Analysis of PDLP : A First Order Method For Solving Linear Programming Problems

Corentin TISSIER

Supervised by Nicolas BOUMAL

EPFL

Introduction

In recent years, first order methods have become popular and effective optimization techniques that can be applied to a wide range of problems in different fields. In particular, PDHG (Primal Dual Hybrid Gradient) also known as Chambolle Pock method gained significant attention and was studied in [3, 4].

The fundamental idea behind PDHG lies in the reformulation of an optimization problem as a saddle point problem involving both primal and dual variables. By exploiting the duality gap and utilizing a subgradient-based iterative scheme, PDHG algorithms are able to minimize the primal and maximize the dual simultaneously, thus reaching optimal solutions efficiently in various applications.

PDLP (Primal Dual LP) is an improved version of PDHG (Primal Dual Hybrid Gradient) with enhancements specifics to Linear Programming (LP) [1]. The main PDLP enhancements, namely adaptive restart and adaptive stepsize update, enable the algorithm to significantly improve the speed of the baseline PDHG algorithm for LP problems.

LP problem formulation

Primal formulation

LP problems are constrained optimisation problems of the form:

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & c^\intercal x \\ \text{subject to} \quad & Gx \geq h \\ & Ax = b \\ & l \leq x \leq u \end{aligned} \tag{1}$$

where $G \in \mathbb{R}^{m_1\times n}$, $A \in \mathbb{R}^{m_2\times n}$, $c \in \mathbb{R}^n$, $h \in \mathbb{R}^{m_1}$, $b \in \mathbb{R}^{m_2}$, $l \in (\mathbb{R} \cup \{-\infty\})^n$ and $u \in (\mathbb{R} \cup \{+\infty\})^n$.

Saddle point formulation

This primal formulation can be combined with its dual into a single equivalent saddle point optimization problem with simple feasible set region:

$$\min_{x\in X}\max_{y\in Y} \mathcal{L}(x,y) = \min_{x\in X}\max_{y\in Y} c^\intercal x - y^\intercal Kx + q^\intercal y \tag{2}$$

with $X := \{x \in \mathbb{R}^n : l \leq x \leq u\}$ and $Y := \{y \in \mathbb{R}^{m_1+m_2} : y_{1:m_1} \geq 0\}$, $K^\intercal := (G^\intercal, A^\intercal)$ and $q^\intercal := (h^\intercal, b^\intercal)$. This formulation enables us to use PDHG to solve the LP problem (1).

PDHG heuristic

Using subgradient and proximal operator theory, one can prove that a solution of a saddle point problem must satisfy in the LP case (2):

$$\begin{cases} x^* = \mathbf{proj}_X(x^* - (c - K^\intercal y^*)) \\ y^* = \mathbf{proj}_Y(y^* + (q - Kx^*)). \end{cases}$$

This motivates the following vanilla PDHG algorithm.

Vanilla PDHG algorithm for LP

Require: $x_0 \in \mathbb{R}^n, y_0 \in \mathbb{R}^{m_1+m_2}, \sigma_k, \tau_k > 0$

while *Not Converged* **do**

$\quad \hat{x}_{k+1} = x_k + \tau_k K^\intercal y_k;$

$\quad x_{k+1} = \mathbf{proj}_X(\hat{x}_{k+1} - \tau_k c) = \min\big(\max\big(x_k - \tau_k(c - K^\intercal y_k), l\big), u\big);$

$\quad \hat{y}_{k+1} = y_k - \sigma_k K(2x_{k+1} - x_k);$

$\quad y_{k+1} = \mathbf{proj}_Y(\hat{y}_{k+1} + \sigma_k q) = \max\big(y_k + \sigma_k(q - K(2x_{k+1} - x_k)), 0\big);$

end while

where the min and the max are taken entrywise. This algorithm is known to converge when we use constant step sizes $\tau\sigma\|K\|^2 < 1$ with $\|K\|$ the operator norm of the matrix K.

Restarted PDHG

Consider the very simple saddle point problem:

$$\min_{x\in\mathbb{R}}\max_{y\in\mathbb{R}} \mathcal{L}(x,y) = \min_{x\in\mathbb{R}}\max_{y\in\mathbb{R}} ax + xy + by. \tag{3}$$

It admits the unique solution $(x^*, y^*) = (b, a)$. When we apply PDHG to this problem, one can see on Figure 1 that the last iterates makes steady but slow progress. On the other end the average iterates approach very fast the solution at first but then have slow progress. We can clearly see that both the last iterate and the average iterate converge much faster with the restart.

