

A semantic network-based evolutionary algorithm for modeling memetic evolution and creativity

Atılım Güneş Baydin · Ramon López de Mántaras · Santiago Ontañón

Received: date / Accepted: date

Abstract We introduce a novel evolutionary algorithm (EA) with a semantic network-based representation. For enabling this, we establish new formulations of EA variation operators, crossover and mutation, that we adapt to work on semantic networks. The algorithm employs commonsense reasoning to ensure all operations preserve the meaningfulness of the networks, using ConceptNet and WordNet knowledge bases. The algorithm can be classified as a novel memetic algorithm (MA), given that (1) individuals represent pieces of information that undergo evolution, as in the original sense of memetics as it was introduced by Dawkins; and (2) this is different from existing MA, where the word “memetic” has been used as a synonym for local refinement after global optimization. For evaluating the approach, we introduce an analogical similarity-based fitness measure that is computed through structure mapping. This setup enables the open-ended generation of networks analogous to a given base network.

Keywords Evolutionary computation · Memetic algorithms · Memetics · Analogical reasoning · Semantic networks

Atılım Güneş Baydin (✉)
Hamilton Institute & Department of Computer Science
National University of Ireland Maynooth
Co. Kildare, Ireland
E-mail: atilingunes.baydin@nuim.ie

Ramon López de Mántaras
Artificial Intelligence Research Institute, IIIA - CSIC
Campus Universitat Autònoma de Barcelona
08193 Bellaterra, Spain
E-mail: mantaras@iia.csic.es

Santiago Ontañón
Department of Computer Science
Drexel University
3141 Chestnut Street, Philadelphia, PA 19104, USA
E-mail: santi@cs.drexel.edu

1 Introduction

We introduce an evolutionary algorithm (EA) that generates semantic networks under a fitness measure based on information content and structure. This algorithm is, to the best of our knowledge, the first instance in literature where semantic networks are created via an evolutionary optimization process, with specially developed structural variation operators respecting the semantics of commonsense relations.

We demonstrate our approach via a fitness function measuring analogical similarity to a given base network. This is particularly interesting from an analogical reasoning perspective, because it enables us to spontaneously generate analogical mappings *and* novel analogous cases, in contrast with existing algorithms capable of generating *only* the mapping between two given cases.

Seeing the evolutionary optimization of information represented within semantic networks as an implementation of the idea of “memes” in cultural evolution, we pose this algorithm as a novel type of memetic algorithm (MA). In this designation, we use the term “memetic” in a different technical sense from existing models classified as MA, and with an implication closer to the original meaning as it was first introduced as a metaphor by Dawkins [8] in his book *The Selfish Gene* and later popularized by Hofstadter and Dennett [21].

This is due to several reasons.

Within the existing field of MA, one models the effects of cultural evolution as a local refinement process for each individual, running on top of a global, population-based, optimization [30]. So, the emphasis is on the local refinement of each individual due to

memetic evolutionary factors¹. In algorithmic terms, this results in a combination of population-based global search with a local search step run for each individual. Thus, the only connection of the existing work in MA with the idea of “memetics” is using this word as a synonym for “local refinement of candidate solutions”.

In contrast, the emphasis in our approach is directly on the memetic evolution itself, given

1. it is the units of information (represented as semantic networks) that are undergoing variation, reproduction, and selection, exactly as in the original metaphor by Dawkins [8];
2. we have variation operators developed specific for this knowledge representation-based approach, respecting the semantics and commonsensical correctness of the evolving structures; and
3. the whole process is guided by a fitness measure that is defined as a function of some selected set of features of the knowledge represented by each individual.

The article is organized as follows. In Sect. 2 we provide background information on the subjects of evolution, creativity, and culture, followed by a brief review existing models of graph-based EA, to enable a discussion of how our contribution is related with existing work in the field. In Sect. 3, we go over a detailed description of our algorithm, including details of representation and the newly introduced variation operators specific for semantic networks. We introduce the analogical similarity-based fitness measure in Sect. 4, presenting results of experiments with the spontaneous generation of analogies. Sect. 5 ends the article with concluding remarks and a discussion of limitations and future directions for our approach.

2 Background

2.1 Evolution, creativity, and culture

Following the success and explanatory power of evolutionary theory in biology, insights about the ubiquity of evolutionary phenomena have paved the way towards an understanding that these processes are not necessarily confined to biology. That is to say, whenever one has a system capable of exhibiting a kind of variation,

heredity, and selection, one can formulate an evolutionary account of the complexity observed in almost any scale and domain. This approach is termed *Universal Darwinism*, generalizing the mechanisms and extending the domain of evolutionary processes to systems outside biology, including economics, psychology, physics, and even culture [3, 9].

Within this larger framework, the concept of *meme* first introduced as a metaphor by Dawkins [8] as an evolving unit of culture² analogous to a *gene*, hosted, altered, and reproduced in minds, later formed the basis of the approach called memetics³.

Popularized by Hofstadter and Dennett [21], the explanation found itself use in cultural and sociological studies. For example, Balkin [1] argues that ideologies can be explained using a meme-based description, produced through processes of cultural evolution and transcending the lives of individuals. This evolutionary approach to ideology also enables genetic-inspired descriptions of cultural phenomena, such as *ideological drift*. Similarly, creativity, as an integral part of culture, has also been addressed by these studies at the intersection of evolution and culture [15]. At this point, one has to clarify in which of the two highly related, but conceptually different, ways one uses the notions of “evolution” and “creativity” together.

The first point of view basically discusses the role of arts and creativity in the general framework of classical evolutionary biology, considering the provided advantages for adaptation and survival. All known societies enjoy creative pursuits such as literature, music, and visual arts; and there is evidence from the field of archaeology that this interest arose relatively early in the development of human species. Combined with the knowledge that a sense of aesthetic is also encountered in several other species of the animal kingdom, there is ample evidence to consider an “evolutionary basis” for creativity [26].

Alternatively, inspired by the insight that evolutionary processes are not confined to biology, and using evolutionary theories of sociocultural change, one can consider that culture itself is possibly recreated through evolutionary processes occurring in the abstract environment of thoughts, concepts, or ideas. An example for this kind of interpretation in social sciences is the evolutionary epistemology theory put forth by Campbell [3].

¹ From a biological perspective, this sense emphasizes the effect of society, culture, and learning on the survival of individuals on top of their physical traits emerging through genetic evolution. An example would be the use of knowledge and technology by the human species to survive in diverse environments, far beyond the physical capabilities available to them solely by the human anatomy.

² Or, information, idea, or belief.

³ Quoting Dawkins [8]: “*Examples of memes are tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches. Just as genes propagate themselves in the gene pool by leaping from body to body via sperms or eggs, so memes propagate themselves in the meme pool by leaping from brain to brain...*”

Surely, a unifying approach considering all types of evolution is also possible, studying it as a general phenomenon applying at different levels to both physical and cultural systems. Within the creativity field, this kind of approach is taken by Skusa and Bedau [37], who study the processes occurring in systems exhibiting biological or cultural evolution. They base their insights on their work on evolutionary activity statistics for visualizing and measuring diverse systems [2]. Similarly, Gabora and Kaufman [15] investigate the issues at the intersection of creativity and evolution, from both biological and cultural senses.

A similar dichotomy also exists in the interpretation of the role of evolutionary algorithms in computational creativity.

Researchers realize that EA can be applied to computational creativity problems, considering them as a new area of complex and difficult technical problems where they can employ the proven power of EA as a black-box optimization tool.

Again, as in the case of sociocultural evolution, one can also consider the creativity process itself as taking place through evolutionary processes in an abstract “creativity space”. In cases where evidence for an underlying evolutionary process can be spotted (as in the case of cultural evolution), in addition to providing solutions to difficult problems, one can also consider the simple but powerful explanatory power of evolutionary theory for the seemingly complex task of creativity.

The work that we present in this article is open to both interpretations. In addition to being a technique for the generation of semantic networks for a given creativity task, we can also use it—due to its memetic interpretation—to model the evolution of human culture through passing generations.

2.2 Graph-based evolutionary algorithms

There are several existing algorithms using graph-based representations for the encoding of candidate solutions in EA [29]. The most notable work among these is genetic programming (GP) [22], where candidate solutions are pieces of computer program represented in a tree hierarchy. The trees are formed by *functions* and *terminals*, where the terminal set consists of variables and constants, and the function set can contain mathematical functions, logical functions, or functions controlling program flow, specific to the target problem.

In parallel distributed genetic programming (PDGP) [36], the restrictions of the tree structure of GP are relaxed by allowing multiple outputs from a node, which allows a high degree of parallelism in the evolved programs. In evolutionary graph generation (EGG) [6] the

focus is on evolving graphs with applications in electronic circuit design. Genetic network programming (GNP) [25] introduces compact networks with conditional branching and action nodes; and similarly, neural programming (NP) [40] combines GP with artificial neural networks for the discovery of network structures via evolution.

The use of a graph-based representation makes the design of variation operators specific to graphs necessary.

In works such as GNP, this is facilitated by using a string-based encoding of node names, types, and connectivity, permitting operators very close to their counterparts in conventional EA; and in PDGP, the operations are simplified by making nodes occupy points in a fixed-size two-dimensional grid.

Our approach in this article, on the other hand, is closely related with how GP handles variation.

In GP crossover operation, two candidate solutions are combined to form two new solutions as their offspring. This is accomplished by randomly selecting *crossover fragments* in both parents, deleting the selected fragment of the first parent and inserting the fragment from the second parent. The second offspring is produced by the same operation in reverse order. An important advantage of GP is its ability to create nonidentical offspring even in the case where the same parent is selected to mate with itself in crossover⁴.

In GP, there are two main types of mutations: the first one involves the random change of the type of a function or terminal at a randomly selected position in the candidate solution; while in the second one an entire subtree of the candidate solution can be replaced by a new randomly created subtree.

What is common within GP related algorithms is that the output of each node in the graph can constitute an input to another node. In comparison, for the semantic network-based representation that we will introduce, the range of connections that can form a graph of a given set of concepts is constrained by commonsense knowledge, that is to say, the relations have to make sense to be useful. To address this issue, we introduce new crossover and mutation operations for memetic variation, making use of commonsense reasoning [18, 31] and adapted to work on semantic networks.

