Computational Optimization - MSc AIDA UoM

Professor Nikolaos Samaras

Academic Year: 2021-2022

The objectives of the presolving procedure are as follows:

1. Reduce the dimensions of the linear problem.
2. Improve the numerical properties and computational characteristics of linear problems.
3. Detect if the linear problem is infeasible or unbounded.
4. Highlight certain properties of the linear problem that were not apparent during its modeling.

Presolving procedures can be repeated until no criterion finds application or for a specific time period that is desirable. Also, preprocessing procedures can be applied by any solver regardless of the algorithm it uses. Identifying all redundant rows and columns of a linear problem is a computationally expensive process. The execution order of the preprocessing procedures plays a significant role.

Tasks:

[**A.**] Write code in Python that reads a file in matrix storage format according to the following form:

$$\begin{aligned} \min(\max) \quad & c^T x \\ \text{s.t.} \quad & Ax \otimes b \\ & x \geq 0 \end{aligned}$$

where $\otimes = \{\leq, =, \geq\}$, $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$.

- A: Dimensions $m \times n$. Matrix A stores the coefficients of the technological constraints.
- b: Dimensions $m \times 1$. Vector b stores the right-hand sides of the technological constraints.
- c: Dimensions $1 \times n$. Vector c stores the coefficients of the objective function.
- $Eqin$: Dimensions $m \times 1$. Vector $Eqin$ stores the types of constraints. If $Eqin(i) = -1$, then the i^{th} constraint is of type $\leq$. If $Eqin(i) = 1$, then it is of type $\geq$. If $Eqin(i) = 0$, then it is of type $=$.
- $MinMax$: Dimensions 1×1. This variable specifies the type of the problem. If $MinMax = -1$, then the problem is a minimization. If $MinMax = 1$, then it is a maximization.

[**B.**] Write code in Python that implements the preprocessing method Eliminate k-ton Equality Constraints, as presented in Lecture 3.

[**C.**] Run a small computational study like the one in slide of Lecture 03. You will run the metaprograms you have from the work of the first week for *k=1,2,3,4,5*.