First Passage Time in Geometric Resetting of First-Order Methods in Linear Programming

Anonymous authors
Paper under double-blind review

Abstract

Restarts are one of several ingredients of first-order methods for large linear programmes, such as the primal-dual hybrid gradient method (PDHG) that underlies PDLP, and the one with the most developed theory. We study the expected number of iterations such a method needs when it is restarted at random: at each iteration, with some probability, the iterate is replaced by a draw from a restart law. On an idealised family of instances with a plateau and a wall, randomised restarting beats PDLP by a factor that grows like the square root of the dimension. Once the method is placed in the framework of first passage under restart, the discrete-time renewal formula of Bonomo and Pal gives the expected iteration count under a constant restart probability, and its derivative at zero gives a dispersion criterion for infinitesimal rates. We add a representation of the gain at every rate through the mean residual life of the restart-free count: restarting can never help when that count is new better than used in expectation, and can never hurt when it is new worse than used. For base methods whose iteration count obeys a two-sided linear-rate bound, geometric restarting can improve the expected count only by a factor tending to one as the tolerance decreases. In a reduced-state control model the optimal state-dependent policy is a two-level threshold rule which never restarts in a neighbourhood of the target. On 68 LP relaxations of MIPLIB 2017 models, a state-dependent stochastic rule in place of the adaptive restart of PDLP has a geometric-mean iteration ratio of 1.074 on the 66 instances on which both methods terminate, needs fewer iterations on 27 of them, and needs one twentieth of the iterations on the instance on which PDLP is slowest.

1 Introduction

Linear programming on a large scale arises throughout operations research, machine learning and combinatorial optimisation. For three decades after Dantzig's formulation the simplex method had the field to itself (Dantzig et al., 1955; Fearnley & Savani, 2015); over the following four, interior-point methods rose to prominence (Karmarkar, 1984; Gill et al., 1986; Andersen et al., 1996; Wright, 1997; Gondzio, 2012). In the past five years PDLP (Applegate et al., 2021; 2026) and related first-order solvers (Lu & Yang, 2025a; Lin et al., 2025; Chen et al., 2026) have become competitive on instances whose factorisation cannot be held in memory, or for which a moderate accuracy suffices. PDLP combines the primal-dual hybrid gradient (PDHG) iteration with presolve, diagonal preconditioning, adaptive step sizes, a primal weight and a restart scheme. Of these ingredients the restart has the most developed theory: Applegate et al. (2022) prove that, under a sharpness condition, restarted PDHG converges linearly, whereas the guarantees for the unrestarted method are sublinear. Restarting too often discards the progress accumulated since the last restart; restarting too seldom leaves the iteration in a slow regime.

A question of the same shape has been studied in statistical physics under the name of stochastic resetting (Evans & Majumdar, 2011b;a; 2014; Redner, 2001; Evans et al., 2020). A diffusing particle seeks a target and is returned to its origin at random times. Evans & Majumdar (2011b) showed that the return can shorten the mean time to reach the target, and Reuveni (2016) and Pal & Reuveni (2017) gave the criterion: an infinitesimal rate of resetting helps if and only if the coefficient of variation of the reset-free first-passage time exceeds one. An iterative optimisation method is, in outline, a process of the same kind: it has a target, the set of points at which the termination test is satisfied, and it may be sent to a fresh point at random times. Between restarts, however, its motion

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

is deterministic, and its only randomness is the restart itself. This paper asks when randomised restarting reduces the expected iteration count of a first-order method for linear programming, and how the restart probability should depend on the state of the iteration. Section 1.1 gives an idealised family on which randomised restarting wins by a margin that grows with the dimension; it motivates what follows. Our contributions are the following.

1. *A renewal model of restarted first-order methods.* We cast a first-order method with randomised restarts as a first passage under restart. The discrete-time renewal formula and dispersion criterion of Bonomo & Pal (2021) then give the expected iteration count under a constant restart probability (Theorem 2, Corollary 2).
2. *A global analysis.* A mean-residual-life representation of $T_p - T_0$ (Theorem 3), valid for every p, yields a dichotomy (Corollary 3): restarting at any rate cannot help if the restart-free count is new better than used in expectation, and cannot hurt if it is new worse than used. For base methods with a two-sided linear rate, the best speed-up available from geometric restarting tends to one as the tolerance decreases, and small rates strictly hurt (Proposition 1, Corollary 4).
3. *An optimal state-dependent policy.* In a reduced-state control model with a stochastic-monotonicity property, verified for a diffusion surrogate though not for the projected PDHG map, the optimal policy is a two-level threshold rule (Theorem 4) which, under a further condition, never restarts on a neighbourhood of the target (Corollary 5).
4. *Experiments.* A state-dependent stochastic rule, in place of the adaptive restart of PDLP, is close to neutral in iteration count on 66 instances, with one large gain on the instance on which PDLP is slowest. On a single instance, the renewal formula predicts correctly the sign of the effect of a constant restart rate (Section 7).