⁴ This is in stark contrast with approaches such as GA, where a crossover operation of identical parents would yield identical offspring due to the linear nature of the representation

3 The algorithm

Our algorithm, outlined in Algorithm 1, proceeds similar to conventional EA, with a relatively small set of parameters.

Algorithm 1 Procedure for the novel semantic network-based memetic algorithm. Refer to Table 1 for an overview of involved parameters.

```

1: procedure MEMETICALGORITHM
2:    $P(t = 0) \leftarrow \text{INITIALIZEPOPULATION}(Size_{pop},$ 
    $Size_{network}, Score_{min}, Count_{timeout})$ 
3:   repeat
4:      $\phi(t) \leftarrow \text{EVALUATEFITNESSES}(P(t))$ 
5:      $N(t) \leftarrow \text{NEXTGENERATION}(P(t), \phi(t), Size_{pop},$ 
    $Size_{toun}, Prob_{win}, Prob_{rec}, Prob_{mut}, Score_{min},$ 
    $Count_{timeout})$ 
6:      $P(t + 1) \leftarrow N(t)$ 
7:      $t \leftarrow t + 1$ 
8:   until stop criterion
9: end procedure

```

Descriptions of initialization, fitness evaluation, selection, and memetic variation steps are presented in detail in the following sections. The parameters affecting each step of the algorithm, along with their explanations, are summarized in Table 1.

3.1 Semantic networks

Semantic networks are graphs that represent semantic relations between concepts. In a semantic network, knowledge is expressed in the form of directed binary relations, represented by edges, and concepts, represented by nodes. This type of graph representation has found use in many subfields of artificial intelligence, including natural language processing, machine translation, and information retrieval [38].

Figure 1 shows a graph representation of a simple semantic network. In addition to the graphical representation, we also adopt the notation of $\text{IsA}(\text{bird}, \text{animal})$ to mean that the concepts `bird` and `animal` are connected by the directed relation $\text{IsA}(\cdot, \cdot)$, i.e. “bird is an animal”.

An important characteristic of a semantic network is whether it is definitional or assertional: in definitional networks the emphasis is on taxonomic relations (e.g. $\text{IsA}(\text{human}, \text{mammal})$) describing a subsumption hierarchy that is true by definition; in assertional networks, the relations describe instantiations and assertions that are contingently true (e.g. $\text{AtLocation}(\text{human}, \text{city})$) [38]. In this study, we combine the two approaches for increased expressivity. As such, semantic networks provide a simple yet powerful means to represent the “meme”

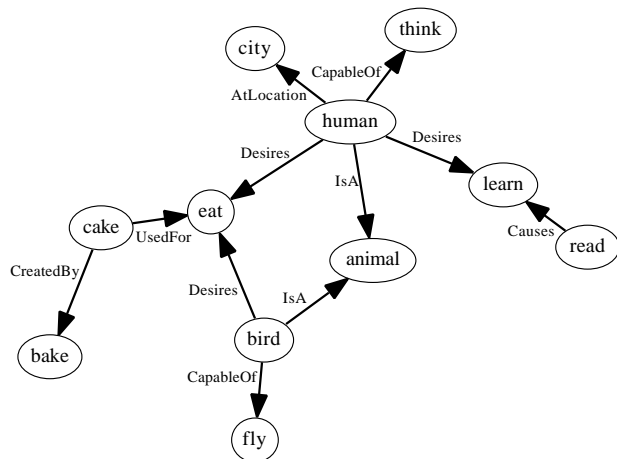


Fig. 1: A semantic network with 11 concepts and 11 relations.

metaphor of Dawkins as data structures that are algorithmically manipulatable, allowing a procedural implementation of memetic evolution.

3.2 Commonsense reasoning

A foundational issue that comes with our approach is the problem of reconciling the intrinsically random nature of evolutionary operations with the requirement that the evolving semantic networks should be meaningful.

This is so because, unlike existing graph-based approaches such as GP or GNP, not every node in a semantic network graph can be connected to an arbitrary other node through an arbitrary type of relation. This issue is relevant in every type of modification operation that needs to be executed during the course of our algorithm.

Simply put, the operations should be constrained by commonsense knowledge: a relation such as $\text{IsA}(\text{bird}, \text{animal})$ is meaningful, while $\text{Causes}(\text{bird}, \text{table})$ is not.

We address this problem by utilizing the nascent subfield of AI named *commonsense reasoning* [18, 31]. Within AI, since the pioneering work by McCarthy [27], commonsense reasoning has been commonly regarded as a key ability that a system must possess in order to be considered truly intelligent [28].

Commonsense reasoning refers to the type of reasoning involved in everyday human thinking, based on *commonsense knowledge* that an ordinary person is expected to know, or “the knowledge of how the world works” [31]. It comprises information such as $\text{HasA}(\text{human}, \text{brain})$, $\text{IsA}(\text{sun}, \text{star})$, or $\text{CapableOf}(\text{ball}, \text{roll})$,

Table 1: Parameter set of the novel evolutionary algorithm (Algorithm 1).

	Parameter	Interval	Explanation
Evolution	$Size_{pop}$	$[1, \infty)$	Number of individuals forming the population
	$Prob_{rec}$	$[0, 1]$	Probability of applying crossover operation
	$Prob_{mut}$	$[0, 1]$	Probability of applying mutation operation
Semantic networks	$Size_{network}$	$[1, \infty)$	Maximum size of randomly created semantic networks in the initial population
	$Score_{min}$	$[-10, 10]$	Minimum quality score of commonsense relations used throughout the algorithm
	$Count_{timeout}$	$[1, \infty)$	Timeout value for the number of trials in commonsense retrieval operations
Tournament selection	$Size_{tourn}$	$[1, Size_{pop}]$	Number of individuals randomly selected from the population for each tournament selection event
	$Prob_{win}$	$[0, 1]$	The probability that the best individual in the ranked list of tournament participants wins the tournament

which are acquired and taken for granted by any adult human, but which need to be introduced in a particular way to a computational reasoning system.

Knowledge bases such as the Cyc project maintained by Cycorp company⁵, ConceptNet project of MIT Media Lab⁶, and the Never-Ending Language Learning (NELL) project of Carnegie Mellon University⁷ are set up to collect and classify commonsense information for the use of research community. In our current implementation, we make use of ConceptNet and the lexical database WordNet⁸ to address the restrictions of processing commonsense knowledge.

3.2.1 Knowledge bases

The ConceptNet project is a part of the Open Mind Common Sense (OMCS) initiative of the MIT Media Lab, based on the input of commonsense knowledge from general public through several ways, including parsed natural language and semi-structured fill-in-the-blanks type forms [18]. As of 2013, ConceptNet is in version 5 and, in addition to data collected through OMCS, it has been extended to include other data sources such as the Wikipedia and Wiktionary projects of the Wikimedia Foundation⁹ and the DBpedia project¹⁰ of the University of Leipzig and the Freie Universität Berlin.

Access to the ConceptNet database is provided through a web API using JavaScript Object Notation (JSON) textual data format. Due to performance reasons, we use the previous version of ConceptNet, version 4, in

our implementation. This is because of the high volume of queries to ConceptNet during the creation of random semantic networks and the application of variation operators. ConceptNet 4 provides the complete dataset in locally accessible and highly efficient SQLite database format, enabling substantially faster access to data compared with the web API and JSON format of the current version.

According to the study by Diochnos [10], ConceptNet version 4 includes 566,094 assertions and 321,993 concepts. The variety of assertions in ConceptNet, initially contributed by volunteers from general public, makes it somewhat prone to noise. According to our experience, noise is generally due to charged statements about political issues, biased views about gender issues, or attempts of making fun. We address the noise problem by ignoring all assertions with a reliability score (determined by contributors' voting) below a set minimum $Score_{min}$ (Table 1)¹¹.

The lexical database WordNet [13] maintained by the Cognitive Science Laboratory at Princeton University also has characteristics of a commonsense knowledge base that make it attractive for our purposes. WordNet is based on a grouping of words into *synsets* or *synonym rings* which hold together all elements that are considered semantically equivalent¹².

In addition to these synset groupings, WordNet includes *pointers* that are used to represent relations between the words in different synsets. These include semantic pointers that represent relations between word

⁵ <http://www.cyc.com/>

⁶ <http://conceptnet5.media.mit.edu/>

⁷ <http://rtw.ml.cmu.edu/rtw/>

⁸ <http://wordnet.princeton.edu>

⁹ <http://www.wikimedia.org/>

¹⁰ <http://dbpedia.org/>

¹¹ The default reliability score for a statement is 1 [18]; and zero or negative reliability scores are a good indication of information that can be considered noise.

¹² Another definition of *synset* is that it is a set of synonyms that are interchangeable without changing the truth value of any propositions in which they are embedded.

Table 2: Set of correspondences we define between WordNet and ConceptNet relation types.

WordNet		ConceptNet	
Relation	Example	Relation	Example
Hypernym	<i>canine</i> is a hypernym of <i>dog</i>	IsA	IsA(<i>dog</i> , <i>canine</i>)
Holonym	<i>automobile</i> is a holonym of <i>wheel</i>	PartOf	PartOf(<i>wheel</i> , <i>automobile</i>)
Meronym	<i>wheel</i> is a meronym of <i>automobile</i>	PartOf	PartOf(<i>wheel</i> , <i>automobile</i>)
Attribute	<i>edible</i> is an attribute of <i>pear</i>	HasProperty	HasProperty(<i>pear</i> , <i>edible</i>)
Entailment	<i>to sleep</i> is entailed by <i>to snore</i>	Causes	Causes(<i>sleep</i> , <i>snore</i>)

meanings and lexical pointers that represent relations between word forms.

For treating WordNet as a commonsense knowledge base compatible with ConceptNet, we define the set of correspondences we outline in Table 2. Similar approaches have also been used by other researchers in the field, such as by Kuo and Hsu [23].

In the implementation of our algorithm, we answer the various types of queries to commonsense knowledge bases (such as the RANDOMCONCEPT() call in Algorithm 3) via ConceptNet or WordNet on a random basis. When the query is answered by information retrieved from WordNet, we return the information formatted in ConceptNet structure based on the correspondences outlined in Table 2 and attach the maximum reliability score of 10, since the information in WordNet is provided by domain experts and virtually devoid of noise.

