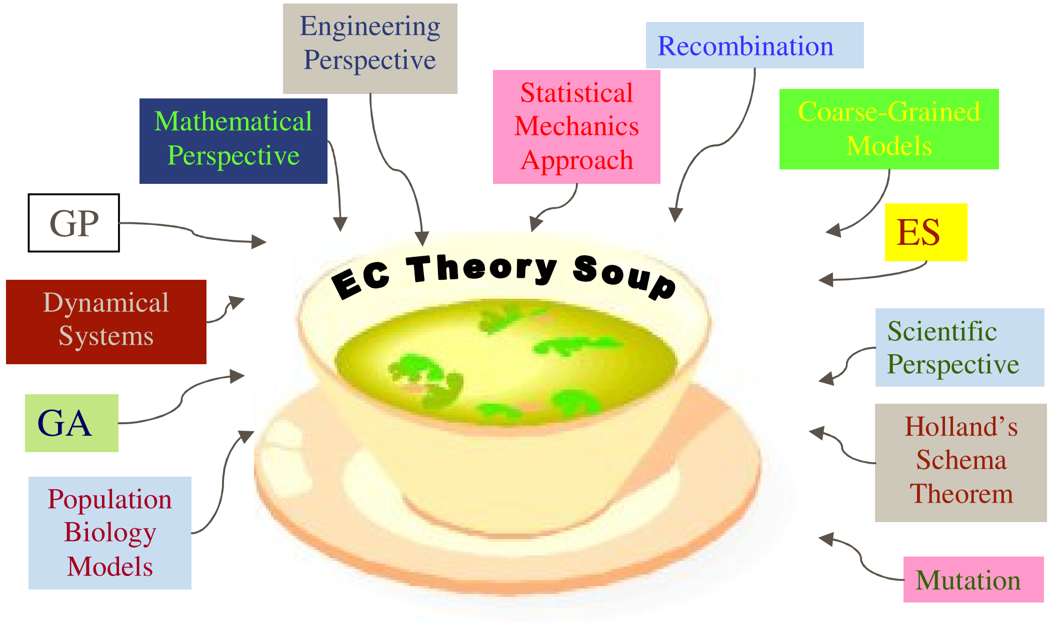
Chris Stephens' invited talk at EuroGP 2003.

EC Theory – A tale of Elephants, Blind Men and Soup!

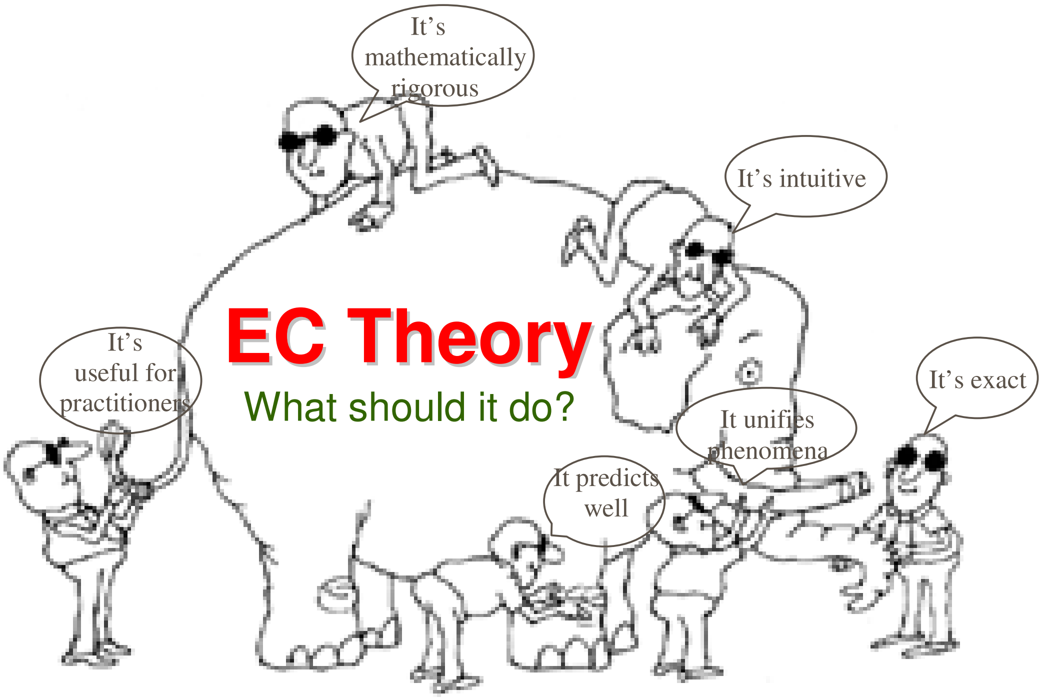
A "biased" summary by Alberto Moraglio of Essex University

The last speech of EuroGP 2003 conference was given by Professor Chris Stephens. Being the closing speaker, to keep the audience interest high after three full days of parallel streams of conferences and workshops is quite a challenge. Chris managed it very well by presenting Evolutionary Computation Theory as a tale for children including the *EC theory soup*, the *EC theory elephant* and *parent-guidance warning marks* for slides containing hardcore mathematics.



