COMP9602 — Convex Optimization
Assignment 3 (12%) [Learning Outcome 3]
Due by: 23:59 Friday December 8 2023

— Submission: please submit a softcopy through Moodle following the steps below:

(1) Login Moodle.

(2) Find "Assignments" in the left column and click "Assignment 3".

(3) Click "Add submission", browse your solution file and save it. Done.

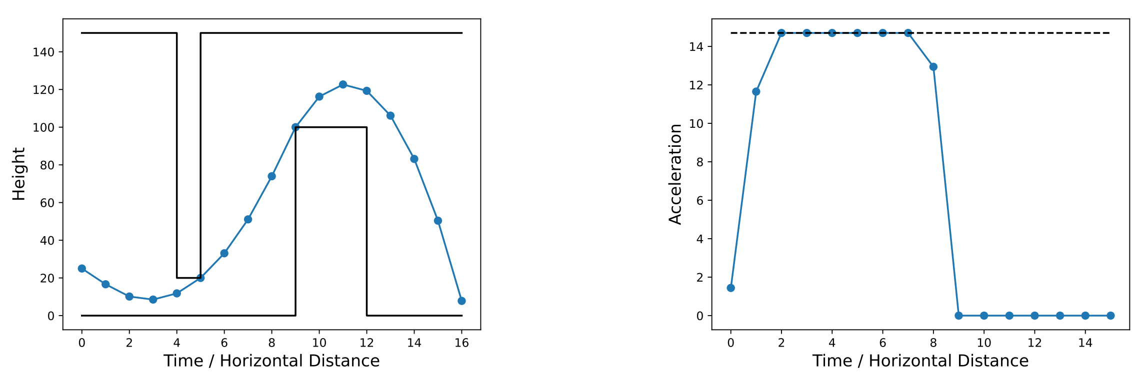
(4) You will receive an automatic confirmation email, if the submission was successful.

(5) You can "Edit submission" to your already submitted file, but ONLY before the deadline.

(6) Only .zip files are accepted, and at most one file is allowed to be uploaded.

— Total mark is 100.

We consider the problem of optimizing a 2D drone's trajectory across obstacles. We will use discrete time $t = 0, 1, \ldots, T$ (seconds), to describe the motion of the drone. Suppose the drone maintains a constant horizontal speed of 1m/s, and its vertical speed is v_t at time t. That is, if the drone is at location (x_t, y_t) at time t, its location will be $(x_t + 1, y_t + v_t)$ at time $t+1$. Due to gravity, the drone's vertical velocity diminishes by $g = 9.8$ each second, countered by an upward acceleration of a_t per second generated by its propellers. Hence, the relation between v_t and v_{t-1} is given by $v_t = v_{t-1} + a_t - 9.8$.
Initially, at $t = 0$, the drone is positioned at $(0, y_{init})$ (thus, $x_t = t$), and $v_{-1} = 0$ (i.e., $v_0 = a_t - 9.8$). Then, it must bypass a series of obstacles defined by upper and lower vertical limits, h_x and l_x, at each integer x (equivalently, t), by regulating the acceleration a_t at each time step within $t = 0$ to $t = T - 1$. Assume the propellers cannot generate downward force, and the maximum attainable upward acceleration is a_{max}.
Additionally, the energy consumed by the drone's motors at each time step is related to the upward acceleration a_t as $\phi(a_t)$, which is a positive, increasing and convex function. The objective is to determine the optimal accelerations (a_t's) to minimize the total energy consumption while circumventing all obstacles.



(a) Optimal trajectory of the drone (blue). Black lines show obstacles **h** and **l**.

(b) Optimal acceleration at each time step. Black dotted line indicates a_{max}.

Figure 1: An example of a minimum energy trajectory.

(a) Formulate the problem as a convex optimization problem (for the general case, not specific to the above example) with only *inequality constraints*. Your formulation should be convex for any function ϕ that is positive, increasing, and convex (no further assumptions on ϕ). Feel free to introduce new variables (and define them clearly). Briefly explain why your problem formulation is convex and why it is equivalent to the problem stated above.

(b) Consider the following energy profile

$$\phi(a) = 1 + a + a^2 + a^3.$$

We aim to solve the optimization problem using the **ellipsoid method**. For your objective function and each (set of) constraints, derive a formula to obtain a sub-gradient that will be used in the ellipsoid algorithm.

(c) Implement the ellipsoid algorithm to solve the problem with ϕ defined in **(b)** in Python or MATLAB. You are given two problem instances, one with $T = 2$ (instance A) and one with $T = 32$ (instance B). For each instance, the constants $y_{init}, a_{max}, \mathbf{h}, \mathbf{l}$ is given:

- Python: You are given `2d_instance.npz` (for instance A) and `32d_instance.npz` (for instance B). Numpy is required for loading the data files. To load data from the `.npz` files, use:

  ```
  import numpy as np
  instance = np.load(file, allow_pickle=False)
  ```

  where `instance` is a dictionary containing the loaded numpy arrays with keys `a_max`, `y_init`, `lx`, `hx`, and `file` is the path to the downloaded instance data.

- MATLAB: You are given `2d_instance.mat` (for instance A) and `32d_instance.mat` (for instance B). To load data from the `.mat` files, you can directly:

  ```
  load(file)
  ```

  where `file` is the path to the downloaded instance data. Variables `a_max`, `y_init`, `lx`, `hx` will be loaded into the workspace.

`a_max` and `y_init` are scalars; `lx` and `hx` are $\mathbf{T+1}$ dimensional arrays specifying l_x, h_x, $x = 0, \dots, T$ (**Note**: x starts from **zero** for l_x and h_x). For instance A, please use $\epsilon = 0.001$ as the **stopping criteria**. For instance B, please use $\epsilon = 0.1$.

You should produce the following figures:

- For both instances, plot $f_{best}^{(k)}$ (the best objective value found so far) and $f^{(k)} - \sqrt{(g^{(k)})^T P^{(k)} g^{(k)}}$ (which is a lower bound on f^*) versus iteration number in the same figure. For constraint iterations (i.e., iterations where the current $x^{(k)}$ is infeasible), you should record `None` or `NaN` for the lower bound.

- For both instances, plot the drone's optimal trajectory (height vs. time/horizontal distance). Optionally you can also plot $\mathbf{h}$ and $\mathbf{l}$ in the same figure (as in Fig. 1a) for better visualization.

- For both instances, plot the drone's optimal accelerations (a_t's) vs. time/horizontal distance as in Fig. 1b.

- For instance A, plot the ellipsoid ($\mathcal{E}^k$) found in each iteration during the execution of the algorithm. If your code runs more than 20 iterations, you only need to plot the first 20 ellipsoids (for better visualization). Optionally, you can use a gradient color map (e.g., viridis) to better differentiate the ellipsoids generated in different iterations.

Additionally, please include the **optimal objective value** (energy consumed) in your solution sheet.

Submission: Please submit a .zip package containing solution sheet in .pdf, source code file(s) to implement the algorithm (which should be executable), and the figure files. You are encouraged to provide a readme file to explain your code, if you deem it helpful.