In our implementation we use WordNet version 3, contributing definitional relations involving around 117,000 synsets. Another thing to note here is that, in ConceptNet version 5, WordNet already constitutes one of the main incorporated data sources. This means that, in case we switch from ConceptNet version 4 to version 5, our approach of accessing WordNet would be obsolete.

3.3 Initialization

At the start of a run, the population of size $Size_{pop}$ is initialized (Algorithm 2) with individuals created through a procedure that we call *random semantic network gen-*

eration (Algorithm 3), capable of assembling random semantic networks of any given size.

Figure 2 presents an example of a random semantic network created via this procedure. This works by starting from a network comprising a sole concept randomly picked from commonsense knowledge bases and running a semantic network expansion algorithm that

1. randomly picks a concept in the given network (e.g. *human*);
2. compiles a list of relations, from commonsense knowledge bases, that the picked concept can be involved in (e.g. *CapableOf(human, think)*, *Desires(human, eat)*, ...);
3. appends to the network a relation randomly picked from this list, together with the other involved concept; and
4. repeats this process until a given number of concepts have been appended to the network, or a set timeout $Count_{timeout}$ has been reached (as a failsafe for situations where there are not enough relations involving the concepts in the network being created).

It is very important to note here that *even if it is grown in a random manner, the generated network itself is totally meaningful, because it is a combination of meaningful pieces of information harvested from commonsense knowledge bases.*

The initialization algorithm depends upon the parameters of $Size_{network}$, the intended number of concepts in the randomly created semantic networks, and $Score_{min}$, the minimum ConceptNet relation score that should be satisfied by the retrieved relations (Table 1).

Algorithm 2 Procedure for the creation of initial random population.

```

1: procedure INITIALIZEPOPULATION( $Size_{pop}$ ,
    $Size_{network}$ ,  $Score_{min}$ ,  $Count_{timeout}$ )
2:   initialize  $P$  ▷ The return array
3:   for  $Size_{pop}$  times do
4:      $r \leftarrow$  RANDOMNETWORK( $Size_{network}$ ,  $Score_{min}$ ,
    $Count_{timeout}$ ) ▷ Generate a new random network
5:     APPENDTO( $P$ ,  $r$ )
6:   end for
7:   return  $P$ 
8: end procedure

```

3.4 Fitness measure

After the initial generation is populated by individuals created by the random semantic network generation algorithm that we outlined, the algorithm proceeds by assigning fitness values to each individual. Since

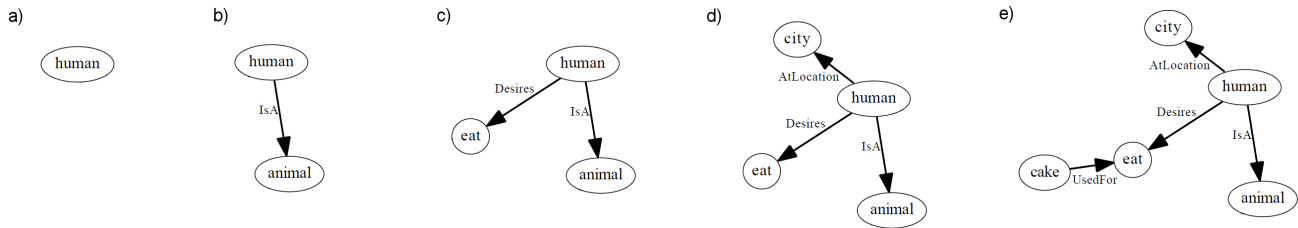


Fig. 2: The process of random semantic network generation, starting with a single random concept in (a) and proceeding with (b), (c), (d), (e), adding new random concepts from the set of concepts related to existing ones.

Algorithm 3 The random semantic network generation algorithm. The algorithm is presented here in a form simpler than the actual implementation, for the sake of clarity.

```

1: procedure RANDOMNETWORK( $Size_{network}$ ,  $Score_{min}$ ,
    $Count_{timeout}$ )
2:   initialize  $net$             $\triangleright$  Empty return network
3:   initialize  $c$             $\triangleright$  Random initial seed concept
4:   for  $Count_{timeout}$  times do
5:      $c \leftarrow$  RANDOMCONCEPT( $Score_{min}$ )
6:      $rels \leftarrow$  INVOLVEDRELATIONS( $c$ )
7:     if SIZE( $rels$ )  $\geq$   $Size_{network}$  then
8:       APPENDTO( $net$ ,  $c$ )
9:     break for            $\triangleright$  Favor a seed with more than a
   few relations
10:    end if
11:  end for
12:   $t \leftarrow 0$ 
13:  repeat
14:     $c \leftarrow$  RANDOMCONCEPTIN( $net$ )
15:     $rels \leftarrow$  INVOLVEDRELATIONS( $c$ )            $\triangleright$  The set of
   relations involving  $c$ 
16:     $r \leftarrow$  RANDOMRELATIONIN( $rels$ )
17:    if SCORE( $r$ )  $\geq$   $Score_{min}$  then
18:      APPENDTO( $net$ ,  $r$ )            $\triangleright$  Append to the network
    $net$  the relation  $r$  and its involved concepts
19:    end if
20:     $t \leftarrow t + 1$ 
21:  until SIZE( $net$ )  $\geq$   $Size_{network}$  or  $t \geq$   $Count_{timeout}$ 
22:  return  $net$ 
23: end procedure

```

our approach constitutes the first instance of semantic network-based EA, it falls on us to introduce fitness measures of interest for its validation.

As an example for showcasing our approach, in Sect. 4, we define a fitness measure based on analogical similarity to an existing semantic network, giving rise to spontaneous generation of semantic networks that are in each generation more and more structurally analogous to a given network.

In general terms, a direct and very interesting application of our approach would be to devise realistically formed fitness functions modeling selectionist theories of knowledge, which remain untested until this time.

One such theory is the *evolutionary epistemology* theory of Campbell [3], which describes the development of human knowledge and creativity through selectionist principles, such as the *blind variation and selective retention* (BVSr) principle.

It is also possible to make the inclusion of certain concepts in the evolving semantic networks a requirement, allowing the discovery of networks formed around a given set of seed concepts. This can be also achieved through starting the initialization procedure (Algorithm 3) with the given seed concepts.

After all the individuals in the current generation are assigned fitness values, the algorithm proceeds with the creation of the next generation of individuals through variation operators (Algorithm 5). But before this, the algorithm has to apply selection to pick individuals from the current population that will be “surviving” to produce offspring.

3.5 Selection

After the assignment of fitness values, individuals are replaced with offspring generated via variation operators applied on selected parents. We employ tournament selection, because it is better at preserving population diversity¹³ and allowing selection pressure to be adjusted through simple parameters [35].

Tournament selection involves, for each selection event, running “tournaments” among a group of $Size_{tourn}$ randomly selected individuals. Individuals in the tournament pool then challenge each other in pairs and the individual with the higher fitness will win with probability $Prob_{win}$. This method simulates biological mating patterns in which two members of the same sex compete to mate with a third one of different sex for the recombination of genetic material. Individuals with

¹³ *Diversity*, in EA, is a measure of homogeneity of the individuals in the population. A drop in diversity indicates an increased number of identical individuals, which is not desirable for the progress of evolution.

higher fitness have better chance of being selected, but an individual with low fitness still has a chance, however small, to produce offspring. Adjusting parameters $Size_{tourney}$ and $Prob_{win}$ (Table 1) gives us an intuitive and straightforward way to adjust the selection pressure on both strong and weak individuals.

In our implementation, we also allow reselection, meaning that the same individual from a particular generation can be selected more than once to produce offspring in different combinations. Algorithm 4 gives an overview of the selection procedure that we implement.

Algorithm 4 Implemented selection algorithm.

```

1: procedure SELECT( $P(t)$ ,  $\phi(t)$ ,  $Size_{tourney}$ ,  $Prob_{win}$ )
2:    $w \leftarrow \text{RANDOMMEMBER}(P(t))$   $\triangleright$  Current winner
3:   for  $Size_{tourney} - 1$  times do
4:      $o \leftarrow \text{RANDOMMEMBER}(P(t))$   $\triangleright$  The next opponent
5:     if LOOKUPFITNESS( $\phi(t)$ ,  $o$ )  $\geq$  LOOKUPFITNESS( $\phi(t)$ ,  $w$ ) then
6:       if RANDOMREAL(0, 1)  $\leq Prob_{win}$  then
7:          $w \leftarrow o$   $\triangleright$  Opponent defeats current winner
8:       end if
9:     end if
10:  end for
11:  return  $w$ 
12: end procedure

```

3.6 Memetic variation operators

Variation operators form the last step in the cycle of our algorithm by creating the next generation of individuals before going back to the step of fitness evaluation (Algorithm 1).

As we mentioned in Sect. 3.2, our representation does not permit arbitrary connections between different nodes in the network and requires special variation operators that should respect the commonsense structure of represented knowledge.

In the following sections, we present the *commonsense crossover* and *commonsense mutation* operators that we set up specific to semantic networks.

Using these operators, the next step in the cycle of our algorithm is the creation of the offspring through variation (Algorithm 5). Crossover is applied to parents selected from the population until $Size_{pop} \times Prob_{rec}$ offspring are created (Table 1), where each crossover event creates two offspring from two parents.

Following the tradition in the GP field [22], we design the variation process such that the offspring created by crossover do not undergo mutation. The mutation operator is applied only to the rest of individuals that are copied, or “reproduced”, directly from the previous generation.

For generating the remaining part of the population, we reproduce $Size_{pop} \times (1 - Prob_{rec}) - 1$ number of individuals selected, and make these subject to mutation. We employ elitism: the last individual (hence the remaining -1 in the previous equation) is a copy of the one with the current best fitness.

