A Model for Creative Problem Solving Based on Divergent Production of Solutions*

Luís Macedo[1,2]

1 INTRODUCTION

Looking at problem solving tasks in artistic domains like music composition, writing (screenplays, books), etc., we may see that the main aim of Problem Solving (the achievement of a solution that satisfies as well as possible the problem specification) is not enough. Actually, in these domains the problems are several times the same and may have multiple correct solutions. For example, a XXth century composer may have the same problem of composing a symphony as Beethoven already did a few centuries ago. This means he/she is in front of a previously solved problem, whose resolution can not be attained by remembering and then proposing the old solution, but instead by a process that leads to an original and appropriate solution.

Scientists have the same situation as well: they have to produce better solutions than the ones produced before (if they exist). For example, a researcher, that is looking for a treatment to a disease that already has one but that is not 100% accurate, has to pursue a different but more accurate treatment, as well as a mathematician or a software programmer looks for more efficient and accurate solutions for a problem.

In artistic tasks, both appropriateness and originality of solutions is important. However, in scientific tasks, appropriateness of solutions is the main goal, and originality may come as a consequence of satisfying this goal (a better solution may be unexpectedly different from the other ones). Therefore, generating solutions in this kind of tasks is sometimes an originality and appropriateness guided process: the goal is the construction of correct solutions that stand apart from previous ones. Pursuing surprise, the unexpected, the non-obvious and the aesthetic is explicitly present in artistic tasks but not in scientific tasks, although those properties may be implicitly achieved also in scientific tasks. Actually, when a composer makes a piece of music he/she has the aim of producing a certain emotional state in its listener characterised among other things by surprise. The achievement of a good treatment to a disease may also cause surprise, although that is not the aim.

Creativity is pointed as the main mind skill used in this kind of Problem Solving tasks, whose solutions are characterised by combining appropriateness and originality in a way that causes surprise. Although the creativity phenomenon is far from a complete and consensual understanding, some researchers have addressed the issue of aiding human creativity with computers [5, 6].

The psychologist Guilford [2] has claimed that the exploration of creative solutions is mainly due to the mind ability that he called divergent production. This ability involves the generation of a variety of solutions to a same problem. It is used to solve those kind of problems for which there are multiple correct solutions that may be classified in a continuous evaluation space about their originality and appropriateness. In contrast to divergent production, he considered convergent production as the ability to logically produce the right solution to a given problem.

We suggest a model for Creative Problem Solving, called INSPIRER, based on the idea of divergent production of solutions. The main aim is to provide the system with the property of searching for several alternative and non-obvious solutions that would not be found if a convergent, a logical, or an obvious reasoning process were used.

2 CREATIVE PROBLEM SOLVING AS DIVERGENT PRODUCTION OF SOLUTIONS

Within our approach, the process of constructing a solution to a problem is based on restructuring prior episodic and theoretic knowledge. To do that, we adopt a flexible knowledge representation, splitting knowledge into pieces, in order to recombine or relate them in new, but appropriate ways. To take advantage of its expressiveness, knowledge representation has a graph format: the nodes (called knowledge nodes) represent the knowledge pieces, and the edges represent the relations between those knowledge pieces (Figure 1). These knowledge graphs are a kind of nested graphs as each node may be described by another set of interrelated sub-nodes (i.e., another graph) to which we call the node's internal context, and so on. It is worth of notice that a same node may belong to internal contexts of different nodes. We call the node's external context to its node's neighbourhood graph, i.e., the set of nodes and relations that surround it. With this kind of knowledge representation we may have relations between different grainsize knowledge nodes (Figure 1).

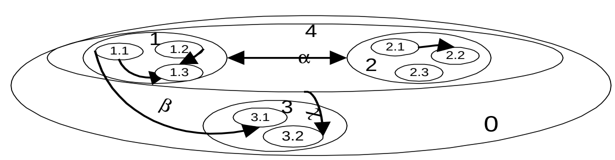

Figure 1. A knowledge graph, which is itself a knowledge node.

---

* Amílcar Cardoso advises this work.

[1] Instituto Superior de Engenharia de Coimbra, Quinta da Nora, 3030 Coimbra, Portugal
[2] Centro de Informática e Sistemas da Univ. de Coimbra, Dept. Eng. Informática da Univ. de Coimbra, 3030 Coimbra, Portugal

ECAI 98. 13th *European Conference on Artificial Intelligence*
Edited by Enri Prade
Young Researcher Paper
Published in 1998 by John Wiley & Sons, Ltd.

With the aim of overwhelming the inefficient processes provided by graph structured representations, we represent a graph by an adjacency matrix and explore some benefits that come from it (e.g.: the matrix A^L contains information about the paths of length L between any pair of nodes of a graph represented by the adjacency matrix A).

