

Nine Ways to Find a Solution

Zeeshan-ul-hassan Usmani
Florida Institute of Technology
Department of Computer Sciences
Melbourne, Florida, USA

Abstract - *The complete history of problem solving in computer sciences is a history of continuous attempts to discover, study, mimic and implement ideas, models, metaphors, and sequences from nature. This paper explores nine different ways of finding a solution and gives examples from computer science. These nine ways are mainly inspired from nature.*

1 Nine Ways

There are infinite ways to make analogies and find solutions from nature. Nature has always more than one ways to do the same thing. Nature, sometimes, demonstrates all possible ways for a particular task. It is our role to choose the appropriate way and the best solution for a given task and there are nine ways to do that.

1.1 Modeling

A model is an abstract definition of real world phenomena. It is a theoretical construction with a set of variables to demonstrate logical and quantitative functions and relationships among them under a set of given (or assumed) conditions and environments [4]. By modeling, we minimize the real world into our hands; sometimes we divide the whole phenomena in small parts and make models for each of them, so we can test it within limited time and resources. A model is always less than the real thing and it does not have to look like the real thing, but it should behave like the real one. Models should not be fuzzy, they must be precise and serve a clear purpose, there could be more than one model for the same phenomena but they must serve different purposes. We imitate the same state of affairs in simulation with different variables and then see the behavior of our simulation or model. The use of many parameters in modeling or simulation makes it difficult to guess the actual role and influence of individual parameters. Artificial Neural Networks and Bio-hardware are good examples of modeling in computer science.

1.2 Theory

The word 'theory' is derived from the Greek word 'theorein', which means 'to look at'. Theory is "someone else's" knowledge that is confirmed from reality, also called Established knowledge. We can define theory as 'a set of statements closed under certain rules of inference'. Humans construct theories in order to explain, predict and master

phenomena (e.g. inanimate things, events, ecosystem etc) and it is usually necessary for the theory to characterize a critical experiment i.e. an experimental result which cannot be predicted by any established theory. A theory must be testable and falsifiable (as opposite to religion or faith). We humans prefer to generalize concepts and ideas, which we call a theory. According to Albert Einstein [3] "Theory should be as simple as possible but not simpler." You cannot disprove theories by a single instance, theories stand until disproved and long standing theories become laws. There is always a gap between a conjecture (which do not have the backing of experimental results) and a theory (which is backed by experimental data).

According to Stephen W. Hawking [5] "A theory is a good theory if it satisfies two requirements: It must accurately describe a large class of observations on the basis of a model that contains only a few arbitrary elements, and it must make definite predictions about the results of future observations." We are in continuous search for more and more refined theories and generally, we learn more when theories collapse. Theories are always better than hypotheses, conjectures, beliefs, anecdotes, and 'wisdom of the ancients'. The Evolution theory and DNA are the good examples of using theories in computer science.

1.3 Prediction

Author likes to define a prediction as a bold step supplemented by a model or theory (or both). Prediction has been the ancient wish for thousands of years to see the future, or predict it. Soothsayers use stars for this, magicians use magic-balls and we, the scientists, use theories and models for the same. A prediction is defined as a logical consequence of an observation or what will happen under specific conditions. It could be as simple as "If A is true, then B" to as complicated as a traffic jam and stock market. Data mining and Evolutionary algorithms are good examples of prediction in computer science.

1.4 Evidence

Evidences are observable events which tend to prove or disprove a proposition [2]. Evidences are illusive, one can only have the suspects, what you find and how you interpret it make the decision very tough. We can only make intelligent guesses under certain conditions and given

knowledge. Stigmergy and pheromone in self-organized systems are good examples of evidence.

1.5 Patterns

A pattern can be defined as a sequence or repetition of same rules, images or steps which can be used to generate complete (or part) of some phenomena, design, system or behavior [6]. If we look around, the whole world seems to be a collection of different patterns from the simplest to most complex ones. Whether it is the elliptical orbits of planets around the Sun, or branching of trees, golden ratio or stripes, zigzags or designs on sea shells, Fibonacci series or prime numbers, we can see patterns everywhere. There are so many recent developments in computer sciences based on natural patterns, like ACO (Ant Colony Optimization) based on Ant Patterns, Routing algorithm based on honey bee dance, Swarm-Art based on patterns emerges by social insects' interactions and several applications based on Cellular Automata.

1.6 Investigation

Investigation is the result of curiosity when we want to learn hidden behaviors or systems. If we don't have enough information or explanations about system, questioning is the only way to get results. If you don't ask, you will not get any answers. One can classify self-organization as an example of Investigation, where agents react using the local knowledge and feedback.

1.7 Trial and Error

It is an ancient 'brute force' method of solving problems in which many solutions are tried until errors are reduced or eliminated. It needs no or very little apriori knowledge to find solutions. Many discoveries of today's scientific world are the consequences of the same method. There could be two subtypes of this method - random and 'most likely to least likely' solutions search [1]. Either way, there are few drawbacks to this approach, such as it just focuses on finding solutions and does not care about why it works. There is no generalization possible in this approach and we do not know whether it is the most optimal solution or not.

One can think of Bogo-Sort as one of the examples of this technique. Bogo-Sort is a method to sort a deck of cards by repeatedly throwing the deck in the air, picking the cards up at random, and then testing whether the cards are in sorted order. This method is time consuming but works well in case of small number of hypotheses. TSP (Traveling Salesman Problem) and ACO (Ant Colony Optimization) are good examples of Trial and Error in computer science.

1.8 Context

Context can be defined as a set of facts or circumstances that surround a situation or event. The usage of something can entirely depend on its context. It is the context that determines how you see objects and how you tackle problems; a solution that works in one situation may not work in another (similar) situation with different contexts. Emergence is a good example of Context in computer science.

1.9 Adaptation

Adaptation is any change in the structure or function of an organism, which makes it better suited to its environment (e.g. the torpedo body shape of tuna allows for fast swimming in the open ocean). Adaptation is a natural process and hard coded in our genes. Those who do not adapt will die; it's a never ending process. The presence of this method can be found in neural networks, classifiers and genetic algorithms.

2 Conclusion

There are different possible ways of finding a solution. Bio-inspired computing or solution based on nature can be a good alternative of traditional method and model for computing. When searching for a solution, one should keep in mind these ways to work on. These are efficient and proven ways nature has been using for past 3.8 billion years.

3 References

- [1] Allen B. Tucker. *Computer Science Handbook, Second Edition*. Chapman & Hall/CRC; 2nd edition, 2004.
- [2] A. Best. *Evidence: Examples and Explanations (Examples & Explanations Series)*. Aspen Publishers; 5th edition, 2004.
- [3] A. Einstein. *World As I See It*. Citadel Press; Reissue edition, July 1993.
- [4] Bruce Hannon Bernard Mc Garvey. *Dynamic Modeling for Business Management: An Introduction (Modeling Dynamic Systems)*. Springer, 1st Edition, 2004.
- [5] Stephen W. Hawking. *The Brief History of Times*. Cambridge Press, 1992.
- [6] W. D. Hillis. *The Pattern on the Stone: The Simple Ideas That Make Computers Work (Science Masters Series)*. Basic Books; 1st edition, 1998.