Algorithm 5 Procedure for generating the next generation of individuals.

```

1: procedure NEXTGENERATION( $P(t)$ ,  $\phi(t)$ ,  $Size_{pop}$ ,  $Size_{tourney}$ ,  $Prob_{win}$ ,  $Prob_{rec}$ ,  $Prob_{mut}$ ,  $Score_{min}$ ,  $Count_{timeout}$ )
2:   initialize  $N$   $\triangleright$  The return array
3:    $c \leftarrow Size_{pop} Prob_{rec} / 2$   $\triangleright$  Number of crossover events
4:    $r \leftarrow Size_{pop} - 2c$   $\triangleright$  Number of reproduction events
5:   for  $c$  times do
6:      $p1 \leftarrow \text{SELECT}(P(t), \phi(t), Size_{tourney}, Prob_{win})$ 
7:      $p2 \leftarrow \text{SELECT}(P(t), \phi(t), Size_{tourney}, Prob_{win})$ 
8:      $o1, o2 \leftarrow \text{CROSSOVER}(p1, p2)$   $\triangleright$  Crossover the two
        parents
9:     APPENDTO( $N$ ,  $o1$ )  $\triangleright$  Two offspring from each
        crossover
10:    APPENDTO( $N$ ,  $o2$ )
11:  end for
12:  for  $r - 1$  times do
13:     $m \leftarrow \text{SELECT}(P(t), \phi(t), Size_{tourney}, Prob_{win})$ 
14:     $m \leftarrow \text{MUTATE}(m, Prob_{mut})$   $\triangleright$  Mutate an
        individual
15:    APPENDTO( $N$ ,  $m$ )
16:  end for
17:  APPENDTO( $N$ , BESTINDIVIDUAL( $P(t)$ ,  $\phi(t)$ ))  $\triangleright$  Elitism
18:  return  $N$ 
19: end procedure

```

3.6.1 Commonsense crossover

We introduce two types of *commonsense crossover* that are tried in sequence by the variation algorithm.

The first type attempts a sub-graph interchange between two selected parents similar to common crossover in standard GP; and where this is not feasible due to the commonsense structure of relations forming the parents, the second type falls back to a combination of both parents into a new offspring.

3.6.2 Type I Crossover (Subgraph Crossover)

Firstly, a pair of concepts, one from each parent, that are *interchangeable*¹⁴ are selected as *crossover concepts*, picked randomly out of all possible such pairs.

For instance, for the parent networks in Figure 3 (a) and (b), **bird** and **airplane** are interchangeable, since

¹⁴ We define two concepts from different semantic networks as *interchangeable* if both can replace the other in all, or part, of the relations the other is involved in, queried from commonsense knowledge bases.

they can replace each other in the relations `CapableOf(·, fly)` and `AtLocation(·, air)`.

In each parent, a subgraph is formed, containing:

1. the crossover concept;
2. the set of all relations, and associated concepts, that are not common with the other crossover concept. For example, in Figure 3 (a), `HasA(bird, feather)` and `AtLocation(bird, forest)`; and in Figure 3 (b), `HasA(airplane, propeller)`, `MadeOf(airplane, metal)`, and `UsedFor(airplane, travel)`; and
3. the set of all relations and concepts connected to those found in the previous step, excluding the ones that are also one of those common with the other crossover concept.

For example, in Figure 3 (a) including `PartOf(feather, wing)` and `PartOf(tree, forest)`; and in Figure 3 (b), including `MadeOf(propeller, metal)`; but excluding the concept `fly` in Figure 3 (a), because of the relation `CapableOf(·, fly)`.

This, in effect, forms a subgraph of information specific to the crossover concept, which is insertable into the other parent. Any relations between the subgraph and the rest of the network not going through the crossover concept are severed (e.g. `UsedFor(wing, fly)` in Figure 3 (a)).

The two offspring are formed by exchanging these subgraphs between the parent networks (Figure 3 (c) and (d)).

3.6.3 Type II Crossover (Graph Merging Crossover)

Given two parent networks, such as Figure 4 (a) and (b), where no interchangeable concepts between these two can be located, the system falls back to the simpler type II crossover.

A concept from each parent that is *attachable*¹⁵ to the other parent is selected as a *crossover concept*.

The two parents are merged into an offspring by attaching a concept in one parent to another concept in the other parent, picked randomly out of all possible attachments (`CreatedBy(art, human)` in Figure 4 (c). Another possibility is `Desires(human, joy)`). The second offspring is formed randomly in the same way. In the case that no attachable concepts are found, the parents are merged as two separate clusters within the same individual.

¹⁵ We define a distinct concept as *attachable* to a semantic network if at least one commonsense relation connecting the concept to any of the concepts in the network can be discovered from commonsense knowledge bases.

3.6.4 Commonsense mutation

We introduce several types of *commonsense mutation* operators that modify a parent by means of information from commonsense knowledge bases.

For each mutation to be performed, the type is picked at random with uniform probability. If the selected type of mutation is not feasible due to the commonsense structure of the parent, another type is again picked. In the case that a set timeout of $Count_{timeout}$ trials has been reached without any operation, the parent is returned as it is.

3.6.5 Type I (Concept Attachment)

A new concept randomly picked from the set of concepts *attachable* to the parent is attached through a new relation to one of existing concepts (Figure 5 (a) and (b)).

3.6.6 Type IIa (Relation Addition)

A new relation connecting two existing concepts in the parent is added, possibly connecting unconnected clusters within the same network (Figure 5 (c) and (d)).

3.6.7 Type IIb (Relation Deletion)

A randomly picked relation in the parent is deleted, possibly leaving unconnected clusters within the same network (Figure 5 (e) and (f)).

3.6.8 Type IIIa (Concept Addition)

A randomly picked new concept is added to the parent as a new cluster (Figure 5 (g) and (h)).

3.6.9 Type IIIb (Concept Deletion)

A randomly picked concept is deleted with all the relations it is involved in, possibly leaving unconnected clusters within the same network (Figure 5 (i) and (j)).

3.6.10 Type IV (Concept Replacement)

A concept in the parent, randomly picked from the set of those with at least one *interchangeable* concept, is replaced with one of its interchangeable concepts, again randomly picked. Any relations left unsatisfied by the new concept are deleted (Figure 5 (k) and (l)).

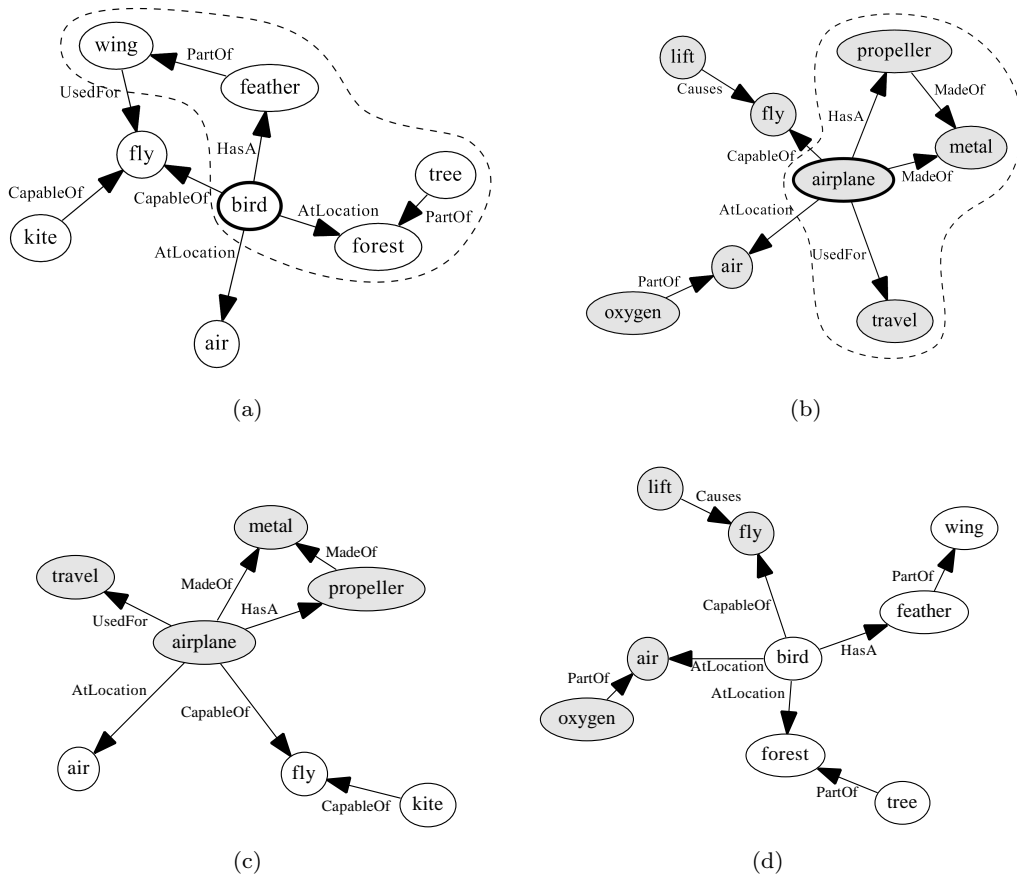


Fig. 3: Commonsense crossover type I (subgraph crossover). (a) Parent 1, centered on the concept *bird*; (b) Parent 2, centered on the concept *airplane*; (c) Offspring 1; (d) Offspring 2.

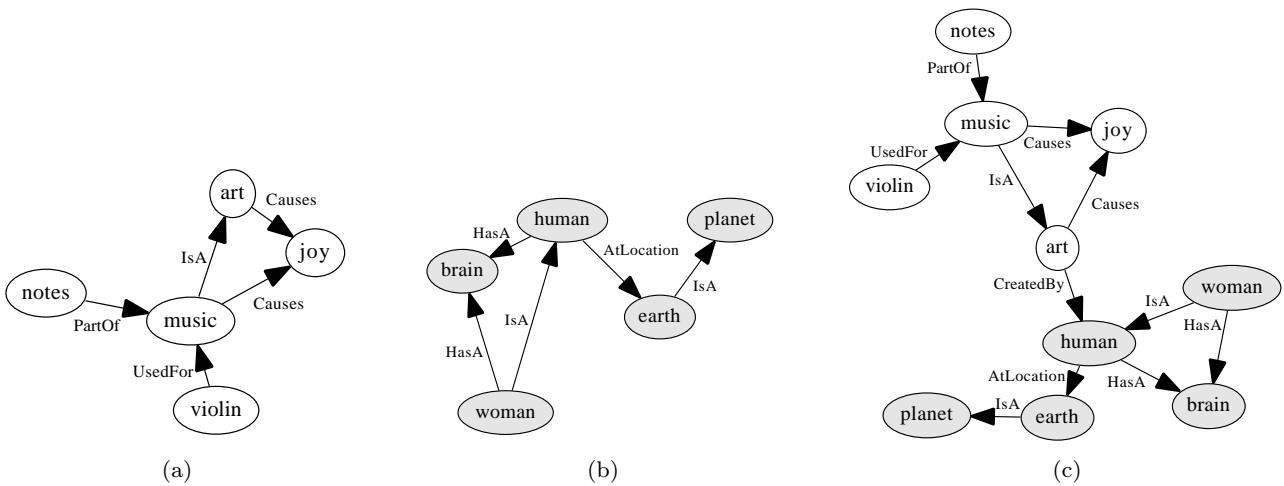


Fig. 4: Commonsense crossover type II (graph merging crossover). (a) Parent 1; (b) Parent 2; (c) Offspring, merging by the relation *CreatedBy*(*art*, *human*). If no concepts attachable through commonsense relations are encountered, the offspring is formed by merging the parent networks as two separate clusters within the same semantic network.