1.1 A MOTIVATING EXAMPLE: A PLATEAU AND A WALL

The following idealised family shows that randomised restarting can beat PDLP by a factor that grows with the dimension, and it shows what such a gain requires. Take a variable $w \in [0,1]$ that enters no constraint and has cost $-\delta$, so that the optimum has $w = 1$, together with a block $\min\{\hat{c}^\top \hat{x} : \ \hat{K}\hat{x} \geq \hat{h},\ \hat{l} \leq \hat{x} \leq \hat{u}\}$ whose constraint matrix has $\|\hat{K}\|_2 = \sqrt{n}$. Run PDHG with $\tau = \sigma$ and $\tau\|\hat{K}\|_2 < 1$, as the step-size condition requires. Since w enters no constraint, one step moves it to $\min\{1,\ w + \tau\delta\}$: w crosses a plateau at speed $\tau\delta$ and stops at the wall $w = 1$. Take the target to be $w = 1$ together with a target for the block.

Theorem 1 (Plateau and wall)**.** *Suppose that from every starting point the block reaches its target within $f \geq 2$ PDHG iterations, whatever w does, and let $L := \lceil 1/(\tau\delta) \rceil$, so that $L \geq \sqrt{n}/\delta$.*

(i) Any method whose iterates are PDHG steps or convex combinations of earlier iterates needs at least L iterations from $w = 0$. This covers PDHG with the restarts of PDLP, which go to averages of past iterates, run with this step size from the origin, and the rule of Section 6, whose restarts go to the segment between the current iterate and the average.

(ii) Let a restart draw $w \in \{0, 1\}$ with equal probabilities and the block from any fixed law. Geometric restarting at rate $p = 1/f$ reaches the target in at most $2\mathrm{e}f$ iterations in expectation.

The proof is in Appendix B. The gain comes from a restart law that places mass on the wall itself, so that a restarted run skips the plateau with probability one half. Restarts confined to the convex hull of the trajectory cannot skip it, which is the content of part (i), and the family lies outside the hypotheses of Corollary 4 below, because the plateau adds L iterations to the slow half of the restart points. To exploit the wall, sPDLP needs one change: a fourth candidate destination that moves coordinates to the bounds their reduced costs point to (Section 6). With it, sPDLP needs about 150 iterations at every dimension, whereas PDHG with restarts to the average needs thousands, growing with n (Figure 1). The example is idealised: presolve would remove a variable that enters no constraint, diagonal preconditioning can reduce the factor $\sqrt{n}$, and the plateau matters only while δ is large compared with the tolerance. Appendix C extends it to many walls, where restarting only the unsettled coordinates beats restarting the whole iterate by a factor exponential in their number, and relates it to vanishing and exploding gradients and to linear programmes for training ReLU networks.

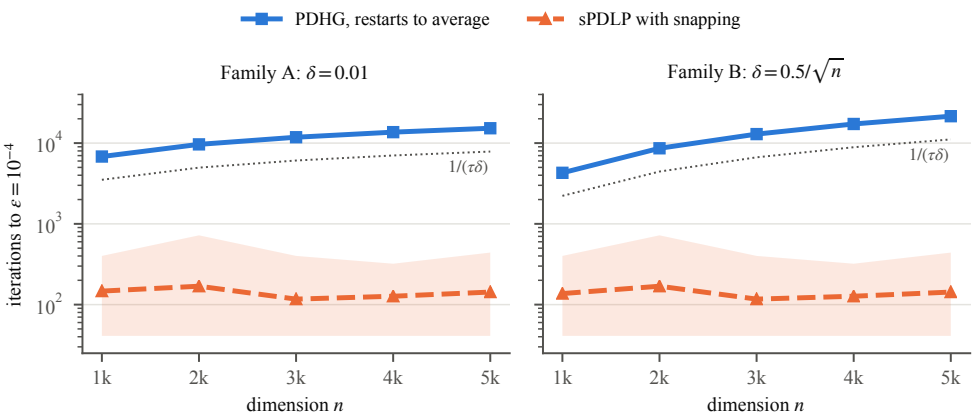


Figure 1: Iterations to reach $\varepsilon = 10^{-4}$ on plateau-and-wall instances with n from 1000 to 5000, in a reference implementation of PDHG with fixed step sizes and no preconditioning. Restarts to the running average, the destination PDLP uses, here every 64 iterations, roughly double the plateau length $1/(\tau\delta)$ (dotted). sPDLP with snapping (Section 6) needs between 117 and 169 iterations on average over 20 seeds at every n; the band shows the range over seeds. In family B, δ shrinks like $1/\sqrt{n}$, so that the plateau also grows under diagonal preconditioning.

