Seeking Divisions of Domains on Semantic Networks by Evolutionary Bridging

João Gonçalves, Pedro Martins, António Cruz, and Amílcar Cardoso

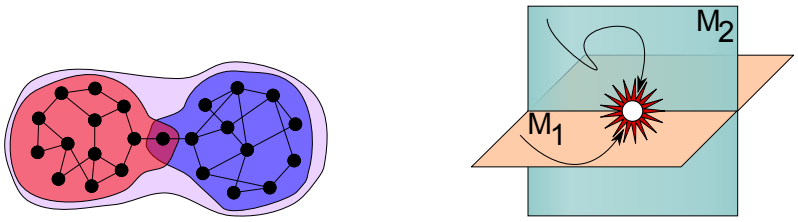
CISUC, Department of Informatics Engineering, University of Coimbra
{jcgonc,pjmm}@dei.uc.pt
antonioc@student.dei.uc.pt
amilcar@dei.uc.pt

Abstract. Computational Creativity systems based on Conceptual Blending (CB) and Bisociation theories operate on input knowledge to reveal seemingly unrelated information. The input spaces or domains can be of various sources and contain vast amounts of knowledge. It is central a process that selects useful building blocks of semantic data that does not narrow the search space of the creative algorithm. It is also vital that the data selection process is of high performance in order to handle a large knowledge base in a useful time. With those objectives in mind, we propose an evolutionary high performance algorithm that extracts two semantic sub-graphs from a knowledge base to be used as building blocks in computational blending processes.

1 Introduction

A creative process can be seen as a form of heuristic search for a construct on a vast semantic space of concepts and domains. In this paper we propose an evolutionary approach inspired by the work of Nagel [6] for selecting two domains from a broader knowledge structure using high performance algorithms. This allows a faster extraction of a more concise representation of the data to be used in computational concept generation techniques, such as CB and Bisociative Knowledge Discovery, among other applications. As the amount of available knowledge dramatically expands each year, high performance algorithms are required to cope with the extraction of new insights, together with growth of multidisciplinary knowledge bases. In [3] Juršič, based on the ABC model by Swanson [11] [10], proposes the CrossBee system for supporting creativity insight in knowledge discovery of literature. In the ABC model, Swanson remarks that reference citations and other bibliographic indications potentially reveal new knowledge, which is not clearly intended neither logically exposed in the literature. That is, ABC exposes the $A \implies B \implies C$ logical consequence, being B the term which relates the remaining terms A and C. Using this idea, CrossBee tries to explore bridging terms linking two apparently disconnected literature domains. In CrossBee, the bridge terms contain relations between two terms, each from a different domain, that were mined from a literature knowledge base. The terms are extracted from various sections in the literature texts, such as bibliography, citations, logical consequences and other references present in the texts. Having the literature containing the terms from A referencing terms regarding B, and

simultaneously the literature C also references terms from B, then a unintended modus ponens inference suggest a hidden relation between A and C. Hence, the system following closely the ABC idea by Swanson.



(a) (b)

Fig. 1: Two domains connected by a single concept (left: adapted from [1]) and the juxtaposition of two frames of reference in Koestler's Bisociation (right: adapted from [4]).

A similar work by Nagel et al. [6] introduces a formalised spreading activation algorithm to identify bridging concepts in a semantic graph (Fig. 1a). The bridging terms interconnect nodes from disjoint semantic domains, following an identical idea to CrossBee. However, the main intention in this case is to juxtapose two apparently unrelated domains through a single term [5]. This notion of pairing two disjoint frames of reference using a singular connection was put forth by Koestler and named Bisociation [4] in his work, *The Act of Creation* [4]. There, Arthur Koestler attempts to describe creative behaviour present in humour, arts and science. This model consists on "the perceiving of a situation or idea ... in two self-consistent but habitually incompatible frames of reference (M_1 and M_2)" as shown in Fig. 1b. For instance, in humour, bisociations could relate the unforeseen transformation from one meaning to another [7]. In their paper, Nagel explores a search space, defined using a bisociation score, which rates individually each bridging node subdividing the semantic graph in two completely disjoint sets. However, their approach does not allow a tolerance of the intersection between the two domains. Thus, it is in a sense a hard margin solution and in our opinion, it terminates the search prematurely in real world problems, as underlined in their conclusion. However, their highly formalised work served as a basis for our present approach. In the following section, we offer our evolutionary approach in the form of a Genetic Algorithm (GA), inspired by the work of Nagel.