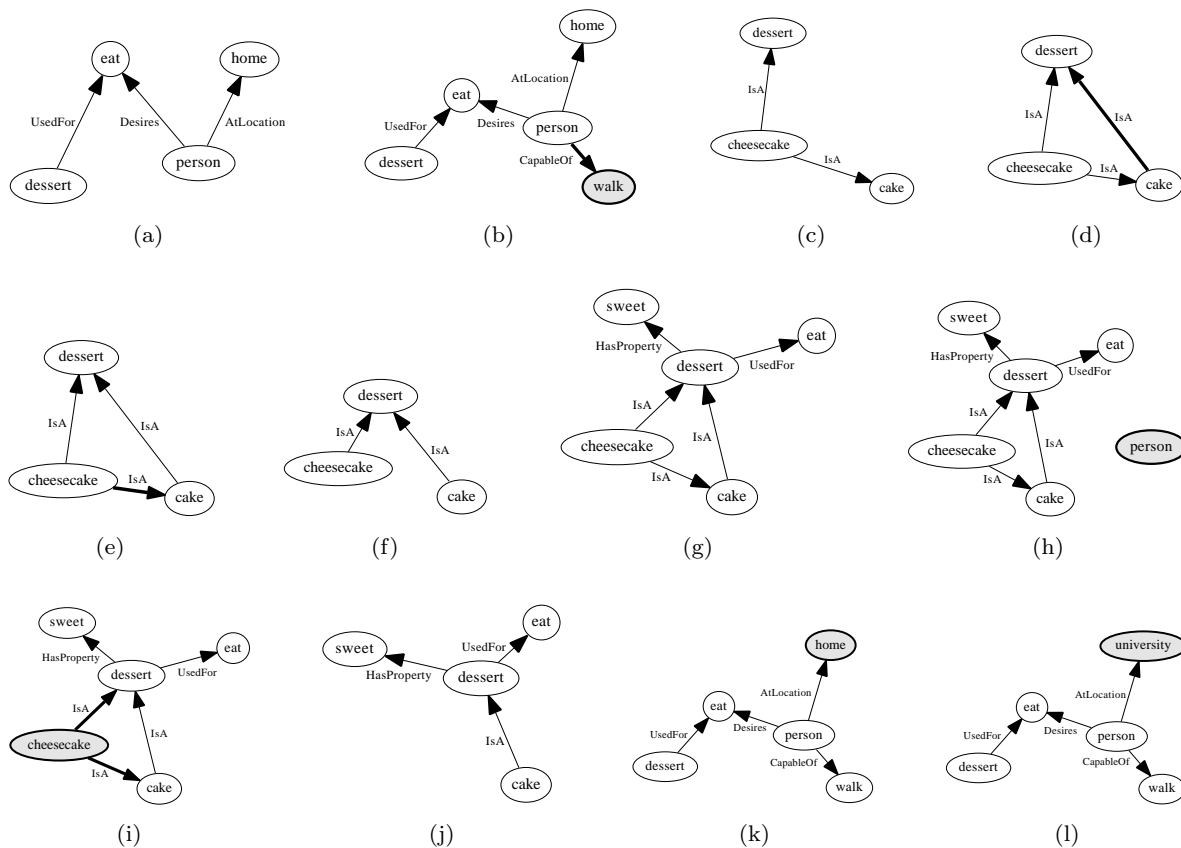


Fig. 5: Examples illustrating commonsense mutation. (a) Mutation type I (before); (b) Mutation type I (after); (c) Mutation type IIa (before); (d) Mutation type IIa (after); (e) Mutation type IIb (before); (f) Mutation type IIb (after); (g) Mutation type IIIa (before); (h) Mutation type IIIa (after); (i) Mutation type IIIb (before); (j) Mutation type IIIb (after); (k) Mutation type IV (before); (l) Mutation type IV (after).

4 Analogy as a fitness measure

For experimenting with our approach, we select analogical reasoning as an initial application area, by using analogical similarity as our fitness measure.

This constitutes an interesting choice for evaluating our work, because it not only validates the viability of the novel algorithm, but also produces results of interest for the fields of analogical reasoning and computational creativity.

4.1 Analogies and creativity

There is evidence that analogical reasoning is at the core of higher-order cognition, and it enters into creative discovery, problem-solving, categorization, and learning [20]. Analogy-making ability is extensively linked with creative thought and regularly plays a role in creativity expressed in arts and sciences. Boden [4] classifies analogy as a form of *combinational creativity*, noting

that it works by producing unfamiliar combinations of familiar ideas.

In addition to literary use of metaphors and allegories in written language, analogies often constitute the basis of composition in all art forms including visual or musical. For example, in classical music, it is highly common to formulate interpretations of a composer's work in terms of tonal allegories [5]. In visual arts, examples of artistic analogy abound, ranging from allegorical compositions of Renaissance masters such as Albrecht Dürer, to modern usage in film, such as the many layers of allegory in Stanley Kubrick's 2001: A Space Odyssey [34].

In science, analogies have been used to convey revolutionary theories and models. A key example of analogy-based explanations is Kepler's explanation of the laws of heliocentric planetary motion with an analogy to light radiating from the Sun¹⁶.

¹⁶ Kepler argued, in his *Astronomia Nova*, as light can travel undetectably on its way between the source and destination,

Another instance is Rutherford’s analogy between the atom and the Solar System, where the internal structure of the atom is explained by electrons circling the nucleus in orbits like planets around the Sun. This model, which was later improved by Bohr to give rise to the Rutherford-Bohr model, was one of the “planetary models” of the atom, where the electromagnetic force between oppositely charged particles were presented analogous to the gravitational force between planetary bodies. Earlier models of the atom were, also notably, explained using analogies, including “plum pudding” model of Thomson and the “billiard ball” model of Dalton (Figure 6).

In contemporary studies, analogical reasoning is mostly seen through a structural point of view, framed by the structure mapping theory based on psychology [16].

Other approaches to analogical reasoning include the view of Hofstadter [19] of analogy as a kind of high-level perception, where one situation is perceived as another one. Veale and Keane [41] extend the work in analogical reasoning to the more specific case of metaphors, which describe the understanding of one kind of thing in terms of another. A highly related cognitive theory is the *conceptual blending* idea developed by Fauconnier and Mark [12], which involves connecting several existing concepts to create new meaning, operating below the level of consciousness as a fundamental mechanism of cognition. An implementation of this idea is given by Pereira [33] as a computational model of abstract thought, creativity, and language.

Computational approaches within the analogical reasoning field have been mostly concerned with the *mapping* problem [14]. Put in a different way, models developed and implemented are focused on constructing mappings between two given source and target domains (Figure 7 (a)). This focus neglects the problem of retrieval or recognition of a new source domain, given a target domain, or the other way round.

By combining our algorithm for the evolution of semantic networks with a fitness measure based on analogical similarity, we can essentially produce a method to address this creativity-related subproblem of analogical reasoning, which has remained, so far, virtually untouched.

We accomplish this by

1. providing our evolutionary algorithm with a “reference” semantic network that will represent the input to the system; and

2. running the evolutionary process under a fitness measure quantifying analogical similarity to the given “reference” network.

This, in effect, creates a “survival of the fittest analogies” process where, starting from a random initial population of semantic networks, one gets semantic networks that get gradually more analogous to the given reference network.

In our implementation, we define the fitness measure to take the reference semantic network as the *base* and the individual whose fitness is just being evaluated as the *target*. In other terms, this means that the system produces structurally analogous target networks for a given base network. From a computational creativity perspective, an interpretation for this would be the “imagining”, or creation, of a novel case that is analogous to a case at hand.

This designation of the base and target roles for the two networks is an arbitrary choice, and it is straightforward to define the fitness function in the other direction. So, if the system would be set up such that it would produce base networks, given the target network, one can then interpret this as the the classical retrieval process in analogical reasoning, where one is supposed to retrieve a base case that is analogous to the currently encountered case, for using it as a basis for solution.

If one subscribes to the “retrieval of a base case” interpretation, since the ultimate source of all the information underlying the generated networks is the commonsense knowledge bases, one can treat this source of knowledge as a part of the system’s memory, and see it as a “generic case base” from which the base cases are retrieved.

On the other hand, if we consider the “imagination of a novel case” interpretation, our system, in fact, replicates a mode of behavior observed in psychology research where an analogy is not always simply *recognized* between an original case and a retrieved analogous case from memory, but the analogous case can sometimes be *created* together with the analogy [7].

Considering the depth of commonsense knowledge sources, this creation process is virtually open-ended; and due to the random nature of the algorithm, it produces different analogous cases in each run. This capability of open-ended creation of novel analogous cases is, to our knowledge, the first of its kind and makes our approach interesting for the analogical reasoning and computational creativity fields.

and yet illuminate the destination, so can motive force be undetectable on its way from the Sun to planet, yet affect planet’s motion.

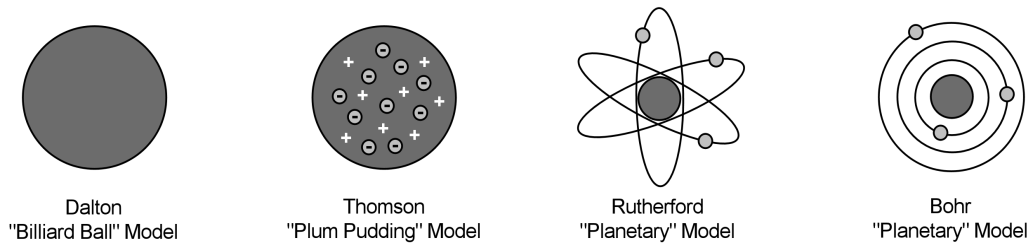


Fig. 6: The Dalton (1805), Thomson (1904), Rutherford (1911), and Bohr (1913) models of the atom with their corresponding analogies.