2 Related work

Applegate et al. (2022) gave the first rigorous analysis of restarted PDHG for linear programming, proving linear convergence under sharpness for several restart schemes; Applegate et al. (2026) document the engineering, and Applegate et al. (2024) extend the analysis to infeasibility detection. Later work refines the rate through the geometry of the problem (Lu & Yang, 2024a; Xiong & Freund, 2024), gives guarantees in terms of limiting error ratios and of the uniqueness of the optimiser (Xiong & Freund, 2026; Xiong, 2026), accelerates by restarting a Halpern iteration (Lu & Yang, 2024b), and carries the method to quadratic programming (Lu & Yang, 2025b) and to GPUs (Lu & Yang, 2025a; Lin et al., 2025; Chen et al., 2026). For accelerated gradient methods, adaptive restarts are due to O'Donoghue & Candès (2015); Roulet & d'Aspremont (2020), Fercoq & Qu (2019), Renegar & Grimmer (2022) and Adcock et al. (2025) give schedules that are near-optimal under sharpness or quadratic growth. In all of these the restart is a deterministic function of the trajectory. That randomised restarts shorten heavy-tailed running times is classical in combinatorial search: Luby et al. (1993) give optimal universal schedules for Las Vegas algorithms, and Gomes et al. (1998) and Gomes et al. (2000) documented heavy-tailed running times of backtrack search for satisfiability and constraint satisfaction and showed that randomised restarts remove them. The bimodal example of Remark 1 is of this type.

Evans & Majumdar (2011b;a; 2014) introduced diffusion with stochastic resetting; Evans et al. (2020) survey the field. Reuveni (2016) and Pal & Reuveni (2017) gave the criterion for infinitesimal rates, Pal & Prasad (2019) analyse the transitions of the optimal rate, Pal et al. (2022) read the criterion through the inspection paradox, and Eliazar & Reuveni (2020) analyse sharp restart in continuous time through the residual lifetime and the hazard rate of the completion time; see also Ahmad et al. (2019); Aspman et al. (2026). Bonomo & Pal (2021) set up first passage under restart in discrete time for a generic process whose memory is erased at each restart, derive the mean completion time under geometric restart and the criterion $\mathrm{CV}^2 > 1 - 1/\mathbb{E}[N]$, and apply them to random walks on a one-dimensional lattice. Theorem 2 and Corollary 2 below are their eq. (20) and criterion (25), up to the convention for the timing of a restart; we give self-contained proofs because the later results are built from the same decomposition. While often seen as restricted to univariate settings, the renewal framework of Pal & Reuveni (2017) and Bonomo & Pal (2021) is not confined to one dimension, since it depends only on the law of the scalar reset-free time. What the resetting literature has in common is its object: a stochastic process, typically Brownian motion or a lattice walk in one dimension or in a few (Evans & Majumdar, 2014), returned to a fixed point or to a prescribed law. Our setting differs in three respects. The motion between restarts is a deterministic iteration in dimension n plus the number of constraints, so that all randomness of the restart-free count is

induced by the restart destination. The restart law is chosen by the algorithm and depends on the trajectory, which is why the renewal assumption is an idealisation here. And the quantities that matter are tied to the convergence rate of the base method. To our knowledge, Theorem 1, the representation of Theorem 3 at every rate and its dichotomy for geometric restart in discrete time, Proposition 1 and Corollary 4, and the treatment of state-dependent restarting of an optimisation method as a monotone dynamic programme in Section 5, whose argument is itself standard (Puterman, 1994), are new.

3 Setting

We consider the linear programme

$$\min_{x\in\mathbb{R}^n} c^\top x \quad \text{s.t. } Gx \geq h,\ Ax = b,\ l \leq x \leq u, \tag{1}$$

which, with $K^\top = (G^\top, A^\top)$ and $q^\top = (h^\top, b^\top)$, is the saddle-point problem $\min_{x\in X}\max_{y\in Y}\ c^\top x - y^\top Kx + q^\top y$ over $X = \{x : l \leq x \leq u\}$ and $Y = \{y = (y_G, y_A) : y_G \geq 0\}$. PDLP applies the alternating projected steps

$$x^{k+1} = \mathrm{proj}_X\big(x^k - \tau(c - K^\top y^k)\big), \qquad y^{k+1} = \mathrm{proj}_Y\big(y^k + \sigma(q - K(2x^{k+1} - x^k))\big), \tag{2}$$

with step sizes $\tau, \sigma > 0$, adapted during the run, to the iterates $z^k = (x^k, y^k)$, and restarts to their running average $\bar{z}^k$. Progress is monitored, as in Applegate et al. (2021), through the convergence score

$$s^k := r_{\mathrm{p}}^k + r_{\mathrm{d}}^k + g^k, \tag{3}$$

the sum of the relative primal residual, the relative dual residual and the relative gap, defined in Appendix A. We write $s(z)$ for the score of an arbitrary point z; the solver stops when $s^k \leq \varepsilon_{\mathrm{opt}}$.