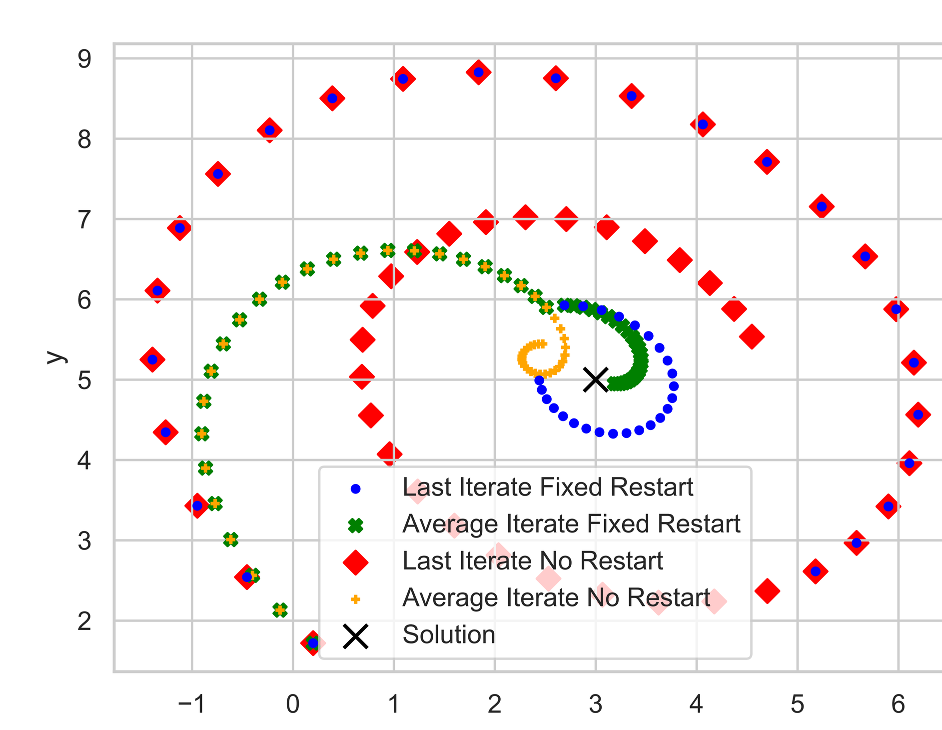

Figure 1. Plot of the first 50 iterates of non-restarted and restarted PDHG on problem (3) with $a = 5, b = 3$, step sizes $\sigma = \tau = 0.2$ and restart length 25 (if restart).

However, using fixed restart length doesn't guaranty better convergence properties and it remains to find a restart scheme adapted to LP problems.

Adaptive Restart

PDLP uses a restart scheme based on the normalised duality gap [2]:

$$\rho_r(z) := \frac{\max_{\{\hat{z}\in X\times Y \mid \|z-\hat{z}\|\leq r\}} \{\mathcal{L}(x,\hat{y}) - \mathcal{L}(\hat{x},y)\}}{r}.$$

As in the case of LP problems the normalised duality gap is always positive and is 0 if and only if evaluated at a solution, this is a good metric for optimality which enables to trigger restart only when significant progress can be made using it.

Adaptive stepsizes

To have a better control of the scaling between the primal and dual iterates, in PDLP the stepsizes are reparameterized by:

$$\tau = \eta/\omega \quad \text{and} \quad \sigma = \omega\eta \quad \text{with } \eta \in (0,\infty) \quad \text{and} \quad \omega \in (0,\infty).$$

With this parameterization, PDHG converges if the *step size* η is such that $\eta\|K\| < 1$. This condition is relaxed in PDLP where we ensure by backtracking at each PDHG steps the following bound on the step size:

$$\eta \leq \frac{\|z_{k+1} - z_k\|^2}{2(y_{k+1} - y_k)^\intercal K(x_{k+1} - x_k)}.$$

The primal weight ω_n is used to monitor the scaling between the primal and dual iterates. It is updated only after each restart:

$$\omega_n = \exp\left(0.5\log\left(\frac{\|y_{n,0} - y_{n-1,0}\|_2}{\|x_{n,0} - x_{n-1,0}\|_2}\right) + 0.5\log(\omega_{n-1})\right).$$

This update scheme enables the distances to optimality in the primal and dual to converge to 0 at a similar pace.

Numerical experiments

The results of the numerical experiments produced on the LP problems described in Table 1 are stated in Table 2. We can see that the fixed restart scheme doesn't necessarily improve the performance of PDHG as illustrated with ran10x10b. On the other hand, PDLP improves the convergence speed over PDHG significantly in every of these test instances which confirms the potential of these enhancements. We finally note that only adaptive restart with no stepsizes update is not always enough to improve the performance of PDHG as illustrated once again with ran10x10b.

Problem name	n	m_1	m_2
bk4x3	24	12	7
ran10x10b	200	100	20
ran17x17	578	289	34

Table 1. Test problem sizes.

Problem name	PDHG	Fixed Restart	Adaptive Restart	PDLP
bk4x3	17600	7100	5500	3200
ran10x10b	42200	81000	42700	6100
ran17x17	21100	15300	12100	10000

Table 2. Number of iterations needed before convergence for different PDHG enhancements.

References

[1] David Applegate, Mateo Díaz, Oliver Hinder, Haihao Lu, Miles Lubin, Brendan O'Donoghue, and Warren Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. *Advances in Neural Information Processing Systems*, 34:20243–20257, 2021.

[2] David Applegate, Oliver Hinder, Haihao Lu, and Miles Lubin. Faster first-order primal-dual methods for linear programming using restarts and sharpness. *Mathematical Programming*, pages 1–52, 2022.

[3] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40:120–145, 2011.

[4] Tom Goldstein, Min Li, Xiaoming Yuan, Ernie Esser, and Richard Baraniuk. Adaptive primal-dual hybrid gradient methods for saddle-point problems. *arXiv preprint arXiv:1305.0546*, 2013.