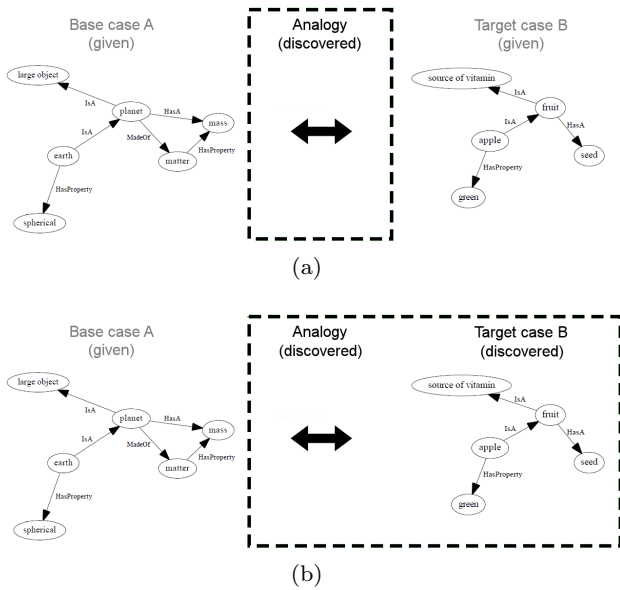


Fig. 7: Contribution to computational analogy-making. (a) Existing work in the field, restricted to finding analogical mappings between a given pair of domains (b) Our novel approach, capable of creating novel analogies as well as the analogous case itself.

4.2 Structure mapping engine

The Structure Mapping Engine (SME) [11] is an analogical matching algorithm firmly based on the psychological *structure mapping theory* of Gentner. It is a very robust algorithm, having been used in many practical applications by a variety of research groups, and it has been considered the most influential work on the modeling of analogy-making [14].

An important characteristic of SME is that it ignores surface features and it can uncover mappings between potentially very distant domains, if they have a similar representational structure.

A typical example given for illustrating the working of SME is the analogy between the Rutherford-Bohr atom model and the Solar System, which we already

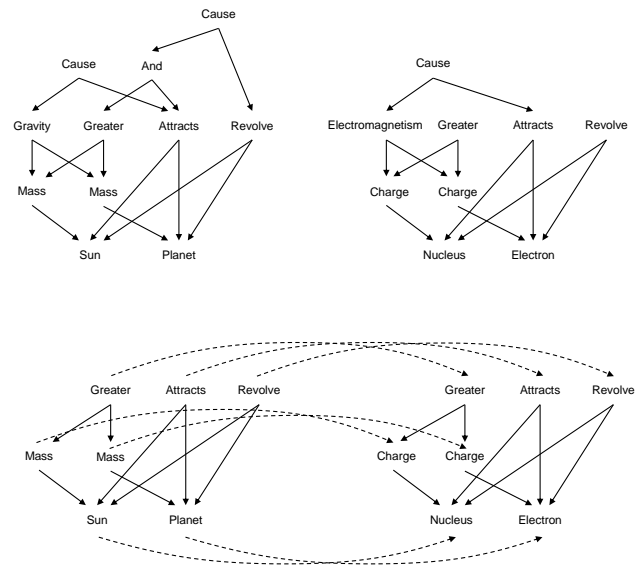


Fig. 8: Representation of the Rutherford-Bohr atom model – Solar System analogy as graphs: predicates about the two domains (top), analogy as a mapping of structure between the two domains (bottom).

mentioned. Using a predicate calculus representation, Figure 8 illustrates a structural mapping between these domains.

Here we make use of our own implementation of SME based on the original description by Falkenhainer et al. [11] and adapt it to the simple structure of semantic networks.

Using SME in this way necessitates the introduction of a mapping between the concept–relation based structure of semantic networks and the predicate calculus based representation traditionally used in SME applications.

A highly versatile such mapping is given by Larkey and Love [24]. Given information such as “Jim (a man) loves Betty (a woman)”, one can transform the predicate calculus representation of `loves(Jim, Betty)`, `gender(Jim, male)`, `gender(Betty, female)` into a

Table 3: Correspondences between SME predicate calculus statements [11] and semantic network structure that we define for applying structure mapping to semantic networks.

Predicate calculus	Semantic networks
Entity	Concept (node)
Relation	Relation (edge)
Attribute	IsA or HasProperty relation
Function	Not employed

semantic network representation by converting predicates into nodes such as **gender** and **loves**; and creating argument nodes for each argument of a predicate. This kind of mapping makes it possible, theoretically, to represent arbitrarily complex information within the simple representation framework of semantic networks. As an example, one can represent meta-information such as “John knows that Jim loves Betty”.

However, the approach of Larkey and Love [24] requires the creation of ad hoc “relation nodes” for the representation of relations between concepts and the usage of unlabeled directed edges. On the other hand, the existing structure of the commonsense knowledge bases that we interface extensively, mainly ConceptNet, are based on nodes representing concepts and labeled directed edges representing relations. In this representation, nodes can have arbitrary names but the names of edges come from a limited set of basic relation names¹⁷. Because of this, we take another approach for mapping between semantic networks and predicate calculus.

Due to these reasons, we define a basic list of correspondences between the two representation schemes, where we treat “entities” as concepts, relations as relations, attributes as IsA relations; and exclude functions. Table 3 gives the list of correspondences that we employ in our SME implementation.

4.3 Results

With the implementation of analogical similarity-based fitness measure that we described so far, we carried out numerous experiments with reference networks representing different domains. In this part, we present the results from two such experiments.

¹⁷ For ConceptNet version 4: IsA, HasA, PartOf, UsedFor, AtLocation, CapableOf, MadeOf, CreatedBy, HasSubevent, HasFirstSubevent, HasLastSubevent, HasPrerequisite, MotivatedByGoal, Causes, Desires, CausesDesire, HasProperty, ReceivesAction, DefinedAs, SymbolOf, LocatedNear, ObstructedBy, ConceptuallyRelatedTo, InheritsFrom.

Table 4: Parameter set used during experiments. Refer to Table 1 for an explanation of parameters.

	Parameter	Value
Evolution	$Size_{pop}$	200
	$Prob_{rec}$	0.85
	$Prob_{mut}$	0.15
Semantic networks	$Size_{network}$	5
	$Score_{min}$	2
	$Count_{timeout}$	10
Tournament selection	$Size_{tourn}$	8
	$Prob_{win}$	0.8

Table 4 provides an overview of the parameter values that we used for conducting these experiments.

The selection of crossover and mutation probabilities for a particular application have been a traditional subject of debate in EA literature [39]. Since the foundation of the field, in essence, the arguments have been mainly centered on the relative importance of the crossover and mutation operators in the progress of evolution. For our approach, we make the decision to follow the somewhat established consensus in the graph-based EA field [32], dominated by genetic programming (GP) and the selection of parameters by the pioneering work of Koza.

Thus, we use a crossover probability of $Prob_{rec} = 0.85$, similar to the high crossover probabilities typically ≥ 0.9 encountered in GP literature [22].

However, unlike the typical GP mutation value of ≤ 0.1 , we employ a somewhat-above-average mutation rate of $Prob_{mut} = 0.15$.

Due to the fact that our algorithm is the first attempt at having a graph-based evolutionary model of memetics, this mutation rate is somewhat arbitrary and is dependent on our subjective interpretation of the mutation events in memetic processes. Nonetheless, there is preliminary support for a high mutation rate in memetics, where it has been postulated, for example by Gil-White [17], that memes would have a high tendency of mutation.

We select a population size of $Size_{pop} = 200$ individuals, and subject this population to tournament selection with a tournament size of $Size_{tourn} = 8$ and a winning probability $Prob_{win} = 0.8$.

Using this parameter set, here we present the results from two runs of experiment:

1. analogies generated for a network describing some basic astronomical knowledge, shown in Figure 9; and
2. analogies generated for a network describing familial relations, shown in Figure 11.

For the first reference base network (Figure 9), after a run of the algorithm for 35 generations, the system produced the target network shown in Figure 10.

The produced target network exhibits an almost one-to-one structural correspondence with the reference network, missing only one node (`mass` in the original network) and two relations both pertaining to this missing node (`HasA(planet, mass)` and `HasProperty(matter, mass)`). The discovered analogy is remarkably inventive, and draws a parallel between the Earth and an apple: Just as the Earth is like an apple, planets are like fruits and the solar system is like a tree holding these fruits. Just as the solar system is a part of the universe, a tree is a part of a forest.

It is an intuitive analogy and leaves us with the impression that it is comparable with the classic analogy between the atom and the Solar System that we mentioned in the beginning of this section. Table 5 gives a full list of all the correspondences.

For the second reference network (Figure 11), in a run after 42 generations, our algorithm produced the network shown in Figure 12.

The produced analogy can be again considered “creative”, drawing a parallel between human beings and musical instruments. It considers a mother as a clarinet and a father as a drum; and just as a mother is a woman and a father a man, a clarinet is an instance of wind instrument and a drum is an instance of percussion instrument. The rest of the correspondences also follow in a somewhat intuitive way. Again, Table 6 gives a list of correspondences.

During our experiments, we observed that under the selected parameter set, the evolutionary process approaches equilibrium conditions after approximately 50 generations. This behavior is typical and expected in EA approaches and manifests itself with an initial exponential or logarithmic growth in fitness that asymptotically approaches a fitness plateau, after which fitness increasing events will be sporadic and negligible.

Figure 13 shows the progression of the average fitness of the population and the fitness of the best individual for each passing generation, during the course of one of our experiments with the reference network in Figure 9, which lasted for 50 generations. We observe that the evolution process asymptotically reaches a fitness plateau after about 40 generations.

Coinciding with the progression of fitness values, we observe, in Figure 14, the sizes of individual semantic networks both for the best individual and as a population average. Just as in the fitness values, there is a pronounced stabilization of the network size for the best individual in the population, occurring around the 40th generation. While the value stabilizes for the best

Table 5: Experiment 1: Correspondences between the base and target networks, after 35 generations.