Let Ψ denote one iteration of a deterministic base method. We model the restarted trajectory as

$$z^{k+1} = \begin{cases} \Psi(z^k), & \text{with probability } 1 - p_k, \\ \tilde{z}^{k+1} \sim \mu_k(\cdot \mid z^k), & \text{with probability } p_k, \end{cases} \tag{4}$$

where $p_k \in [0, 1]$ is the restart probability and μ_k a restart law. The analysis applies to any deterministic base method. In the rule of Section 6 the stochastic restart replaces the adaptive restart of PDLP, so there Ψ is the PDLP iteration with its own restart switched off. To obtain a tractable one-dimensional state we project z^k onto the *progress variable*

$$R_k := \|x^{k+1} - x^k\|_2 + \|y^{k+1} - y^k\|_2, \tag{5}$$

a by-product of every step, which tends to zero as the iterates converge. The target is $\mathcal{T}_\varepsilon = \{R \leq \varepsilon\}$, and the mean first passage time from R is $T(R) = \mathbb{E}[\inf\{k : R_k \leq \varepsilon\} \mid R_0 = R]$. A diffusion surrogate for R_k, used only to exhibit a model in which the monotonicity hypothesis of Section 5 holds, is described in Appendix F.

4 Geometric restarting: an exact analysis

4.1 Standing assumptions

In this section the restart probability is a constant, $p_k \equiv p$, and the restart law is fixed. Fix a target set $\mathcal{T}$; the reader may take $\mathcal{T}_\varepsilon$ or the termination test $\{s \leq \varepsilon_{\mathrm{opt}}\}$, as nothing below depends on the choice. Write μ for the law of the point produced by a restart.

Assumption 1 (Restart-free iteration count)**.** Let N be the number of iterations the base method needs to reach $\mathcal{T}$ when started from a point drawn from μ and left to run without further restarts. We assume that N takes values in $\{1, 2, \dots\} \cup \{\infty\}$, that $\mathbb{P}[N < \infty] > 0$, and that successive restart epochs produce independent copies $N_1, N_2, \dots$ of N, independent also of the restart coins.

Since the base map is deterministic, all the randomness in N comes from the restart: $N = N(\tilde{z})$ with $\tilde{z} \sim \mu$. The value $N = \infty$ covers stalling. Independence holds when each restart draws afresh from a fixed law, as in Theorem 1; for the trajectory-dependent restart action of Section 6 it fails, and the results describe a renewal idealisation whose error we do not quantify. A restart is not charged as an iteration; Remark 3 in the appendix gives the constants under the opposite convention.

4.2 THE RENEWAL FORMULA

Theorem 2 (Mean first passage time under geometric restarting; Bonomo & Pal, 2021)**.** *Let Assumption 1 hold, let $p \in (0,1)$, and let T_p be the expected number of iterations to reach $\mathcal{T}$ when the restart is applied, independently at each iteration, with probability p, starting from a fresh draw from μ. Write*

$$\phi(p) \;:=\; \mathbb{E}\big[(1-p)^N\big] \;=\; \sum_{t\geq 1} \mathbb{P}[N=t]\,(1-p)^t \qquad (\textit{with } (1-p)^\infty := 0).$$

Then $\phi(p) \in (0,1)$ and

$$T_p \;=\; \frac{(1-p)\big(1-\phi(p)\big)}{p\,\phi(p)} \;<\; \infty. \tag{6}$$

Moreover $T_p \to \mathbb{E}[N]$ as $p \downarrow 0$, where $\mathbb{E}[N] = \infty$ is permitted on both sides.

The proof (Appendix D.1) decomposes the run into epochs. Let Γ be the geometric index of the first restart. An epoch ends in absorption if and only if $\Gamma \geq N$, an event of probability $\mathbb{E}[(1-p)^{N-1}] = \phi(p)/(1-p)$, and it consumes $\min(N,\Gamma)$ iterations in either case; Wald's identity then gives $T_p = \mathbb{E}[\min(N,\Gamma)]/\mathbb{P}[\Gamma \geq N]$.

Corollary 1 (Restarting removes stalling)**.** *If $\mathbb{P}[N=\infty] > 0$ then $T_0 = \mathbb{E}[N] = \infty$, whereas $T_p < \infty$ for every $p \in (0,1)$, since $\phi(p) > 0$.*

This is the discrete analogue of the observation of Evans & Majumdar (2011a) that resetting renders an infinite mean first passage time finite. It asserts only finiteness: by Proposition 1, T_p grows exponentially in the number of iterations a concentrated N requires.

