution} - \{on(C_7, C_1), not\_on(C_7, C_6)\}$

$C_{problem} = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8\}$
$R_{problem} = \{R_1, R_2\}$
$H_{problem} = \{\}$
$K_{problem} = \{\}$
$G_{problem} = \{G_{solution}\}$
$I = \{C_{problem}, R_{problem}, H_{problem}, G_{problem}, K_{problem}\}$

2. Next we encode the solution of the non-creativity problem.
   $C_{sol} = \{C_2, C_3\}$
   $A_{c_2 sol} = catch(Wax)$
   $A_{c_3 sol} = attach(Wall, Candleholder)$
   $A_{sol} = catch(Wax)$
3. The next step is to apply the decomposition step to the solution objects. For this problem, this step gives no new concepts.
4. The following results were obtained from the replacement process:
   Possible replacements for C_3: *Thumbtacks, Fork, Hook, Pin, Needle, Screw*
   Possible replacements for C_2: *Bucket, Bin, Pot, Pringles_tube, Matchbox, Kettle, Box, Kleenex*

   For this problem an additional constraint has to be added at the replacement stage. The constraint is that the replacement object must be capable of being pierced. This is important because the goal consists of $on(C_7, C_1)$, which says that the candle must be attached to the wall. This will rule out some replacements objects such as *Bucket, Bin, Pot, Kettle.* Addition of such constraints is done manually in our current implementation but making the process computational is in our future work.
   Thus,
   Possible replacements for C_2: *Pringles_tube, Matchbox, Kleenex, Box, Coaster*
   We chose *Pringles_tube* as the replacement of *Candle_holder* and *Needle* as a replacement for *Nails.* After this step,
   $C_2 - Pringles\_tube$
   $C_3 - Needle$
5. The next step is affordance manipulation. The following results were obtained for this step:
   Possible affordances for $Pringles_tube$: *pringles_tube in bin, person eat_from pringles_tube, pringles_tube on*
   $Possible affordances for Needle$: *needle attached_to ball_of_yarn, needle attached_to spool, thread intertwin*
   We chose *pringles_tube in bin* and *needle attached_to spool* as the affordance to be shown. Thus, *pringles_tube* is shown in this context of affordance and its affordance to catch wax and prevent it from dripping is hidden.
   The following new concepts and relations are obtained:
   $C_{10} - Bin$
   $C_{11} - Spool$
   $R_3 - in(C_{10}, C_2)$
   $R_4 - attached(C_{11}, C_3)$

The creativity variant of the problem will show - *a candle, a box of matches and a needle attached to a spool on a table next to a wall. There will be a* *pringles_tube in a bin. The task will be to fix a lit candle unto the wall in a way that does not allow the wax to drip below.*

6 Results

The strength of the approach presented in this paper lies in the fact that multiple creativity problems can be created when starting from the same non-creative input problems.

In the previous section we chose one path at each step of the transformation process for the ease of explanation. In this section we show how multiple paths are obtained at each step of the transformation process. Table 1 lists the number of paths obtained in our computational experimentation for each of the problem. We also show the number of potential creativity problems that can be created. Figure 4 shows how different output creativity problems are obtained by tracing different paths.

The current knowledge base used for our computational experimentation consists of 497 objects. The knowledge base includes object parts, shapes and material of object parts and alternative uses of these objects. A larger dataset of objects will help improve the results of the replacement process. A larger alternative uses data and a knowledge base of problem templates will improve the results of affordance manipulation process. Both of these will help increase and diversify the output creativity problems.

Table 1. Number of paths at each step of transformation process

	Decomposition	Replacement	Affordance Manipulation	Number of potential problems
Two strings Problem	1	153	2052	2052
Cardboard Problem	0	5	17	17
Candle Problem	0	30	378	378

7 Discussion

After this initial computational experimentation, we conclude that the initial theoretical approach is feasible in terms of generating creativity problems in the future, in large quantities and variants. Regarding the quality of these problems, no evaluation has been provided yet – the authors aim to construct a metric of quality and apply methods of evaluation in future work.

