Towards Estimation of Distribution Algorithm Explainability

Daniel Conde, Concha Bielza, Pedro Larrañaga

Computational Intelligence Group, Universidad Politécnica de Madrid (Spain)

Abstract

Lately there is an increasing concern in the heuristic optimization field about topics such as explainability, interpretability and transparency. In 30 years of research in Estimation of Distribution Algorithms (EDAs) little to no attention has been given to these aspects. Some of these issues can be solved by adopting a few visualization tools common in practical applications of other Evolutionary Computation algorithms, usually aimed to support the reasoning and analysis about the algorithm behavior. However, EDAs present particular features different from other optimization algorithms and new EDA-specific visualization tools should be designed that go beyond the standard representation of the probabilistic graphs over generations. Some relevant questions that naturally arise are: How can we track the evolution of the probabilistic model performing the search? How is the evaluation of individuals simulated from this probabilistic model covering the problem landscape? Can these visualization tools be useful to decide whether the evolution of the probability distributions characterizing the selected individuals has converged so as to stop the algorithm? In this work we investigate new visualizations to address some of these questions that hopefully will add valuable information to the user at a glance in an easy to understand way.

In particular, we propose to track dynamically the univariate distribution of the variables of the promising solutions, those that update the probabilistic model for the next iteration. This is a clear aid for univariate EDAs and to understand the algorithm search in general. In multivariate EDAs we use heatmaps to summarize the bivariate variable relationships the algorithm has encountered in the search, easily highlighting the most interesting (frequent) ones. Additionally, we incorporate some metrics to compare the different probability distributions learned in consecutive iterations during the algorithm execution.

We will showcase these tools by analyzing the behavior of different EDAs in well-known benchmark problems. The software that we have developed has been integrated in the freely available Python library EDAspy, which is entirely dedicated to EDAs.

We believe these techniques can help the user to tune and gain understanding about the algorithm, and possibly to identify particular problem features encountered during the EDA run, improving the reach and usability of these algorithms.