Corollary 2 (Dispersion criterion; Bonomo & Pal, 2021)**.** *Suppose in addition that $\mathbb{P}[N<\infty] = 1$ and $\mathbb{E}[N^2] < \infty$. Then $p \mapsto T_p$ is right-differentiable at 0, with*

$$\left.\frac{dT_p}{dp}\right|_{p=0^+} \;=\; \tfrac{1}{2}\Big(\mathbb{E}[N]^2 - \mathbb{E}[N] - \mathrm{Var}(N)\Big), \tag{7}$$

so that, writing $\mathrm{CV}(N) := \sqrt{\mathrm{Var}(N)}/\mathbb{E}[N]$,

$$\exists\, p>0 :\; T_p < T_0 \qquad \Longleftrightarrow \qquad \mathrm{CV}(N)^2 > 1 - \frac{1}{\mathbb{E}[N]}. \tag{8}$$

If the inequality in (8) *is reversed and strict, then $T_p > T_0$ for all sufficiently small $p > 0$.*

The proof is in Appendix D.4. As $\mathbb{E}[N] \to \infty$ the criterion becomes the $\mathrm{CV} > 1$ criterion of Reuveni (2016) and Pal & Reuveni (2017); the correction $-1/\mathbb{E}[N]$ reflects the granularity of the iteration. A deterministic count $N \equiv t$ gives $T_p = T_p^{\det}(t) := \sum_{k=0}^{t-1}(1-p)^{-k}$, increasing in p, and a geometric count gives $T_p = \mathbb{E}[N]$ for every p (Remark 2).

Remark 1 (The criterion is sufficient only). Fix an integer $t > 1$ and let $N = 1$ or $N = t$, each with probability $\frac{1}{2}$. Then $\mathbb{E}[N] = \frac{1+t}{2}$ and $\mathrm{Var}(N) = \frac{(t-1)^2}{4} \leq \mathbb{E}[N]^2 - \mathbb{E}[N]$, so criterion (8) fails and small restart probabilities hurt. Yet letting $p \uparrow 1$ in (6), with $\phi(p) \sim \frac{1}{2}(1-p)$, gives $T_p \to 2$, whereas $T_0 = \frac{1+t}{2}$. For $t = 101$, the derivative at zero is $+25$ while $\inf_p T_p = 2$. This is the mechanism of Las Vegas restart schedules (Luby et al., 1993), of Theorem 1, and in the resetting literature of the discontinuous transition of Pal & Prasad (2019): when the running time has a fast mode, one should re-draw aggressively until it is hit, and a derivative at $p = 0$ cannot see this.

4.3 THE SIGN OF THE GAIN AT EVERY RATE

For a given p, is T_p smaller or larger than T_0? The answer is governed by the *mean residual life* of N, $m(j) := \mathbb{E}[N - j \mid N > j]$ for integers $j \geq 0$ with $\mathbb{P}[N > j] > 0$, the expected number of further iterations needed once j have been spent without success; write $m := m(0) = \mathbb{E}[N]$. Restarting after j iterations exchanges $m(j)$ for m. The following identity says that $T_p - T_0$ is a geometrically weighted average of such exchanges.

Theorem 3 (Mean-residual-life representation)**.** *Let Assumption 1 hold with $\mathbb{P}[N<\infty] = 1$ and $m < \infty$. Then for every $p \in (0,1)$,*

$$T_p - T_0 \;=\; \frac{1-p}{\phi(p)} \sum_{j\geq 1} p(1-p)^{j-1}\,\mathbb{P}[N>j]\,\big(m - m(j)\big). \tag{9}$$

The proof (Appendix D.5) takes three lines from $T_p = \mathbb{E}[\min(N,\Gamma)]/\mathbb{P}[\Gamma \geq N]$ and $\mathbb{E}[N] = \mathbb{E}[\min(N,\Gamma)] + \mathbb{E}[(N-\Gamma)^+]$. Every weight in (9) is positive, so the sign of $T_p - T_0$ is decided by the signs of $m - m(j)$.

Corollary 3 (Global dichotomy)**.** *Under the hypotheses of Theorem 3:*

- *(i) if $m(j) \leq m$ for every $j \geq 1$ with $\mathbb{P}[N > j] > 0$, that is, if N is* new better than used in expectation *(NBUE), then $T_p \geq T_0$ for every $p \in (0,1)$, with strict inequality at every p unless N is geometric;*
- *(ii) if $m(j) \geq m$ for every such j (NWUE), then $T_p \leq T_0$ for every $p \in (0,1)$;*
- *(iii) if the discrete hazard $H(t) := \mathbb{P}[N = t \mid N \geq t]$ is non-decreasing in t (increasing failure rate, IFR), then N is NBUE and (i) applies.*

The proof is in Appendix D.5. Statement (iii) is the algorithmic reading: if a restart-free run that has gone on longer without terminating is thereby more likely to terminate at the next iteration, then no constant restart rate can help. It is the opposite of the heavy-tailed situation of Remark 1, in which a long run signals a slow mode that should be abandoned. Expanding (9) at $p = 0$ recovers (7). When the restart-free count is concentrated, bounds are available at every rate.