The focus of the presentation was on introducing dynamic modelling of evolutionary algorithms, highlighting the multitude of approaches and motivations to it, i.e. EC theory soup ingredients. These differences originate from very different disciplines and their understanding of what the field is or should be about, i.e. different (myopic) views on the EC theory elephant. The need for a unifying theory was acknowledged and a promising route toward unification presented.

The EC area is complicated by the customisation of EC algorithms to specific problems - everybody does their own thing! EC is a big tent in which every practitioner can express his/her artistic side and sign his/her own piece of art introducing excitingly novel features, which for the No Free Lunch Theorem, is guaranteed to be as futile as all the others, at least on average (and in theory!).



“Taxonomy and Universality, Taxonomy and Universality” seem to be the words Chris likes (to repeat) most and, according to him, these are the words that point the way forward to establish order in the EC field. Taxonomy (classification) of EC algorithms should be based on theoretical understanding of the dynamics of algorithms rather than, based on historical/political issues or on superficial similarities among algorithms (e.g. solution representation). Universality is about seeing different evolutionary algorithms and noting that, although superficially different, they display for certain quantities the same or similar behaviour - or maybe better put - they can all be fruitfully described by the same "label" (hence they admit a theory encompassing them all). The periodic table is a great example: each halogen element is different one from the other, but for a large number of properties they exhibit similar - "universal" behaviour (due to the fact that they all have 7 electrons in their outer shell). A "universality class" of systems all share similar properties for a set of variables that are "universal".

What should EC theory do? What should it apply to? What’s the best approach? These are the central questions characterising the state of the art of EC theory today, suggesting that the field is still conceptually fragmented and fairly immature. The unifying approach proposed here is to understand the pair Evolutionary Algorithm/Problem as a black box taking as input EA parameters, fitness landscape and initial population; then using a similarity metric on the resulting dynamic to classify the Algorithm/Problem. In this scenario, toy problems are useful means of identifying the essential characteristics of EAs. (In the EuroGP debate the utility/futility of toy problems was discussed among other things).

EC theory is philosophically difficult for most because it is not clear what it is or should be about. Psychologically it doesn’t fulfil practitioners’ needs, the EC “expectation gap”, so making EC theorists feel frustrated and EC practitioners somewhat dismissive of theory.

The essential components for formalizing the dynamics of a generic EA, i.e. the "Bare Necessities" of EC theory according to Chris, comprises: a set of objects (strings, trees, and in general more complex structures) defined over a configuration (metric) space; a fitness function assigning to each object its fitness; interactions among objects to model selection, mutation and recombination. An EA is an algorithm that takes as input a population of objects and a fitness function, at a given time, and gives as output, the population at a later time. The dynamics (evolution) is therefore a Markov process (describable by dynamical system equations) generated by a set of interactions (operators) that act stochastically.

The "Bare Necessities" model describes the microscopic dynamics of the system – the probabilistic behaviour of every single object in time. There are at least two major advantages to pass from the microscopic level to a coarse grained description of the system. First, drastic reduction of the number of equations makes the difference in terms of computational tractability. Second, "emergence of effective degrees of freedom" allows the system dynamics to be understood in terms of independent macro-variables greatly increasing explicative power.

Coarse graining involves the choice of the basic block (aggregation of microscopic objects) to observe the system in this new light. There are various choices possible: phenotypes, schemata or Building Blocks, to mention a few. However the most natural coarse graining depends on the operators, the fitness landscape and the population. Mathematically speaking, coarse graining can be done via projection (e.g. on phenotype aggregation - unitation) or via coordinate transformations (e.g. useful for observing the system through the Building Blocks basis).

What should it do?	**"Old Stuff"**	**"Bare Necessities"**	**Coarse-Grained**
Exact	No	Yes	Yes
Mathematically rigorous	Yes??	Yes?	Yes?
Unifies phenomena	Yes/No	Yes/No	Yes
Intuitive	Yes/No	No	Yes
Predicts well	No	No	No
Useful for practitioners	Yes/No	No	Yes/No

Comparison of models: Bare Necessities vs. Old Stuff (schema theorem and Building Block Hypothesis) vs. Coarse Graining

The "Bare Necessities" model, together with coarse graining, is a very general framework pointing the way to a grand unification of different branches of EC, such as Genetic Algorithms and Genetic Programming, and different theoretical models, such as Vose's model and Holland's Schema Theorem. After all they are all Markov chains and they are all dynamical systems.

Overall, another chapter in the jungle book of Evolutionary Computing Theory (editor's note).

The EC theory soup and EC theory elephant pictures appear here by courtesy of Professor Chris Stephens.