Within our approach a problem is just an incomplete solution. The system just has to complete it adding iteratively knowledge nodes to it. Each time a knowledge node is added to a solution, the relations following from it may define missing knowledge nodes that must be filled. When retrieving a knowledge node from prior knowledge structures to fill a (partial or completely) missing node, we apply to each candidate knowledge node a structural similarity metric (similar to the one presented in [4]). This similarity metric takes into account the similarities between the contexts of the missing knowledge node and of the candidate one. The efficient computation of the knowledge nodes' context is improved by performing the following computations with the adjacency matrix A of a knowledge graph: A^1, A^2,..,A^L (the value of L determines the wide of the context). The candidate knowledge nodes are then ranked and one of them is selected. There are several selection criteria, each one corresponding to a different ranking of the knowledge nodes. When choosing a criterion k ($k \in [0, 100]$) for the selection of a knowledge node, one implicitly wishes that the knowledge node to be selected must have a similarity metric value as equal as possible to $\psi\%=(100-k)\%$. Thus, considering the generic knowledge node ranking portrayed in Figure 2, in which p_i is the i^{th} knowledge node in the selection ranking counting from the top (the top knowledge node is p_1) with a similarity metric value λ_i, then such ranking obeys to the following conditions: $|\lambda_i - \psi| \leq |\lambda_{i+1} - \psi|$, with i=0, 1, 2,…,n, and n being the number of candidate nodes. Using a criterion $k \neq 0$ one intends to obtain unexpected combinations of knowledge nodes, although taking a cognitive risk that may lead to bizarre combinations. The greater the value of k is, the more the cognitive risks are, and so, probably, the more original and the less appropriate the solution is. It is worth of notice that if the addition of the selected knowledge node causes incompatibilities with the rest of the solution, an adaptation process is attempted. If it fails then the next knowledge node in the ranking is selected, and so on. This way, the ranking assures a minimum cognitive risk.

The system constructs episodic knowledge (cases) for each Problem Solving session, storing any kind of successful or failure steps that it performs (the adaptations made, the criteria used, etc) in order to provide more successful Problem Solving sessions in the future. Both these reasoning cases and regular cases may be generalised into theoretic knowledge through an abstraction process [1].

Divergent production relies heavily in the knowledge node selection phase described above as it is achieved by repeating the construction of an entire solution for a same problem several times, each time changing the selection criterion. This change may be performed using a different selection criterion to each knowledge node selection process. Exploring the possible combinations of the different criteria for the several knowledge node selections performed on a solution construction process, an extremely great number of different solutions may be achieved to a same problem. This process may be controlled by the user, i.e., he/she may select the criterion to be used in a specific retrieval of a knowledge node, and thus, he/she may control somehow the originality and the appropriateness of the solution. An alternative consists in automatically computing the possible combinations of selection criteria, or even taking into account cases of previous Problem Solving sessions. Some of the various solutions produced may be good, and some others don't. The user may do this evaluation, but it is our aim to make the system do this automatically, using for example knowledge of previous Problem Solving sessions. We have previously applied this idea of divergent production of solutions in planning [3].

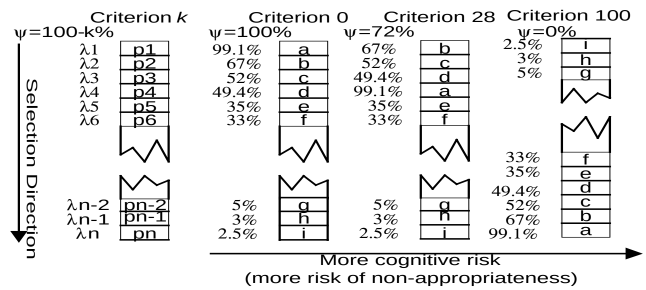

Figure 2. Examples of selection criteria and respective knowledge node rankings.

Convergent production may be approached using selection criterion 0, where no cognitive risks are taken, no originality is pretended, and appropriateness is the only aim.

3 FINAL REMARKS

Problem Solving systems may benefit from divergent production of solutions in two main points. First, some good solutions might be constructed, which would not be so if a logical or an ordinary way of Problem Solving were used. Second alternative solutions might be found to problems that already have one or more solutions.

The approach presented in this paper allows an originality and appropriateness guided control of the Problem Solving process, with the aim of divergently producing solutions.

REFERENCES

[1] R. Bergmann and W. Wilke. On the Role of Abstraction in Case-Based Reasoning. *Proceedings of the Third European Workshop on Case-Based Reasoning*, Lausanne, Switzerland, 1996.

[2] J. Guilford. *Intelligence, creativity and their educational implications*. Robert Knapp, San Diego, CA, 1968.

[3] L. M. Macedo, F. C. Pereira, C. A. Grilo and A.B. Cardoso. A Model for Divergent Production of Plans: its Application in the Music Composition Domain. *Proceedings of Computational Models of Creative Computation*, Dublin, 1997.

[4] L. M. Macedo, F. C. Pereira, C. A. Grilo and A.B. Cardoso. Experimental Study of a Similarity Metric for Retrieving Pieces from Structured Plan Cases: its Role in the Originality of Plan Case Solutions. *Proceedings of the Second International Conference on Case-Based Reasoning*, Providence, USA, 1997.

[5] R. C. Schank. *Explanation Patterns: Understanding mechanically and creatively*. Erlbaum, Hillsdale, NJ, 1986.

[6] T. Veale (ed.). *Computational Models of Creative Computation*. Dublin, 1997.