Remarks on Evolutionary Hamiltonian Graph Theory

Zh.G. Nikoghosyan*

Institute for Informatics and Automation Problems
National Academy of Sciences
E-mail: zhora@ipia.sci.am

We present an appropriate complement to the large palette of evolutionary theories (such as evolutionary psychology, economics, computation, algorithm, programming, game theory, thought and so on) by a new discipline concerning mathematics.

The concept of NP-completeness was introduced in 1971 by Stephen Cook [1], who conjectured that NP-complete problems are not solvable in polynomial time. Today, this conjecture seems much more reasonable motivated by the fact that the developments arising around various NP-complete problems have undergone a natural gradual growth and evolution, generating a great diversity. These developments provide an exclusive valuable domain beyond biology with continuously growing diversity and well described environment-origins-gene structures relations.

We focus on one of the most heavily studied areas in graph theory, that joins together a number of NP-complete cycle problems, called large cycles theory - a simplified version of well-known hamiltonian graph theory, to show that the individuals (claims, propositions, lemmas, conjectures, theorems) on this subject evolve and adapt to their environment generating a great diversity by an iterative process from simplicity to complexity, from primitive beginnings (such as "every complete graph is hamiltonian") to best possible theorems by certain hereditary mechanisms.

Large cycles theory plays the role of a general environment and various statements, including claims, propositions, lemmas conjectures and theorems, play the role of individuals in a population.

This simplified and valuable model has a number of advantages with respect to biology and can be useful towards better understanding the universal mechanisms to explain evolution in a wide variety of domains outside of biology.

(a1) Large cycles theory, originated about 60 years ago, evolves much more rapidly than living forms on Earth, originated about 3.7 billion years ago.

(a2) The origins of theorems in large cycles theory can be strongly determined by exact branchings of the tree of developments.

(a3) Genetic units and hereditary mechanisms in large cycles theory are much more simpler than gene structures of living forms.

We distinguish the following evolution mechanisms in large cycles theory:
(b1) improvements (vertical evolution),
(b2) modifications (horizontal evolution),
(b3) vertical generalizations (vertical evolution leap based on inductive reasoning),

($b4$) horizontal generalizations (horizontal evolution leap based on inductive reasoning),
($b5$) involving new genetic units (genome extension).

Definition 1. Improvement is one of the following procedures:
($c1$) relaxing one of the conditions in theorems and preserving the conclusion,
($c2$) strengthening the conclusion and preserving the conditions.
Definition 2. Modification is one of the following procedures:
($d1$) relaxing of some conditions, at the same time strengthening some others, under the same conclusion,
($d2$) relaxing of some conditions, at the same time relaxing the conclusion,
($d3$) strengthening of some conditions, at the same time strengthening the conclusion.
Definition 3. Vertical generalization is a leap in improvement process based on inductive reasoning toward finding best possible results.
Definition 4. Horizontal generalization is a leap in modification process based on inductive reasoning toward finding best possible results.

We deal a special attention to so called "fundamental theorems", by observing that all theorems in large cycles theory have descended from a number of common ancestors via generalizations, called fundamental theorems. Remember, that the term "fundamental result" is used in various fields of science to characterize mainly the central and most important results in the area.

References

[1] S.A. Cook, The Complexity of Theorem-Proving Procedures, Proceedings, Third Annual ACM Symposium on the theory of computing, ACM, New York (1971) 151-158.