2 Algorithm

The purpose of the algorithm is to identify two partially overlapping sub-graphs S_0 and S_1 of a larger semantic graph S. Given their structure, interrelations and arrangement within the larger graph, we believe the sub-graphs could be seen as domains of knowledge in the broader semantic graph.

The cardinality of each graph structure is identified by the symbol #. Thus, $\#S$ is the number of nodes existing in the graph S and we denote this quantity as the size of the graph. Each sub-graph represents a network of highly interconnected nodes, which if belonging to a semantic graph, could represent a domain of related concepts [1]. Both sub-graphs share at least a single node N_b, the bridge node, and the sub-graphs should be balanced [6] regarding a split through the bridge node. The size of both sub-graphs $\#S_0$ and $\#S_1$ should be maximized, with only the condition of $S_0 \cap S_1 = \{N_b\}$. Then, a unique path will flow from one sub-graph to the other through the bridge node which has the unique index $b \in \{1 \dots \#S\}$. When this happens, the unique bridge node may represent a possible bisociation which juxtaposes one domain (sub-graph) into the other (Fig. 1b).

A degree d_i of a node N_i with $i \in \{1 \dots \#S\}$ represents the number of incident edges to that given node. Nodes with $d = 2$ represent a single relationship between two concepts, being these the most likely candidates for a bridge node [6]. The reasoning behind this choice is the interest in mapping two domains over a single and clear semantic relationship, through the bridge node. In this case, any concept from one sub-graph can be projected onto any other concept from the other sub-graph, offering a foundation for further transformations of concepts using processes from bisociation and CB [7].

Otherwise nodes with $d \geq 3$ map a more vague set of relations between connected concepts. Intuitively, the view of an idea in two distinctly but opposing views is more fine tuned to two set of concepts (two domains) connected by a single node [6]. A simple example which demonstrates this idea is seen in Fig. 1a. On the other hand, highly interconnected nodes express a deeply related network of information or domain. Using the above criteria, the discrete function which rates the optimality (fitness) of the bridge node N_b is defined in (1):

$$f(S_0, S_1, d_b) = \begin{cases} \frac{1}{\alpha^{\frac{|\#S_0 - \#S_1|}{\#S_0 + \#S_1}} + 1} \cdot \log(\#S_0 + \#S_1) \cdot 2^{-\beta(d_b - 2)}, & \text{if } d_b \geq 2 \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The fitness function receives as arguments the sub-graphs S_0, S_1 and the degree of the bridge node N_b as the variable d_b. The parameter α controls how similar in size the sub-graphs S_0 and S_1 are required to be, with increasing α exhibiting greater size similarity. The parameter β is used to control the penalisation given to bridge nodes with a degree $d > 2$, with the penalisation exponentially proportional to the value of β. If the degree of node being rated is 1, that is, a terminal node with a single relation, then f is set to 0 in order to prevent the GA to select terminal nodes as bridge.

Globally, a GA evolves a population of chromosomes where each chromosome represents a bridge node and two sub-graphs of the initial semantic network. Each time a new individual is created with a given bridge node, a breadth first search is executed starting in the latter node and into neighbouring nodes. The dual diffusion process (a sort of spreading activation) progresses radially until a given expansion depth is reached when both sub-graphs intersect, or all the

nodes in the network have been explored. From this expansion, we calculate a performance score representing this graph division which will represent the chromosome in the global fitness landscape. Thus, the GA evolves subdivisions of the semantic network aiming to find the bridging node that maximises the size of the sub-graphs found.

A GA was chosen in order to relax the search for global optima. Given a specific graph, the search space can be of very high complexity which is likely when handling large semantic networks. The optimisation task is represented by the fitness function and the genetic operators embody the transformations that move the bridge node through the semantic graph. By using a GA, it is easy to parallelise the search algorithm with the following steps:

- fitness functions are evaluated in parallel threads;
- new individuals which will define the next generation are created in parallel threads;
- the generation of random chromosomes for the initial population and when the GA stagnates is executed in parallel threads.