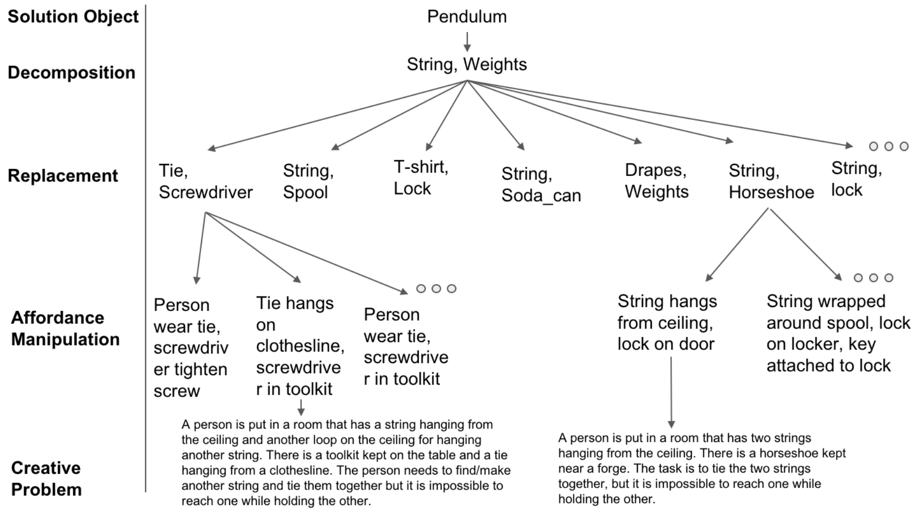

Fig. 4. Multiple paths shown for problem - 1

An interesting question is whether problems created in these manner will be creativity problems or insight problems. Such a question will depend on whether insight is perceived as a quality of a problem, or a quality of the processes of the solver. In our opinion, some solvers may arrive via insightful processes at the answers, while others may do so via creative processes without insight. Still, whether particular problems are more prone to yield insights is a question for which this approach provides high chances of empirical experimentation and answers in the future.

Problem 3 throws light on an important point. The *candle_holder* has two parts that are essential to reaching the goal of the problem. The convex shape is needed for catching the dripping wax and a loop is essential for attaching it to the wall. So, a replacement object for *candle_holder* must have both these parts. Thus, when multiple parts of a solution object are needed to reach the goal, a replacement object must have these parts or similar parts as well (or two replacement objects that can be connected may be necessary). A future insight problem generator should account for this.

Another interesting question is raised by problem 3. The question is - which objects to include in C_{sol}. The term solution objects used in this paper has certain ambiguity to it. For example, in problem 3, *nails* and *candle_holder* were included in C_{sol} as these were considered solution objects. One could argue that *hammer*, *matches* and *matchbox* could also be included in C_{sol} and called solution objects since they are essential objects to reach the solution of the problem. This question still needs answering and will be covered in our future work.

References

1. Wallach, Michael and Kogan, Nathan. Modes of thinking in young children. A study of the creativity-intelligence distinction. new york: Holt, rinehart, and winston, inc., 1965, 357 p. Psychology in the Schools **3**(2), 190–190 (1966). https://doi.org/10.1002/1520-6807(196604)3:2¡190::AID-PITS2310030226¿3.0.CO;2-C
2. Duncker, K.: On problem solving. Psychological Monographs **58**(5, Whole No.270) (1945)
3. Kim, K.H.: Can we trust creativity tests? a review of the torrance tests of creative thinking (ttct). Creativity Research Journal **18**(1), 3–14 (2006). https://doi.org/10.1207/s15326934crj1801_2
4. Maier, N.R.: Reasoning in humans. II. The solution of a problem and its appearance in consciousness. Journal of comparative Psychology **12**(2), 181 (1931)
5. Mednick, S.A.: The remote associates test*. The Journal of Creative Behavior **2**(3), 213–214 (1968). https://doi.org/10.1002/j.2162-6057.1968.tb00104.x
6. Olteţeanu, A.M.: Publications of the Institute of Cognitive Science, vol. 01-2014, chap. Two general classes in creative problem-solving? An account based on the cognitive processes involved in the problem structure - representation structure relationship. Institute of Cognitive Science, Osnabrück (2014)
7. Olteţeanu, A.M.: In: Proceedings of the Workshop on Computational Creativity, Concept Invention, and General Intelligence (C3GI2016). vol. 1767. CEUR-Ws, Osnabrück (2016)
8. Olteţeanu, A.M., Falomir, Z.: Object replacement and object composition in a creative cognitive system. towards a computational solver of the Alternative Uses Test. Cognitive Systems Research **39**, 15–32 (2016)
9. Olteţeanu, A.M., Hass, R.W., Zunjani, F.H.: Computational extraction of metrics and normative data on the alternative uses test on a set of 420 household objects. Behaviour Research Methods (Under Revision)
10. Olteteanu, A.M.: Two general classes in creative problem-solving? an account based on the cognitive processess involved in the problem structure - representation structure relationship. (08 2014)
11. Olteţeanu, A.M.: From simple machines to Eureka in four not-so-easy steps.Towards creative visuospatial intelligence. In: Müller, V. (ed.) Fundamental Issues of Artificial Intelligence, Synthese Library, vol. 376, pp. 159–180. Springer (2016). https://doi.org/10.1007/978-3-319-26485-1_11
12. Royce, J.R.: The nature of human intelligence. j. p. guilford. mcgraw-hill, new york, 1967. xiv + 538 pp., illus. 14.75. Science **162**(3857), 990–991 (1968). https://doi.org/10.1126/science.162.3857.990-a
13. Saugstad, P., Raaheim, K.: Problem-solving and availability of functions. Acta Psychologica **13**, 263 – 278 (1957). https://doi.org/https://doi.org/10.1016/0001-6918(57)90026-4, http://www.sciencedirect.com/science/article/pii/0001691857900264
14. Whitt, J., M. Prentice, N.: The development of children's enjoyment and comprehension of thematic riddles. The American journal of orthopsychiatry **52**, 92–101 (02 1982). https://doi.org/10.1111/j.1939-0025.1982.tb02668.x