Committee-Based Active Learning for Surrogate-Assisted Particle Swarm Optimization of Expensive Problems

In multiobjective optimization, sometimes objective functions maybe costly and/or time consuming to evaluate. For this reason, using surrogates, mathematical models instead of original objective functions can be used. However, many of these algorithms only work on quite low dimensional problems or still require thousands of function evaluations. To overcome this issue, paper proposes surrogate assisted particle swarm optimization algorithm which uses committee-based active learning model management strategy.

In model management strategy, goal is to identify promising solutions which could then be re-evaluated with the original expensive function. In typical cases two criteria are used to identify these promising solutions. First is to select the optimum surrogate model and second is to choose surrogate model which has the most uncertainty in its prediction. In many algorithms these two criteria are usually combined into one, which presents problem when controlling algorithms exploration / exploitation capabilities. To overcome this, the algorithm proposed in the paper uses active learning strategy. There exist many active learning algorithms, but the algorithm proposed in the paper strategy similar to query by committee algorithm (QBC) where each model can vote on the output of series of candidates and the query with maximum disagreement is re-evaluated and added to the dataset.

The proposed algorithm is based on particle swarm optimization which is algorithm that is inspired by swarms of social animals like fish schools or bird flocks. Polynomial regression (PR), radial basis function (RBF), and kriging models are used to form surrogate ensemble. To update the ensemble, samples are selected for re-evaluation using three different criteria. Those criteria are controlled by two model management strategies. One is based on QBC which globally searches for the most uncertain and the best solution and other locally searches the best solution around current optimum. First the algorithm starts with the global model management strategy and switches to local one when no further improvement can be made. Same goes for the local strategy, model switches again back to the global strategy when the fitness value cannot be improved using the local strategy.