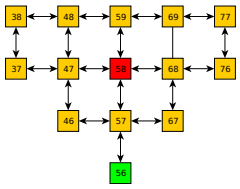


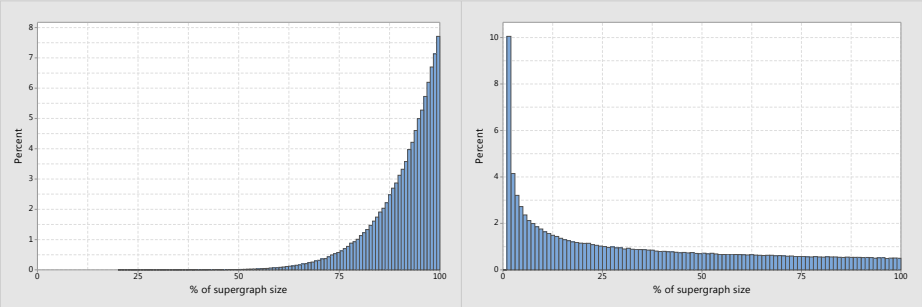
Fig. 2: A combination of random translation movements (a random walk) starting at the red node and given time, eventually leads to the green node. Best viewed in colour.

In our GA we use entirely the mutation operator. Using Fig. 2 as an example, it can be seen that the fundamental operation required to traverse the search space (the graph) is the translation of the bridge node. Thus, the search algorithm samples stochastically (walks randomly) the locations in the graph that maximise the fitness function. The mutation changes the location of the bridge node using the connected nodes (neighbourhood) recursively. In order to allow the bridge node to jump to distant sections of the graph, we use a quadratic random number generator. The probability function controlling the locality behaviour of this process decreases with the increasing number of jumps the bridge node is allowed to make. This number is given in (2).

$$j(r, s) = r^{\gamma} s + 1 \tag{2}$$

The parameter $r \in [0, 1[$ represents a previous output of an uniform pseudo random number generator. The number of nodes $\#S$ in the graph S is represented by the parameter s. The parameter $\gamma \in]0, +\infty[$ controls the average jump distance (translation deepness) the mutation applies to the bridge node.

If $\gamma >> 1$, the mutation tends to move the bridge node towards nearby nodes (Fig. 3b). When $\gamma \to +\infty$, the translation tends to move each bridge node to



(a) $\gamma = 0.1$ (b) $\gamma = 2$

Fig. 3: Probability Density Functions for two values of γ when using the bridge jump mutation function (2).

one of the connected neighbours. For $\gamma \in]0, 1[$, particularly nearby 0, tends to translate the bridge node to distantly connected nodes (Fig. 3a) and, in a sense, promote a random search of the fitness landscape. The case where $\gamma = 1$ forces a straightforward stochastic search whereby each bridge node can be moved to any other node position of the domain S with constant probability.

After the mutation has been applied to the chromosome, the expansion of sub-graphs from the full domain is executed again from the newly calculated bridge node. When this completes, two new sub-graphs divide part (or all) of the domain with roots at the bridge node and depth $d_{(0...1)}$. This expansion is a type of breadth first search starting at the node N_b so that the first nodes to explore are the nearby nodes. The main idea behind this reasoning is shown in Algorithm 1. Using two pairs of open (to expand) and closed (already expanded) nodes, the algorithm inserts the visited nodes in the two sub-graphs which will represent the sub-domains S_0 and S_1. An example of the process is seen in (Fig. 5) from where two different coloured sub-graphs emerge (Fig. 5). The function `nodes`(S_i) returns the set of nodes $\{N_i\}, i \in \{0, \dots \#S_i - 1\}$ in the sub-graph $\{N_k\}, k \in \{0, 1\}$. The variables O_i and C_i represent respectively the set of open (to visit) and closed (already visited) nodes related to the sub-graph i. The function `split`(N_b, S) divides the neighbourhood of the bridge node N_b in two sets of nodes as evenly as possible. When the neighbourhood of N_b is odd, a randomly chosen set S_i receives the additional node so that in the worst case, the difference of cardinality between S_0 and S_1 is 1.

The function `expandOneLevel`(S_i, O_i, C_i, S) cycles through all the nodes in the set O_i, inserts each visited node in the set C_i and in the sub-graph S_i, including the connected edges. Then it extracts the neighbourhood of each visited node and inserts the neighbour nodes in the set O_i, so that in the next iteration of the function `createSubgraphs`() the algorithm expands from the current neighbourhood of O_i. Thus, the function `expandOneLevel()` executes an equivalent single iteration of a breadth first search at the same deepness level. All the nodes and edges are obtained from the graph S.