Base	Target
Concepts	
<code>earth</code>	<code>apple</code>
<code>moon</code>	<code>leave</code>
<code>planet</code>	<code>fruit</code>
<code>solar system</code>	<code>tree</code>
<code>galaxy</code>	<code>forest</code>
<code>universe</code>	<code>forest</code>
<code>spherical</code>	<code>green</code>
<code>matter</code>	(N/A)
<code>mass</code>	<code>seed</code>
<code>large object</code>	<code>source of vitamin</code>
Relations	
<code>HasA(earth, moon)</code>	<code>HasA(apple, leave)</code>
<code>HasProperty(earth, spherical)</code>	<code>HasProperty(apple, green)</code>
<code>HasProperty(moon, spherical)</code>	<code>HasProperty(leave, green)</code>
<code>IsA(earth, planet)</code>	<code>IsA(apple, fruit)</code>
<code>IsA(planet, large object)</code>	<code>IsA(fruit, source of vitamin)</code>
<code>AtLocation(planet, solar system)</code>	<code>AtLocation(fruit, tree)</code>
<code>AtLocation(solar system, galaxy)</code>	<code>AtLocation(tree, mountain)</code>
<code>PartOf(solar system, universe)</code>	<code>PartOf(tree, forest)</code>
<code>MadeOf(planet, matter)</code>	(N/A)
<code>HasA(planet, mass)</code>	<code>HasA(fruit, seed)</code>
<code>HasProperty(matter, mass)</code>	(N/A)

individual, the population average for the network size keeps a trend of (gradually slowing) increase.

Our interpretation of this phenomenon is that, once the size of the best network becomes comparable with the size of the given reference network (Figure 9, comprising 10 concepts and 11 relations) and the analogies considered by the SME algorithm have already reached a certain quality, further increases in the network size would not cause substantial improvement on the SME structural evaluation score. This is because the analogical mapping from the reference semantic network to the current best individual is already highly optimized and very close to the ideal case of a structurally one-to-one mapping (cf. Figure 9, 10 concepts, 11 relations, and Figure 10, 9 concepts, 9 relations).

In general, our experiments demonstrate that, combined with the SME-based fitness measure, the algorithm we developed is capable of spontaneously creating collections of semantic networks analogous to the one given as reference. In most cases, our implementation was able to reach extensive analogies within 50 generations and reasonable computational resources, where

Table 6: Experiment 2: Correspondences between the base and target networks, after 42 generations.

Base	Target
Concepts	
mother	clarinet
father	drum
woman	wind instrument
man	percussion instrument
human	instrument
home	music hall
care	perform glissando
family	(N/A)
sleep	make music
dream	play instrument
female	member of orchestra
Relations	
IsA(mother, woman)	IsA(clarinet, wind instrument)
IsA(father, man)	IsA(drum, percussion instrument)
IsA(woman, human)	IsA(wind instrument, instrument)
AtLocation(human, home)	AtLocation(instrument, music hall)
IsA(man, human)	IsA(percussion instrument, instrument)
PartOf(mother, family)	(N/A)
PartOf(father, family)	(N/A)
CapableOf(mother, care)	CapableOf(clarinet, perform glissando)
CapableOf(human, sleep)	CapableOf(instrument, make music)
HasSubevent(sleep, dream)	HasSubevent(make music, play instrument)
IsA(woman, female)	IsA(wind instrument, member of orchestra)

a typical run of experiment took around 45 minutes on a medium-range laptop computer with AMD Athlon II 2.2 GHz processor and 8 GB of RAM.

5 Conclusions

We presented a novel graph-based EA employing semantic networks as evolving individuals. The use of semantic networks provides a simple yet powerful means of representing pieces of evolving knowledge, giving us a possibility to interpret this algorithm as an implementation of the idea of memetics. Because this work constitutes the first instance of semantic network-based EA, we had to establish the necessary crossover and mutation operators working on this representation.

We make extensive use of commonsense reasoning and commonsense knowledge bases, necessitated by the semantic network-based representation and the requirement that all operations should ensure meaningful conceptual relations. Put another way, we use a combi-

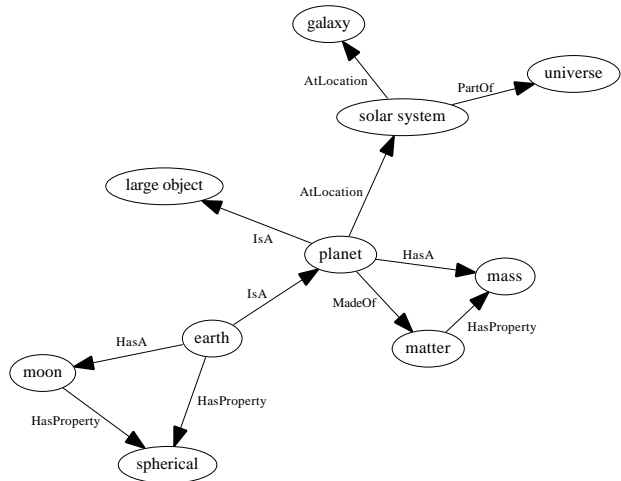


Fig. 9: Experiment 1: Given semantic network, 10 concepts, 11 relations (base domain).

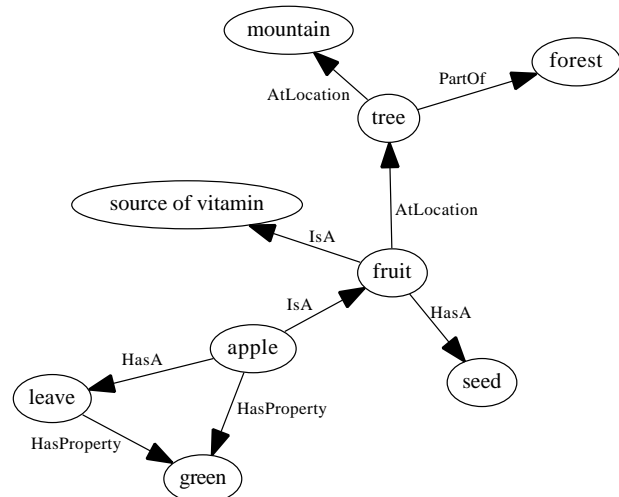


Fig. 10: Experiment 1: Evolved individual, 9 concepts, 9 relations (target domain). The evolved individual is encountered after 35 generations, with fitness value 2.8. Concepts and relations of the individual not involved in the analogy are not shown here for clarity.

nation of random processes constrained by the non-random structural bounds of commonsense knowledge, under selection pressure of the defined fitness function.

For evaluating the approach, we make use of SME as the basis of a fitness function that measures analogical similarity. With the analogical similarity-based fitness calculated between the reference network and the evolving networks in the population, we create a system capable of spontaneously generating networks analogous to any given network. This system represents a first in the analogical reasoning field, because current models

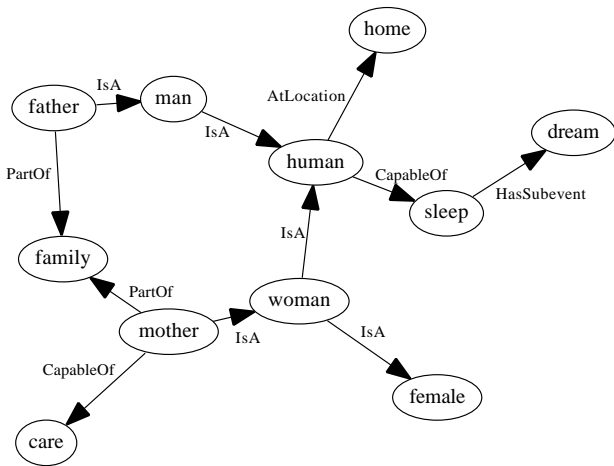


Fig. 11: Experiment 2: Given semantic network, 11 concepts, 11 relations (base domain).

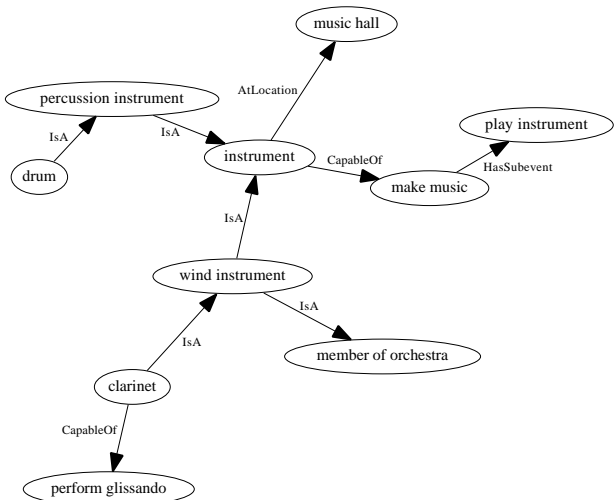


Fig. 12: Experiment 1: Evolved individual, 10 concepts, 9 relations (target domain). The evolved individual is encountered after 42 generations, with fitness value 2.7. Concepts and relations of the individual not involved in the analogy are not shown here for clarity.

have been limited to only finding analogical mappings between two already existing networks.

5.1 Limitations and future work

The most considerable limitation of this work comes from our choice of using semantic networks instead of a more powerful representation scheme. For example, since we are using SME for experimenting with our approach, it would be highly desirable and logical to use predicate calculus to represent evolving individuals. In-

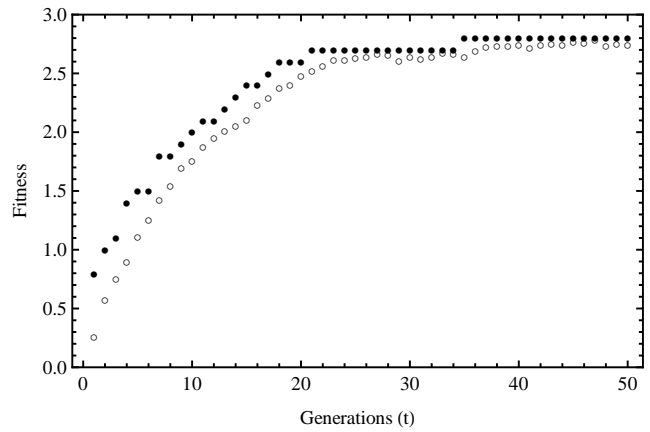


Fig. 13: Progress of fitness during a typical run with parameters given in Table 4. Filled circles represent the best individual in a generation, while the empty circles represent population average.

