A Computational Study of PDHG Algorithm for LP

Important Note: This is an individual project that you must complete independently. You cannot collaborate with anyone inside or outside of the class. You may have discussions/questions with the instructor or the TA, but no one else. Your code must be written entirely by yourself. Any plagiarism found will automatically lead to the disqualification from the class.

Assignment:

Implement a first-order method for solving the following standard LP pair:

$$\min c^T x, \text{ s.t. } Ax = b,\ x \geq 0, \qquad \max b^T y, \text{ s.t. } A^T y \leq c.$$

where b and c are nonzero vectors. Your Matlab function should have the interface

```
function [x,y,iter,Out] = my_pdhg(A,b,c,tol,maxit,...)
```

where

```
% INPUT:
%               A = constraint coefficient matrix
%               b = constraint right-hand side vector
%               c = objective coefficient vector
%             tol = tolerance
%           maxit = maximum number of iterations allowed
%             prt = indicator for printing iteration information
% OUTPUT:
%               x = computed primal solution
%               y = computed dual solution
%            iter = iteration counter
%             Out = output structure with out.Hist to be the
%                   iteration history of max. relative error
```

Requirements:

1. The stopping criterion: either the `maxit` is reached or the `maximum relative error`

$$\max\left\{\frac{\|Ax-b\|}{\|b\|}, \frac{\|\max(0, A^T y - c)\|}{\|c\|}, \frac{|c^T x - b^T y|}{\max(10^{-8}, |b^T y|)}\right\} \leq \text{tol}.$$

2. When the indicator `prt` is on, then once every while (say, 10000 iterations), print out the iteration count and the above 3 quantities, i.e., the relative primal and dual residual norms and the relative duality gap, respectively, in the following format (the last line is when the maximum error $\leq$ tol = 1e-6 is met):

```
iter   1000:  [primal dual gap] = [9.69e-02 1.61e-03 5.13e-02]
iter  10000:  [primal dual gap] = [4.43e-04 8.63e-05 5.17e-04]
                                  o o o
iter 120000:  [primal dual gap] = [1.84e-06 7.79e-07 1.10e-06]
iter 122587:  [primal dual gap] = [9.98e-07 3.86e-07 9.32e-07]
```

The *basic* PDHG framework for LP is the following extremely simple algorithm:

$$\begin{aligned} x^{k+1} &= \text{proj}_{\mathbb{R}^n_+}(x^k + \eta A^T y^k - \eta c) \\ y^{k+1} &= y^k - \sigma A(2x^{k+1} - x^k) + \sigma b \end{aligned}$$

where η is the primal step-size and σ is the dual step-size. Please read the recent survey on the theory of PDHG, in general and for LP specifically, summarized in the preprint by Dr. Haohao Lu: https://arxiv.org/abs/2403.14535, which contains further references on this subject.

This project has a research-like flavor, though with a very limited scope. Your task is to find ways, from the literature or by yourself, to produce a PDHG code which is relatively efficient, as compared with the instructor's codes, on a small class of random LPs generated by the code `gen_randLP.m` (given as a handout along with some codes). We aim to solve the random LPs to a moderately high accuracy with tolerance 10^{-6}, and hopefully to outperform Matlab `linprog` on some large enough problems (even on CPUs).

We note that, without a theoretical backing, an efficient code for a small class of problems does not imply that it is efficient or even convergent in general. However, we may be able to obtain some useful hints from the observed algorithmic behaviors.

Handout Files: The handout codes include the following:

```
gen_randLP.m    random LP generator
base_pdhg.p     instructor's code (basic version)
yz_pdhg.p.      instructor's code (more advanced version)
test_rand.p     test function for random LPs
test_imgs.m     test function for image inpainting
quick_test.m    simple test script
```

The inputs to `test_rand` includes: `(p,Run,rpt)`, where `p` is a positive integer that controls problem size, `Run` is a 1×4 binary vector (e.g., `[1 0 1 0]`) that controls which codes to run out of 4 solvers: `linprog, yz_pdhg, base_pdhg` and `my_pdhg` (which will be your code), and binary `prt` controls output (the value 1 turns on the iteration information output).

Procedure and Submissions:

1. From the course website, download the handout test and solver codes. You can run `test_imgs`, `quick_test` and `test_rand` without or with your code `my_pdhg`.

2. When your code is ready for submission, run the script `test_final`, which will be distributed later, and collect all the outputs.

3. Write a typed report (no more than 2 pages) to describe points of importance about your algorithmic research and implementation, and to explain your code's performance and behavior (as connected to the techniques you implemented). State your reasoned assessment on the algorithm's potential for large-scale LPs.

4. Submit your code `my_pdhg.m`, the output from the required runs, and the project report (in PDF format) to the course website online.

Some Test Results on MacBook Pro:

```
====================================
Trial 1: (m,n) = (5000,50000)
====================================

  --- Matlab linprog ---
Elapsed time is 456.154136 seconds.
P_res: 6.95182201e-16
D_res: 0.00000000e+00
pdGap: 1.22921248e-12
P_obj: 4.97023597e+05
Number of iter: 21

  --- yz_pdhg ---
Elapsed time is 43.790653 seconds.
Number of restarts: 0
P_res: 9.79472933e-07
D_res: 9.27222836e-07
pdGap: 2.20738768e-08
P_obj: 4.97023609e+05
Number of iter: 17783


====================================
Trial 2: (m,n) = (5000,50000)
====================================

  --- Matlab linprog ---
Elapsed time is 320.283985 seconds.
P_res: 6.77228847e-16
D_res: 0.00000000e+00
pdGap: 2.42153618e-12
P_obj: 6.33291813e+05
Number of iter: 20

  --- yz_pdhg ---
Elapsed time is 41.960223 seconds.
Number of restarts: 0
P_res: 9.53949446e-07
D_res: 9.98145684e-07
pdGap: 1.52951408e-07
P_obj: 6.33291811e+05
Number of iter: 16294
```

```
====================================
Trial 3: (m,n) = (5000,50000)
====================================

  --- Matlab linprog ---
Elapsed time is 373.259954 seconds.
P_res: 6.94351286e-16
D_res: 0.00000000e+00
pdGap: 1.00041765e-10
P_obj: 6.42092947e+05
Number of iter: 20

  --- yz_pdhg ---
Elapsed time is 47.123114 seconds.
Number of restarts: 0
P_res: 9.95286645e-07
D_res: 9.66335189e-07
pdGap: 6.69243740e-08
P_obj: 6.42092942e+05
Number of iter: 19228


======= (m,n) = (5000,50000) =======
Average of 3 runs / 2 Solvers used

--------------------------------
Solver:     linprog     yz_pdhg
--------------------------------
 Iter:           20       17768
 Resi:     3.46e-11    9.91e-07
 Time:     3.83e+02    4.43e+01
--------------------------------




Hardware Overview:

  Model Name: MacBook Pro
  Model Identifier: MacBookPro18,4
  Chip: Apple M1 Max
  Total Number of Cores: 10 (8 performance and 2 efficiency)
  Memory: 64 GB
```