Algorithm 1 Function createSubgraphs()

function CREATESUBGRAPHS(N_b, S)
  $\{S_0, S_1\} \leftarrow \texttt{split}(N_b, S)$
  $O_0 \leftarrow \texttt{nodes}(S_0)$
  $O_1 \leftarrow \texttt{nodes}(S_1)$
  $C_0 \leftarrow N_b$
  $C_1 \leftarrow N_b$
  $I \leftarrow \emptyset$
  repeat
    $\texttt{expandOneLevel}(S_0, O_0, C_0, S)$
    $\texttt{expandOneLevel}(S_1, O_1, C_1, S)$
    $I \leftarrow S_0 \cap S_1$
  until $\frac{\#I}{\#S_0+\#S_1} \geq \tau \vee \#S_0 = 0 \vee \#S_1 = 0$
end function

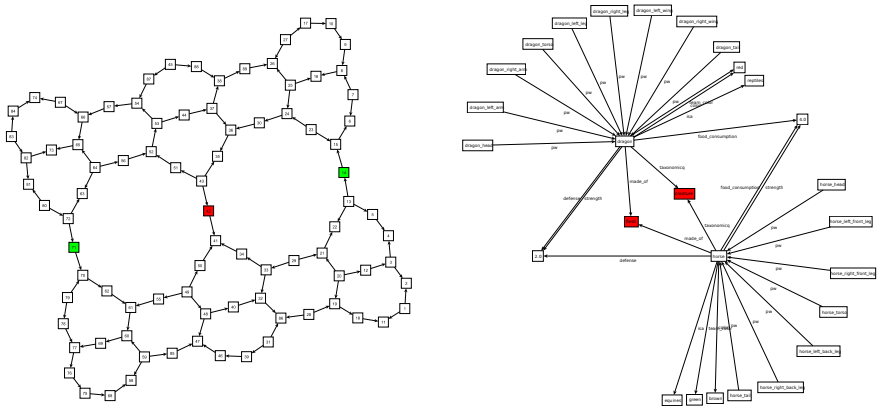
Starting at the bridge node N_b the expansion grows radially throughout the connected nodes, creating the sub-graphs while visiting the explored nodes until the sub-graphs intersect. For a graph in structure similar to Fig. 1a, the sub-graphs are expected to intersect only in the bridge node. However, in real cases, while the expansion is taking place, the intersection can suddenly show a small amount of nodes when in comparison with the size of both sub-graphs S_0 and S_1. When this happens, the algorithm may not be able to find a clean (and useful) division of the graph S.

Using a similar idea to Soft Margin in [2], we include the parameter $\tau \in \mathbb{R}_0^+$ to allow more than one bridge node connecting the sub-graphs S_0 and S_1. With $\tau = 0$, the intersection I between the sub-graphs is allowed to contain only the bridge node, as in Nagel. When the ratio of the intersection to both sub-graphs size increases above τ, the algorithm stops and returns the most recently created sub-graphs starting at node N_b. In sum, τ represents the trade-off between the penalization of highly interconnected sub-graphs and the maximisation of the size of those sub-graphs.

Consider the following example: after a 5 level expansion, the intersection of the two sub-graphs with a size of 2000 nodes each, suddenly increases from 1 (the bridge node only) to 100. This means that the fifth iteration raised the sub-graphs size to intersection ratio from $\frac{1}{2000+2000} = 0.025\%$ to $\frac{100}{2000+2000} = 2.5\%$, a $100\times$ fold increase. It may happen that the sub-graphs contain useful knowledge and for this reason, they should not be discriminated. Depending on the situation at hand, the parameter τ chosen to control the ratio may or not be significant.

3 Results and discussion

The feasibility of our algorithm was tested using three semantic graphs. The first, shown in Fig. 4a, was generated exclusively to test the theory supporting the algorithm. It contains 89 nodes and 106 unlabelled directed edges. The second is from the Horse-Dragon experiment, a well known semantic graph in Conceptual



(a) 89 node graph. (b) Horse-Dragon graph.

Fig. 4: Structure of the 89 node and Horse-Dragon graphs. The highest rated nodes are shown coloured. Best viewed in colour.

Blending, supplied by the authors of [8]. This semantic graph contains 32 relations between 32 attributes of the animals horse and dragon, such as physical parts, health resistance and some taxonomic properties (Fig. 4b). The last is the Perception semantic graph from [9]. The author of Perception defines his knowledge base as a summary of manually annotated common sense concepts and their relations. It contains 3892 nodes and 345463 edges. Unless otherwise stated, the parameters used for the graph division algorithm were $\tau = 0$, $\alpha = 4$, $\beta = 4$ and $\gamma = 2$. The GA evolved a population of 10^3 chromosomes with a mutation rate of 100%, no crossover and a maximum number of 10^3 evolved generations.

