EPFL | MGT-418 : Convex Optimization | Project 5

Questions – Fall 2023
Due on December 13, 11:59PM

Natural Disaster Prevention with Wireless Sensor Networks (graded)

Description

In natural disaster prevention, the use of wireless sensor networks has emerged as a powerful tool for safeguarding communities from the adverse effects of extreme weather events such as floods. A network of smart sensors, strategically placed along riverbanks and continuously monitoring water levels, can provide early warnings and mitigate the impact of disasters. In this project, we use optimization to find the optimal locations of these wireless sensors. Specifically, given n candidate locations for the sensors, we maximize the effectiveness of the sensor network by placing as many sensors as possible.

However, we cannot place sensors too close to each other. Sensor interference occurs when two sensors in a network are too close to each other such that they disrupt each other's operation, often leading to data collisions or degraded performance due to shared resources. If the distance between the two sensors is less than or equal to 25 m, only one should be placed. An optimal trade-off between maximizing coverage and minimizing interference is determined by the integer program

$$\begin{aligned} \max_{x \in \mathbb{R}^n} \quad & \sum_{i \in V} x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1 \quad \forall (i,j) \in E \\ & x_i = \{0,1\} \quad \forall i \in V, \end{aligned} \tag{IP}$$

where $V = \{1, \ldots, n\}$ denotes the set of n possible sensor locations, and $E \subseteq V \times V$ denotes the set of all pairs of locations that are too close to each other and would thus lead to interference if they both accommodated a sensor. The binary decision variable x_i is set to 1 if a sensor is placed at location $i \in V$ and to 0 otherwise. The integer program (IP) is discrete and thus nonconvex. In fact, it is NP-hard, that is, the time required to solve it grows exponentially with n for all known algorithms. Therefore, we will explore convex relaxations that can be solved in time polynomial in n.

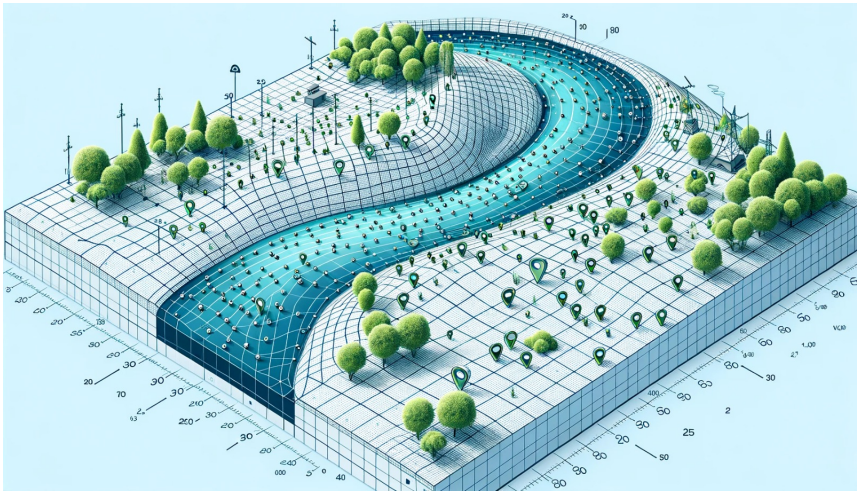

Figure 1: Riverbank with possible sensor locations

Questions

1. **QCQP Reformulation (5 points):** Show that the integer program (IP) and the quadratically constrained quadratic program (QCQP) below are equivalent.

$$\begin{array}{ll} \max\limits_{x\in\mathbb{R}^n} & x^\top x \\ \text{s.t.} & x_i^2 = x_i \quad \forall i \in V \\ & x_i\, x_j = 0 \quad \forall (i,j) \in E \end{array} \tag{QCQP}$$

2. **SDP Relaxation (15 points):** Show that (QCQP) admits the SDP relaxation (SDP), which is obtained by lifting (QCQP) to a higher dimension and then relaxing a rank constraint. *Hint: See pages 219 and onwards in the lecture slides.* Derive Q_i and e_i for all $i \in V$.

$$\begin{array}{ll} \max\limits_{x\in\mathbb{R}^n,\, X\in\mathbb{S}^n} & \operatorname{tr}(X) \\ \text{s.t.} & \operatorname{tr}(Q_i X) - e_i^\top x = 0 \quad \forall i \in V \\ & X_{ij} = 0 \qquad\qquad\quad \forall (i,j) \in E \\ & \begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \succeq 0 \end{array} \tag{SDP}$$

3. **Lagrangian Dual (40 points):** Show that the Lagrangian dual of (QCQP) is given by problem (D-QCQP) below. Does strong duality hold? Justify your answer. *Hint: Use epigraphical variables and recall the Schur complement.*

$$\begin{array}{ll} \min\limits_{\mu\in\mathbb{R},\, \lambda\in\mathbb{R}^n,\, W\in\mathbb{S}^n} & -\mu \\ \text{s.t.} & W_{ij} = 0 \quad \forall (i,j) \notin E \\ & \begin{bmatrix} \mu & \frac{1}{2}\lambda^T \\ \frac{1}{2}\lambda & I + W + \operatorname{Diag}(\lambda) \end{bmatrix} \preceq 0 \end{array} \tag{D-QCQP}$$

4. **Dual of the Dual (25 points):** Show that the Lagrangian dual of (D-QCQP) is equivalent to problem (SDP). Does strong duality hold? Justify your answer.

5. **Implementation (15 points):** The small and large datasets provided in `points_small.npy` and `points_large.npy` contain the coordinates of 6 and 150 possible sensor locations, respectively. Use these coordinates to construct the set E. Specifically, a pair of locations should be included in E whenever their Euclidean distance is smaller or equal to 25 m. Next, find (approximate) solutions and bounds on the optimal number of sensors for both datasets by answering the two questions below. We recommend that you analyze first the small dataset.

   5.1 Solve problem (SDP) to find $(x^\star, X^\star)$ and an upper bound on problem (IP). A skeleton of the code you will have to implement is provided in the Python file `p5q.py`.

   5.2 Apply the following rounding heuristic to transform $x^\star$ to a feasible solution for problem (IP), which in turn provides a lower bound on (IP).

   - Initialize $L = \emptyset$ (locations with sensors) and $V = \{1, \ldots, n\}$ (locations without sensors)
   - **for** $i = 1, \ldots, n$
     - Select any $i^\star \in \operatorname{argmax}_{i\in V} x_i^\star$.
     - If $(i^\star, j) \in E$ for at least one $j \in L$, then set $V \leftarrow V \backslash \{i^\star\}$; otherwise set $L \cup \{i^\star\} \leftarrow L$ and $V \leftarrow V \setminus \{i^\star\}$
   - **end for**
   - **return** L.

   Use the function `integer_solver.py` to solve the integer program (IP) exactly. This function takes the adjacency matrix $A \in \mathbb{R}^{n\times n}$ associated with E as an input and returns a globally optimal binary solution together with the optimal value of (IP). The adjacency matrix A is constructed as follows. The element A_{ij} is set to 1 if $(i,j) \in E$ and to 0 otherwise. We point out that the solution of the NP-hard integer program (IP) can take a long time. Plot both the exact as well as the approximate solutions. Compare the solutions and the runtimes.