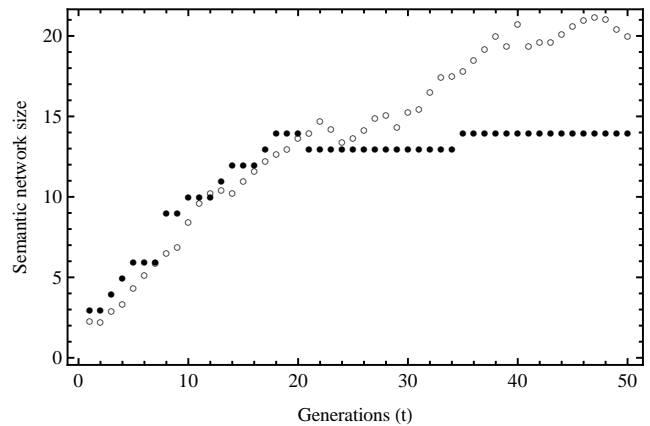


Fig. 14: Progress of semantic network size during a typical run with parameters given in Table 4. Filled circles represent the best individual in a generation, while the empty circles represent population average. Network size is taken to be the number of relations (edges) in the semantic network.

stead, we limit the representation to semantic networks, and provide our own implementation of SME that we adapt to work on the simple directed graph structure of semantic networks.

This choice of limiting representation was mainly directed by our reliance on ConceptNet version 4 as the main commonsense knowledge base used in this study, which is based on simple binary relations using a limited set of relation types. This impedes the representation of more complex information such as temporal relations or causal connections between subgraphs. It should be noted, however, that in the next version, ConceptNet project has made a decision to move to

a “hypergraph” representation, where one can have relations about other instances of relation between concepts. This can, in effect, greatly increase the expressivity of the system.

Another issue in the current study is the selection of parameter values for our EA implementation. Due to the fact that our algorithm is a first attempt at having a graph-based implementation of memetics, we are faced with selecting mutation and crossover rates without any antecedents. Even in theoretical studies of cultural evolution, discussions of the frequency of variation events are virtually nonexistent. This makes our parameter values rather arbitrary, roughly guided by the general conventions in the graph-based EA field.

For future work, it would be interesting to experiment with extensions of the simple SME-based fitness measure that we have used. As semantic networks are graphs, a straightforward possibility is to take graph-theoretical properties of candidate networks into account, such as the clustering coefficient or shortest path length. With these kinds of constraints, selection pressure on the network structure can be adjusted in a more controlled way.

Another highly interesting prospect with the EA system would be to consider different types of mutation and crossover operators, and doing the necessary study for grounding the design of such operators on existing theories of cultural transmission and variation. Combined with realistically formed fitness functions, one can use such a system for modeling selectionist theories of knowledge. Performing experiments with such a setup could be considered a “memetic simulation” and comparable to computational simulations of genetic processes performed in computational biology.

Besides the “memetic” interpretation, a more hands-on application that we foresee we can achieve in the short-term is practical computational creativity. Already with the SME-based fitness measure that we demonstrated in this article, it would be possible to create systems for tasks such as story generation based on analogies [42]. This would involve giving the system an existing story as the input, and getting an analogous story in another domain as the output. For doing this we would need to define a structural representation scheme of story elements, and, preferably an automated way of translating between structural and textual representations.

Acknowledgements This work was supported by a JAE-Predoc fellowship from CSIC, and the research grants: 2009-SGR-1434 from the Generalitat de Catalunya, CSD2007-0022 from MICINN, and Next-CBR TIN2009-13692-C03-01 from MICINN.

References

1. Balkin, J.M.: *Cultural Software: A Theory of Ideology*. Yale University Press (1998)
2. Bedau, M.A., Snyder, E., Brown, C.T.: A comparison of evolutionary activity in artificial evolving systems and in the biosphere. In: Husbands, P., Harvey, I. (eds.) *Proceedings of the Fourth European Conference on Artificial Life*, pp. 125–134. MIT Press, Cambridge (1997)
3. Bickhard, M.H., Campbell, D.T.: Variations in variation and selection: the ubiquity of the variation-and-selective-retention ratchet in emergent organizational complexity. *Foundations of Science* **8**, 215–2182 (2003)
4. Boden, M.A.: Computer models of creativity. *AI Magazine* **30**(3), 23–34 (2009)
5. Chafe, E.: *Tonal Allegory in the Vocal Music of J. S. Bach*. University of California Press (1991)
6. Chen, D., Aoki, T., Homma, N., Terasaki, T., Higuchi, T.: Graph-based evolutionary design of arithmetic circuits. *IEEE Transactions on Evolutionary Computation* **6**(1), 86–100 (2002)
7. Clement, J.: Observed methods for generating analogies in scientific problem solving. *Cognitive Science* **12**, 563–586 (1988)
8. Dawkins, R.: *The Selfish Gene*. Oxford University Press, New York City (1976)
9. Dennett, D.C.: *Darwin’s Dangerous Idea: Evolution and the Meanings of Life*. Simon & Schuster (1995)
10. Diochnos, D.I.: Commonsense reasoning and large network analysis: A computational study of conceptnet 4 (2013). URL <http://arxiv.org/abs/1304.5863>
11. Falkenhainer, B., Forbus, K.D., Gentner, D.: The Structure-Mapping Engine: Algorithm and examples. *Artificial Intelligence* **41**, 1–63 (1989)
12. Fauconnier, G., Mark, T.: *The Way We Think: Conceptual Blending and the Mind’s Hidden Complexities*. Basic Books, New York (2002)
13. Fellbaum, C.: *WordNet: An Electronic Lexical Database*. MIT Press (1998)
14. French, R.M.: The computational modeling of analogy-making. *Trends in Cognitive Sciences* **6**(5), 200–205 (2002)
15. Gabora, L., Kaufman, S.B.: Evolutionary approaches to creativity, pp. 279–300. Cambridge University Press (2010)
16. Gentner, D.: Structure-mapping: A theoretical framework for analogy. *Cognitive Science* **7**, 155–170 (1983)
17. Gil-White, F.: Let the meme be (a meme): insisting too much on the genetic analogy will turn it into

- a straightjacket. In: Botz-Bornstein, T. (ed.) *Culture, Nature, Memes*. Cambridge Scholars, Newcastle upon Tyne (2008)
18. Havasi, C., Speer, R., Alonso, J.: ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In: *Proceedings of Recent Advances in Natural Language Processing* (2007)
 19. Hofstadter, D.R.: *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*. Basic Books, New York (1995)
 20. Hofstadter, D.R.: Analogy as the core of cognition. In: Gentner, D., Holyoak, K.J., Kokinov, B. (eds.) *Analogical Mind: Perspectives from Cognitive Science*, pp. 499–538. MIT Press, Cambridge, MA (2001)
 21. Hofstadter, D.R., Dennett, D.C.: *The Mind's I: Fantasies and Reflections on Self and Soul*. Basic, New York (1981)
 22. Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G.: *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers (2003)
 23. Kuo, Y.L., Hsu, J.: Bridging common sense knowledge bases with analogy by graph similarity. In: *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence* (2010)
 24. Larkey, L.B., Love, B.C.: CAB: Connectionist analogy builder. *Cognitive Science* **27**(2003), 781–794 (2003)
 25. Mabu, S., Hirasawa, K., Hu, J.: A graph-based evolutionary algorithm: Genetic network programming (GNP) and its extension using reinforcement learning. *Evolutionary Computation* **15**(3), 369–398 (2007)
 26. Martindale, C., Locher, P., Petrov, V.M.: *Evolutionary and Neurocognitive Approaches to Aesthetics, Creativity and the Arts*. Foundations and Frontiers of Aesthetics. Baywood Publishing Company (2007)
 27. McCarthy, J.: Programs with common sense. In: *Symposium on Mechanization of Thought Processes*. National Physical Laboratory, Teddington, England (1958)
 28. Minsky, M.: *Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon & Schuster, New York, NY (2006)
 29. Montes, H.A., Wyatt, J.L.: Graph representation for program evolution: An overview. Tech. rep., University of Birmingham School of Computer Science (2004)
 30. Moscato, P.A., Cotta, C., Mendes, A.: *Studies in Fuzziness and Soft Computing – New Optimization Techniques in Engineering*, chap. Memetic Algorithms. Springer, New York (2004)
 31. Mueller, E.T.: *Commonsense Reasoning*. Morgan Kaufmann (2006)
 32. Pereira, F.B., Machado, P., Costa, E., Cardoso, A.: Graph based crossover — A case study with the busy beaver problem. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 2, pp. 1149–1155. Morgan Kaufmann (1999)
 33. Pereira, F.C.: *Creativity and Artificial Intelligence: A Conceptual Blending Approach*. Mouton de Gruyter, New York (2007)
 34. Pezzotta, E.: The metaphor of dance in Stanley Kubrick's 2001: A Space Odyssey, A Clockwork Orange and Full Metal Jacket. *Journal of Adaptation in Film & Performance* **5**(1) (2012)
 35. Pohlheim, H.: Genetic and evolutionary algorithm toolbox documentation (2006). URL <http://www.geatbx.com/docu/algindex-02.html>
 36. Poli, R.: *New Ideas in Optimisation*, chap. Parallel Distributed Genetic Programming. McGraw-Hill (1999)
 37. Skusa, A., Bedau, M.A.: Towards a comparison of evolutionary creativity in biological and cultural evolution. In: Standish, R., Abbass, H.A., Bedau, M.A. (eds.) *Artificial Life VIII*, pp. 233–242. MIT Press (2002)
 38. Sowa, J.F.: *Knowledge Representation: Logical Philosophical, and Computational Foundations*. Brooks/Cole, Pacific Grove, CA (2000)
 39. Spears, W.M.: *Foundations of Genetic Algorithms 2*, chap. Crossover or mutation?, pp. 221–237. Morgan Kaufmann (1992)
 40. Teller, A.: *Algorithm evolution with internal reinforcement for signal understanding*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA (1998)
 41. Veale, T., Keane, M.: The competence of sub-optimal structure mapping on hard analogies. In: *Proceedings of the 15th International Joint Conference on AI*. Morgan Kaufman, San Mateo, CA (1997)
 42. Zhu, J., Ontañón, S.: Evaluating analogy-based story generation: An empirical study. In: *Proceedings of the 9th Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2013)*. Boston, MA (2013)