Proposition 1 (Bounded dispersion)**.** *Let Assumption 1 hold with $t_- \leq N \leq t_+$ almost surely, $1 \leq t_- \leq t_+ < \infty$. Then for every $p \in (0,1)$, with T_p^{det} as above,*

$$T_p^{\mathrm{det}}(t_-) \leq T_p \leq T_p^{\mathrm{det}}(t_+), \quad \textit{hence} \quad T_p \geq t_- \ \textit{and} \ \frac{T_0}{\inf_p T_p} \leq \frac{\mathbb{E}[N]}{t_-} \leq \frac{t_+}{t_-}. \tag{10}$$

If moreover $(t_+ - t_-)^2 < 4\,t_-(t_- - 1)$, then criterion (8) *fails strictly and $T_p > T_0$ for all sufficiently small $p > 0$.*

The proof (Appendix D.6) sandwiches $\min(N,\Gamma)$ and $\mathbb{P}[\Gamma \geq N]$ between their values at $t_\pm$ and uses Popoviciu's bound on the variance. The condition holds when the slowest restart point needs fewer than about three times the iterations of the fastest. Note also that $T_p^{\mathrm{det}}(t_-) \geq (1-p)^{-(t_- - 1)}$: at a fixed rate the expected count grows exponentially in t_-.

Corollary 4 (Linearly convergent base methods)**.** *Suppose there are constants $\gamma \in (0,1)$, $0 < \rho_- \leq \rho_+ < \infty$ and $t_0 \geq 0$ such that, for μ-almost every restart point, the restart-free count $N(\varepsilon)$ to tolerance ε satisfies*

$$t_-(\varepsilon) := \left\lceil \frac{\log(\rho_-/\varepsilon)}{\log(1/\gamma)} \right\rceil \ \leq \ N(\varepsilon) \ \leq \ \left\lceil \frac{\log(\rho_+/\varepsilon)}{\log(1/\gamma)} \right\rceil + t_0 =: t_+(\varepsilon) \qquad \textit{for all } \varepsilon \in (0, \rho_-). \tag{11}$$

Then, as $\varepsilon \downarrow 0$: (i) the best speed-up available from geometric restarting at any rate satisfies $T_0(\varepsilon)/\inf_p T_p(\varepsilon) \leq t_+(\varepsilon)/t_-(\varepsilon) \to 1$; (ii) there is $\varepsilon_0 > 0$ such that for every $\varepsilon < \varepsilon_0$, $T_p(\varepsilon) > T_0(\varepsilon)$ for all sufficiently small $p > 0$.

The proof is in Appendix D.6. An upper bound of the form in (11) is what the sharpness analysis of Applegate et al. (2022) provides for restarted PDHG, up to constants, with ρ the distance of the restart point from the optimal set; the lower bound is an assumption that the method converges no faster than linearly. The spread of log-distances of restart points then becomes a spread of iteration counts that is $O(1)$ while the counts themselves grow like $\log(1/\varepsilon)$. In the family of Theorem 1 the plateau makes t_0 grow with n, which is why the corollary does not apply there.

4.4 Using the formula

Theorem 2 turns the choice of a constant rate into a measurement of the whole curve. Draw B restart points $\tilde{z}^{(1)}, \ldots, \tilde{z}^{(B)} \sim \mu$, run the restart-free method from each under a fixed budget, record $N_1, \ldots, N_B$, and form the plug-in curve

$$\hat{\phi}(p) = \frac{1}{B}\sum_{i=1}^{B}(1-p)^{N_i}, \qquad \hat{T}_p = \frac{(1-p)\big(1-\hat{\phi}(p)\big)}{p\,\hat{\phi}(p)}. \tag{12}$$

Its minimiser over a grid of p estimates the best constant rate; runs that exhaust the budget contribute $N_i = \infty$, so pilot runs need not be completed. The same data give an empirical mean residual life, whose profile indicates by Corollary 3 whether any constant rate can help.

5 State-dependent restarting: the threshold policy

We now let the restart probability depend on the state. Let $\mathcal{S} \subseteq [0, \infty)$ be a Borel state space carrying the reduced state R, with the usual order; let $\mathcal{T} = \{R \le \varepsilon\}$; let $P(\cdot \mid R)$ be the transition kernel of one iteration of the base method in the reduced state, which presupposes a Markov approximation of R_k; and let μ be the law of the reduced state after a restart. A *policy* is a measurable map $\pi : \mathcal{S} \to \mathcal{A}$, where $\mathcal{A} \subseteq [0, 1]$ is a non-empty compact set of admissible restart probabilities, and the process evolves as (4) with $p_k = \pi(R_k)$. Write $T_\pi(R)$ for the expected number of iterations to reach $\mathcal{T}$ from R under π, and $T^\star := \inf_\pi T_\pi$. This is a dynamic programme with non-negative costs in which a restart is an action with transition kernel μ; its Bellman recursion lets a restart pre-empt an absorption, a convention that differs from that of Section 4.2 by a factor $1 - p$ and affects no statement below (Remark 4).

Assumption 2 (Regularity)**.** $\mathcal{A}$ is compact with $\max \mathcal{A} > 0$; the kernels P and μ are Borel; and some policy π has T_π bounded on $\mathcal{S}$.

