Computational Optimization - MSc AIDA UoM

Professor Nikolaos Samaras

Academic Year: 2021-2022

The development of new algorithms for linear programming is always accompanied by their computational efficiency. Computational efficiency includes comparing different algorithms based on two criteria: the number of iterations and CPU time. Specific linear problems used internationally for benchmarking these comparisons in linear programming are called benchmarks. These benchmarks have a specific storage format known as MPS. This format was initially developed by IBM for the electronic storage of linear and integer problems. It is also known as the industrial standard because it has been adopted by all commercial solvers.

Tasks:

[**A.**] Write Python code that reads a file in MPS storage format and converts it 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 $\otimes = \{\leq, =, \geq\}$, $c, x \in R^n$, $b \in R^m$ and $A \in R^{m \times n}$. Specifically:

- 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.

- R: Dimensions (number of ranges in the MPS)x4. This matrix stores the ranges of the constraints as follows:
  - 1st column: constraint name
  - 2nd column: The RHS of the constraint
  - 3rd column: RHS+—range— 1, if range ¿0; -1, if range ¡ 0; 0, if range = 0
- BS: Dimensions (number of bounds in the MPS)x3. This matrix stores the bounds of the constraints as follows:
  - 1st column: variable name

2^{nd} Column	**3^{rd} Column**
Type of Bound	Value of Bound
LO: Lower Bound	Value
UP: Upper Bound	Value
FX: Fixed	Value
FR: Free	None
MI: Minus Infinity	None
PL: Plus Infinity	None

[B.] Write code in Python that implements the conversion from matrix format to MPS storage format (the reverse of question A). Test your code on specified linear problems.

Name	**Constraints**	**Variables**	**Nonzeros**	**Bounds**	**Optimal Value**	**Optimal Value**
aircraft	3.754	7.517	20.267	B	1,57E+03	1567,042349
deter0	1.923	5.468	11.173	B	-2,05E+00	-2,045920
deter1	5.527	15.737	32.187	B	-2,56E+00	-2,557564
sc205-2r-8	189	190	510		-6,04E+01	-60,42961
sc205-2r-50	1.113	1.114	3.030		-3,08E+01	-30,764114
scagr7-2b-64	9.743	10.260	32.298		-8,33E+05	-832900
Lp01	6	8	48			
LP02	7	10	70			