Before the experiments, we validated the algorithm with a 111 node graph (Fig. 5) containing 188 directed edges. After the conclusion of the GA, the two sub-graphs S_0 (green) and S_1 (cyan) are juxtaposed through the bridge node with the label 56. The GA stopped when the intersection between the sub-graphs included the nodes labelled 17, 95 and the bridge node with label 56. Afterwards, we proceeded with the experiments on the three semantic graphs.

Table 1: Fitness scores f for the four highest rated bridge nodes of the *Horse-Dragon* semantic graph.

f	degree(N_b)	label(N_b)	$\#S_0$	$\#S_1$
3.989	2	creature	15	15
3.989	2	flesh	15	15
0.249	3	4	15	15
0.249	3	2	15	15

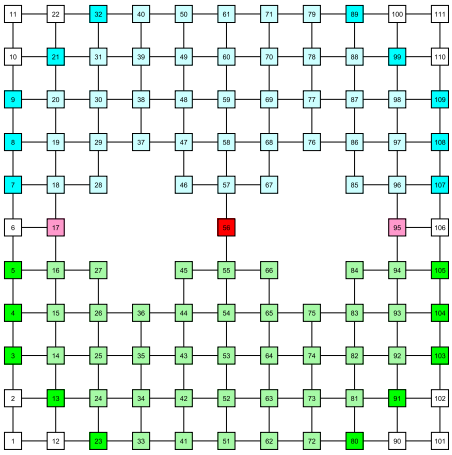


Fig. 5: Optimal sub-graph configuration of the 111 node test graph showing bridge node (red), intersection nodes (pink) and the two created sub-domains (green and cyan) each with a depth of 8 nodes starting at the bridge node. Calculated with intersection tolerance $\tau = 0$. Best viewed in colour.

Our algorithm reported a high amount of possible bridge nodes in the 89 node graph (Fig. 4a). From those, the 3 highest scored nodes are shown coloured, where the node in red scored 50% higher than the green nodes.

The Horse-Dragon [8] semantic graph is shown in Fig. 4b with the four highest rated chromosomes presented in Table 1. The majority of the nodes are terminal ($d = 1$) where a small number of highly interconnected nodes ($d \geq 2$) are clearly visible (Fig. 4b) labelled as *horse* and *dragon*. The best chromosomes generated by the GA produced were the two pairs of sub-graphs with each pair linked by the nodes with label *creature* and *flesh*.

In order to study our algorithm with a more complex and practical problem, we researched the Perception [9] knowledge base with two experiments. For the first, we did a study regarding the effect of τ in the size of the two sub-graphs. As shown in Table 2, the parameter τ highly influences the size of both sub-graphs. Having the Perception graph 3892 nodes, for certain τ values, one or both of the sub-graphs contain more than half the nodes from Perception. Therefore, a compromise has to be made so that both sub-graphs do not drastically intersect between themselves. However, both should contain a minimum amount of knowledge and relations to be useful for CB and Bisociative Knowledge Discovery. From Table 2, an interesting improvement in the size of the sub-graphs happens when τ changes from 0.05 to 0.1. With $\tau \geq 0.5$ the fitness function f does not increase, implying that the limit of the graph has been reached as the size of both sub-graphs are equal and maximum.

For the second experiment, we set $\tau = 0.1$ in order to limit the intersection between the two sub-graphs to 10% of their combined size. A list of results is present in Table 3, with all the bridge nodes having degree of 2. The fitness

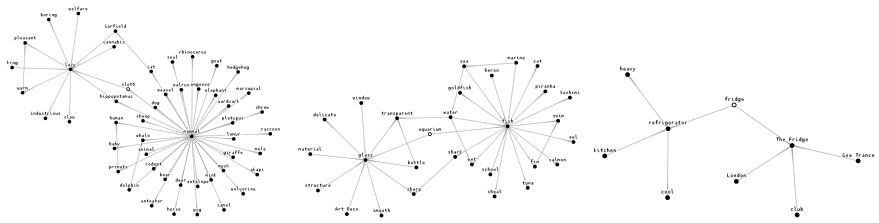
Table 2: Fitness scores f of the highest rated bridge nodes for the Perception semantic graph when varying τ.