The last clause holds under Assumption 1: for any $\bar{a} \in \mathcal{A} \cap (0, 1)$ the constant policy $\pi \equiv \bar{a}$ has, by Remark 4,

$$T_\pi \;\le\; \bar{T} \;:=\; \frac{1}{\bar{a}} + \frac{T_{\bar{a}}}{1 - \bar{a}} \;<\; \infty \tag{13}$$

uniformly on $\mathcal{S}$, with $T_{\bar{a}}$ finite by Theorem 2; hence $T^\star \le \bar{T}$. Since every one-stage cost equals 1, value iteration from the zero function increases to $T^\star$, $T^\star$ satisfies the Bellman equation, and a stationary policy attaining the minimum in it is optimal (Bertsekas & Shreve, 1978, Ch. 8–9).

Lemma 1 (Bellman equation and bang–bang structure)**.** *Under Assumption 2, $T^\star \equiv 0$ on $\mathcal{T}$ and, for $R \notin \mathcal{T}$,*

$$T^\star(R) \;=\; 1 + \min_{a \in \mathcal{A}} \Big[(1 - a)\, Q^\star(R) + a\, C^\star \Big], \qquad Q^\star(R) := \int T^\star \, dP(\cdot \mid R), \quad C^\star := \int T^\star \, d\mu. \tag{14}$$

The bracket is affine in a, so the action $\max \mathcal{A}$ is optimal where $Q^\star(R) > C^\star$, the action $\min \mathcal{A}$ where $Q^\star(R) < C^\star$, and every action where they are equal.

One should therefore restart aggressively exactly when the expected remaining time from where the iteration is heading exceeds that from a fresh restart. To conclude that the switching set is an interval we need the reduced dynamics to be monotone.

Assumption 3 (Stochastic monotonicity)**.** If $R \le R'$ then $P(\cdot \mid R)$ is stochastically dominated by $P(\cdot \mid R')$: $\int f \, dP(\cdot \mid R) \le \int f \, dP(\cdot \mid R')$ for every bounded non-decreasing f.

Lemma 3 in Appendix F verifies this for the Euler–Maruyama discretisation of the diffusion surrogate. It is not verified for the kernel induced on R_k by one projected PDHG step (2), whose projections onto a box and a cone are non-expansive but not obviously monotone once composed with the norm defining R_k; we know neither a proof nor a counterexample. The theorem below is thus exact for the surrogate, and only suggestive for the algorithm.

Theorem 4 (The two-level threshold policy is optimal)**.** *Let Assumptions 2 and 3 hold and let $\mathcal{T} = \{R \le \varepsilon\}$. Then:*

(i) $T^\star$ is non-decreasing on $\mathcal{S}$, and so is $Q^\star$;

(ii) the set $\{R : Q^\star(R) > C^\star\}$ is an up-set, of the form $(\beta^\star, \infty)$ or $[\beta^\star, \infty)$, where

$$\beta^\star := \inf\{R \in \mathcal{S} : Q^\star(R) > C^\star\} \in [0, \infty]; \tag{15}$$

(iii) consequently the stationary policy

$$\pi^\star(R) = \begin{cases} \max \mathcal{A}, & R > \beta^\star, \\ \min \mathcal{A}, & R \le \beta^\star, \end{cases} \tag{16}$$

is optimal.

In particular, $\mathcal{A} = [0, \bar{p}]$ gives an optimal policy that does not restart at all below the threshold.

Algorithm 1 sPDLP: the state-dependent stochastic restart rule, evaluated every 40 iterations

Require: iterates z^k, z^{k-1}; average $\bar{z}^k$; restart count ν_k; epoch length L_k; parameters $\kappa, d, L_0, p_{\max}, p_{\min}$

$s^k \leftarrow s(z^k)$ by (3)
if $s^k \le \kappa\,\varepsilon_{\text{opt}}$ **then return** no restart ▷ no-restart region
end if
$M_k \leftarrow \|x^k - x^{k-1}\|_2 + \|y^k - y^{k-1}\|_2$; compute p_k from (19)–(21)
draw $v \sim \text{Unif}(0,1)$; **if** $v > p_k$ **then return** no restart
draw $\theta \sim \text{Unif}(0,1)$; $z^{\text{dest}} \leftarrow \arg\min\{s(z^k),\, s(\bar{z}^k),\, s(\theta\bar{z}^k + (1-\theta)z^k)\}$
move to z^{dest}; reset the running average; $\nu_{k+1} \leftarrow \nu_k + 1$, $L_{k+1} \leftarrow 0$

The proofs of the lemma and the theorem are in Appendix E.

