STA 5107/4013: Home Assignment #8

Spring 2021/Due Date: April 1st

In this homework we will study several tools for optimizing functions on Euclidean domains: $D \subset \mathbb{R}^1$ or $D \subset \mathbb{R}^2$. Let $E : D \to \mathbb{R}$ denote an objective function and our goal is to solve for:

$$\hat{x} = \text{argmin}_{x \in D} E(x) \ .$$

Your task is to implement these algorithms and to study their performance on a following objective functions. For each problem and algorithm, show the following plots: (1) a plot of the convergence of the iterative process $\{Px_k\}$, (2) plot of the objective function $\{E(x_k)\}$. Show two runs of these results from random initial conditions.

1. $E : [-1, 7] \to \mathbb{R}$ given by

$$E(x) = (1 - e^{-x^2} - 2e^{-(x-3)^2} - e^{-(x-6)^2}) \ .$$

   For this case, study all the three algorithms given below.

2. $E : [-1, 1]^2 \to \mathbb{R}$ given by:

$$\begin{aligned} E(x, y) &= (x \sin(20y) + y \sin(20x))^2 \cosh(\sin(10x)x) \\ &\quad + (x \cos(10y) - y \sin(10x))^2 \cosh(\cos(20y)y) + 0.01(x^2 + y^2) \ . \end{aligned}$$

   For this case, implement only the simulated annealing algorithm.

Algorithm 1 *(***Deterministic-Gradient Search***)*

1. *Initialize x_1 randomly in the domain D. Set $k = 1$. Select a δ (for example $\delta = 0.1$).*
2. *Compute the gradient of f at x_k; call it $\nabla E(x_k)$.*
3. *Update the state using:*

$$x_{k+1} = x_k - \delta \nabla E(x_k) \ .$$

4. *Check if $\|\nabla(x_k)\|$ is small. If not, set $k = k + 1$, and go to Step 2.*

Algorithm 2 *(***Stochastic-Gradient Search***)*

1. *Initialize* x_1 *randomly in the domain* D. *Set* $k = 1$. *Select a* δ *and* T *(for example* $\delta = 0.1$ *and* $T = 1$*).*
2. *Compute the gradient of* f *at* x_k*; call it* $\nabla E(x_k)$*.*
3. *Generate a random vector* z_k *(same size as* x_k*) from a multivariate normal distribution with mean zero and variance* I*.*
4. *Update the state using:*
$$x_{k+1} = x_k + (-\delta \nabla E(x_k) + \sqrt{2\delta T} z_k) \ .$$
5. *Check if the maxIter is reached. If not, set* $k = k + 1$*, and go to Step 2.*

Algorithm 3 *(***Simulated Annealing Algorithm***)*

1. *Initialize* x_1 *randomly in the domain* D. *Set* $k = 1$*, maxIter = 5000. Select a* T *(for example* $T = 1$*).*
2. *Generate a random vector* z_k *(same size as* x_k*) from a multivariate normal distribution with mean zero and variance* I*.*
3. *Generate a candidate using:*
$$y_k = x_k + \sqrt{T} z_k \ .$$
(You have to ensure that $y_k \in D$ *since* E *may not be defined outside* D*.)*
4. *Set:*
$$x_{k+1} = \begin{cases} y_k, & \text{with probability } \rho(x_k, y_k) \\ x_k, & \text{with probability } 1 - \rho(x_k, y_k) \end{cases}$$
where $\rho = min(e^{-(E(y_k) - E(x_k))/T}, 1)$*.*
5. *Check if the maxIter is reached. If not, set* $T = T * \alpha$*,* $k = k + 1$*, and go to Step 2. Choose* $\alpha = 0.995$*.*