τ	f	degree(N_b)	label(N_b)	$\#S_0$	$\#S_1$	$\#(S_0 \cap S_1)$
0.00	4.35	2	panther	47	46	0
0.05	7.07	2	Jerry Springer	586	586	7
0.10	7.70	2	Times	1919	1991	98
0.15	8.19	2	Jesus Christ	2173	2199	119
0.20	8.27	2	Pulp Fiction	2121	2132	186
0.25	8.30	2	wrestling	2264	2281	291
0.33	8.63	2	ashes	3039	3054	943
0.50	8.95	2	emerald	3868	3868	3868
0.75	8.95	2	chromosome	3868	3868	3868

Table 3: Fitness score f for 22 bridge nodes, from the Perception semantic graph with $\tau = 0.1$.

f	degree(N_b)	label(N_b)	$\#S_0$	$\#S_1$	$\#(S_0 \cap S_1)$
7.70	2	Times	1919	1991	98
7.52	2	Athens	1474	1522	62
7.07	2	Jerry Springer	586	586	7
7.03	2	sloth	1242	1177	32
6.98	2	herd	714	729	4
6.96	2	fox	984	1031	27
6.78	2	pilot	529	522	11
5.79	2	aquarium	949	820	13
4.99	2	fridge	427	514	1

declines with the increasing unbalancing between the sub-graphs S_0 and S_1. In Figs (6) we show some of the structures of the nearby connections. It is interesting to observe the relations between connected domains for the "sloth", "aquarium" and "fridge" nodes. We find the last case peculiar, as we did not knew of a heavy and cool (or cold?) trance band. For the remaining bridge nodes, we leave their insight to the reader's judgement.



(a) "sloth" bridge node. (b) "aquarium" bridge node. (c) "fridge" bridge node.

Fig. 6: Three examples of bridging nodes and their neighborhood present in Table 3.

4 Conclusions

In this work we proposed an evolutionary approach to support computational concept generation systems and knowledge discovery. Building on the work of Nagel [6], our work allows the discovery of knowledge divisions in large semantic graphs and the identification of possible key concepts which interconnect the sub-graphs. The algorithm supports various parameters to fine tune this division process in accordance with real world knowledge bases, so that different relations between knowledge domains can be researched and hopefully, give possibility to

new insights between those domains. Lastly, by using a high performance algorithm, the exploratory process can be done in useful time. In the future we expect to improve our approach by experimenting with graph similarity. It would also be interesting to use a form of feedback loop by integrating a concept generating algorithm. This way, the system would direct its search towards bridging nodes and sub-graphs that would be more useful to the task that follows the GA.

Acknowledgements. The authors acknowledge the financial support from the iCIS (Intelligent Computing in the Internet of Services) project (CENTRO-07-0224-FEDER-002003) and from the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the ConCreTe FET-Open project (grant number 611733) and the PROSECCO FET-Proactive project (grant number 600653).

References

1. Michael R. Berthold, editor. *Bisociative Knowledge Discovery*. Volume 7250 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012.
2. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
3. Matjaž Juršič, Bojan Cestnik, Tanja Urbančič, and Nada Lavrač. Cross-domain literature mining: Finding bridging concepts with crossbee. In *Proceedings of the 3rd International Conference on Computational Creativity*, pages 33–40, 2012.
4. Arthur Koestler. *The Act of Creation*. New York:Macmillan, 1964.
5. Tobias Kötter, Kilian Thiel, and Michael R Berthold. Domain bridging associations support creativity. 2010.
6. Uwe Nagel, Kilian Thiel, Tobias Kötter, Dawid Piatek, and Michael R. Berthold. Towards discovery of subgraph bisociations. In Michael R. Berthold, editor, *Bisociative Knowledge Discovery*, volume 7250 of *Lecture Notes in Computer Science*, pages 263–284. Springer Berlin Heidelberg, 2012.
7. F Pereira. *Creativity and artificial intelligence: a conceptual blending approach*. Berlin: Mouton de Gruyter, 2007.
8. Paulo Ribeiro, Francisco C. Pereira, Bruno Marques, Bruno Leitão, and Amílcar Cardoso. A model for creativity in creature generation. In *Proceedings of the 4^{th} Conference on Games Development (GAME ON'03)*. EuroSIS / University of Wolverhampton, 2003.
9. Tom De Smedt. *Modeling Creativity: Case Studies in Python*. Uitgeverij UPA University Press Antwerp, 2013.
10. Don R Swanson. Fish oil, raynaud's syndrome, and undiscovered public knowledge. *Perspectives in biology and medicine*, 30(1):7–18, 1986.
11. Don R Swanson. Two medical literatures that are logically but not bibliographically connected. *Journal of the American Society for Information Science*, 38(4):228–233, 1987.