The (1+λ) evolutionary algorithm with self-adjusting mutation rate

We propose a new way to self-adjust the mutation rate in population-based evolutionary algorithms. Roughly speaking, it consists of creating half the offspring with a mutation rate that is twice the current mutation rate and the other half with half the current rate. The mutation rate is then updated to the rate used in that subpopulation which contains the best offspring. We analyze how the (1 + A) evolutionary algorithm with this self-adjusting mutation rate optimizes the OneMax test function. We prove that this dynamic version of the (1 + A) EA finds the optimum in an expected optimization time (number of fitness evaluations) of O(nA/log A + n log n). This time is asymptotically smaller than the optimization time of the classic (1 + A) EA. Previous work shows that this performance is best-possible among all A-parallel mutation-based unbiased black-box algorithms. This result shows that the new way of adjusting the mutation rate can find optimal dynamic parameter values on the fly. Since our adjustment mechanism is simpler than the ones previously used for adjusting the mutation rate and does not have parameters itself, we are optimistic that it will find other applications.