Optimisation Basics 1 — Practice Questions

CM52054: Foundational Machine Learning
Practice set with fully worked answers

1 Concept & Short-Answer

1) **Why do we often study convex optimisation even when many real-world problems are non-convex?**
Answer: Because theory and guarantees are much stronger for convex problems; a common strategy is to identify/solve convex subproblems inside a larger non-convex problem ("perform convex within non-convex").

2) **Name four broad ways to obtain solutions to optimisation problems discussed in the lecture.**
Answer: (i) Naive exhaustive (grid) search; (ii) Random search; (iii) Numerical/meta-heuristics (e.g., simulated annealing, evolutionary/PSO, MCMC/SMC); (iv) Gradient-based direction search (first/second order).

3) **What does "gradient-based direction search" mean? Give two examples.**
Answer: Use derivative information to choose the next iterate's direction and step size. Examples: steepest gradient descent (1st order) and Newton's method (2nd order using the Hessian).

4) **In one sentence, what is simulated annealing trying to mimic and why?**
Answer: It mimics heating and controlled cooling to probabilistically escape poor regions and settle into low-energy (low-objective) states; the "temperature" controls randomness.

5) **When might random search outperform grid search?**
Answer: In high-dimensional spaces where grid search suffers from the curse of dimensionality; random search can cover space more efficiently for a fixed budget.

6) **Write the generic iterative optimisation loop presented in the 2D example.**
Answer: Initialise x_0. For $t = 0, 1, \ldots$ until termination: (i) choose a direction p_t; (ii) choose step size α_t; (iii) update $x_{t+1} = x_t + \alpha_t p_t$.

7) **List two common termination conditions for iterative methods.**
Answer: Small gradient norm $\|\nabla f(x_t)\| \le \epsilon$; or small step $\|x_{t+1} - x_t\| \le \epsilon$; or maximum iterations reached.

2 Calculations (show your working)

8) **Partial derivatives & gradient.**
Let $f(x_1, x_2) = x_1^2 + x_2^2$.
(a) Compute $\frac{\partial f}{\partial x_1}$ and $\frac{\partial f}{\partial x_2}$.
(b) Write $\nabla f(x)$ and evaluate it at $x = (2, 2)$.
Answer: (a) $2x_1,\ 2x_2$. (b) $\nabla f(x) = [\,2x_1,\ 2x_2\,]^\top$; at $(2, 2)$ it is $[\,4, 4\,]^\top$.

9) **One gradient-descent step.**
With the same f and learning rate $\alpha = 0.1$, start at $x_0 = (2, 2)$. Perform one steepest-descent step.
Answer: $x_1 = x_0 - \alpha\nabla f(x_0) = (2, 2) - 0.1(4, 4) = (1.6, 1.6)$.

3 Worked Mini-Cases

10) **1D "look-around" heuristic.**
You can only evaluate f locally. At x_t, you probe $f(x_t \pm \delta)$ and find $f(x_t - \delta) < f(x_t + \delta)$. (a) Which direction should you move? (b) Give one drawback of this approach.
Answer: (a) Move left (toward $x_t - \delta$). (b) It's purely local, so it can zig-zag, get stuck near plateaus, and doesn't scale well; it gives no global guarantees.

4 Challenge Extensions (optional)

11) **Design a stopping rule combining gradient and progress.**
Answer: Stop when $\|\nabla f(x_t)\| \leq \epsilon_g$ *and* $\|x_t - x_{t-1}\| \leq \epsilon_x$ for k consecutive iterations, or if t exceeds a maximum budget.