Surrogate-assisted evolutionary computation: Recent advances and future challenges

In evolutionary algorithms, usually the fitness value of an individual can be acquired by evaluating the fitness function that is defined. Sometimes in real world situations however, evaluating the fitness function may be very costly and/or time consuming, which makes it difficult to evaluate the fitness of large population for many generations.

To solve this issue, surrogates can be introduced which are usually machine learning models which produce similar results as the real fitness function but are quicker and cheaper to evaluate. Real fitness function is used together with the surrogates in order to prevent false results caused by “bad” surrogates. Strategy for using the surrogates correctly in evolutionary optimization, is often called “model management”.

The first use of approximate fitness functions was first reported in mid 80s and the use of computational models for representing fitness functions was reported in mid 90s. Even though the roots of using surrogates in evolutionary optimization reach quite far, still surrogate model management techniques, especially for problems with high dimensions, remain challenging research topic.

Early methods for surrogate-assisted evolutionary computing used only surrogates and assumed that surrogates represented the original problem so accurately that relying solely on surrogates, would produce “good enough” solution to the optimization problem. This however introduced a problem if surrogate didn’t represent all the aspects of the original problem, the algorithm might produce an optima which actually doesn’t exist in the original problem. This emphasizes the need for model management methods which uses surrogates together with the original fitness function.

In addition to fitness evaluation, surrogates can be used in other evolutionary operations, like population initialization, cross-over, mutation and local search. For example, surrogate can be used to filter out poor individuals during initialization, cross-over or mutation phases.

Roughly methods for surrogate management can be divided to three categories, individual-based, generation-based and population-based. In individual-based methods surrogates are used for evaluating some individuals while the original fitness function is used for other. In generation-based methods surrogates are used for evaluation during some generations and the original function in others. Population-based methods in the other hand create multiple sub-populations where surrogates are used for evaluating some of the populations and the original fitness function is used for the rest, also individuals can move between sub-populations.

Additionally, surrogates can be used also for other purposes in evolutionary algorithms. These use cases include for example reducing noise in noisy fitness functions and thus generating more robust solutions which are not that easily affected by small changes in the environment. Other use case in interactive evolutionary methods is to replace the decision maker (human) with a surrogate and thus preventing the human from getting tired.