Corollary 5 (No restarting near convergence)**.** *If $0 \in \mathcal{A}$, then the optimal policy* (16) *has $\pi^\star \equiv 0$ on $\{R \le \beta^\star\}$. Suppose further that μ puts mass at least $\omega > 0$ on reduced states $R \ge \xi$, for some $\xi > \varepsilon$, and that*

$$\limsup_{R \downarrow \varepsilon} \mathbb{P}\big[R_1 > \varepsilon \mid R_0 = R\big] \;<\; \frac{\omega\, T^\star(\xi)}{\bar{T}}, \tag{17}$$

where $\bar{T}$ is the bound (13) *and $R_1 \sim P(\cdot \mid R)$. Then $\beta^\star > \varepsilon$: the no-restart region is non-degenerate.*

The proof is in Appendix E; Remark 7 explains why a condition such as (17) cannot be dropped, and Remarks 5 and 6 discuss the calibration of $\beta^\star$ and the case for randomisation.

6 A STATE-DEPENDENT STOCHASTIC RESTART RULE

The rule evaluated in Section 7, which we call sPDLP, is stated in Algorithm 1. It is evaluated every 40 iterations, at the termination check of PDLP, and it replaces the adaptive restart of PDLP; the step-size and primal-weight updates are unchanged. It uses the score s^k of (3) to suppress restarting near convergence, and the *merit* $M_k := \|x^k - x^{k-1}\|_2 + \|y^k - y^{k-1}\|_2 = R_{k-1}$, the progress variable of Section 3, to modulate the probability away from it, together with the *epoch length* L_k, the number of iterations since the last restart, and the *restart count* ν_k, the number of stochastic restarts fired so far. Motivated by Corollary 5, the rule sets the restart probability to zero in the region

$$s^k \le \kappa\,\varepsilon_{\text{opt}}, \qquad \kappa = 2.5, \tag{18}$$

where ε_{opt} is the termination tolerance. Outside this region the probability is

$$p_k = \max\big\{1 - \exp(-\eta\,\zeta_k),\, p_{\min}\big\}, \qquad \eta = -\log(1 - p_{\max}), \tag{19}$$

with the activation ζ_k built from the merit, the epoch length and the restart count:

$$\chi_k = 1 - e^{-M_k}, \qquad L_k^{\text{eff}} = L_0\sqrt{M_k}\; 2^{\,0.6\,\lfloor \nu_k/d \rfloor}, \tag{20}$$

$$\alpha_k = \frac{L_k}{L_k + L_k^{\text{eff}}}, \qquad \zeta_k = \alpha_k\big[\chi_k + (1-\chi_k)\alpha_k\big], \tag{21}$$

where $L_0 = 200$ and $d = 3$. Since $\alpha_k \in [0,1)$, one has $\alpha_k^2 \le \zeta_k \le \alpha_k < 1$ and so $p_k \in [p_{\min}, p_{\max})$. The patience factor α_k allows each epoch a window of order L_k^{eff} before a restart becomes likely, and the window lengthens by about one half for every d restarts taken. Only the no-restart region (18) is motivated by the theory. The dependence on L_k and ν_k makes the rest non-Markov in the reduced state, hence outside the control model of Section 5, and the floor $p_{\min} > 0$ departs from the zero rate of Corollary 5; this part of the rule is a heuristic.

When a restart fires, the destination is the lowest-scoring of the current iterate, the running average and a random interpolation between them,

$$z^{\text{dest}} = \arg\min\big\{s(z) :\; z \in \{z^k,\; \bar{z}^k,\; \theta\bar{z}^k + (1-\theta)z^k\}\big\}, \qquad \theta \sim \text{Unif}(0,1). \tag{22}$$

Since z^k is itself a candidate, a restart never increases the score; when z^k is chosen only the averaging window is reset. The residuals of the interpolated candidate follow by linearity from Kx^k, $K\bar{x}^k$, $K^\top y^k$ and $K^\top \bar{y}^k$, so the rule needs no product with K beyond those of the termination check.

Snapping. Every destination in (22) lies in the convex hull of the trajectory, so by Theorem 1(i) this action cannot cross a plateau. A minimal change removes the limitation. When a restart fires, form also the candidate $(\hat{x}, y^k)$, where $\hat{x}$ agrees with x^k except that each coordinate whose reduced cost $(c - K^\top y^k)_j$ is negative, respectively positive, is moved, independently with probability one half, to its upper, respectively lower, bound when that bound is finite. The candidate joins the minimisation in (22), so a restart still never increases the score; coordinates already at the bound their reduced cost points to are unchanged, as in scheme (C) of Appendix C; and the extra cost is one product with K per restart that fires. We call the result sPDLP with snapping. Figure 1 uses it; the benchmark of Section 7 uses Algorithm 1 as stated.

7 Computational results

Setup. We use a benchmark set of 68 instances from the Mittelmann collection, LP relaxations of MIPLIB 2017 models (Gleixner et al., 2021), whose sizes after presolve are listed in Table 3. The baseline is PDLP as implemented in the `FirstOrderLp.jl` code released by Google Research, with its default adaptive normalised-duality-gap restart; sPDLP is the same code with that restart replaced by Algorithm 1. Each run targets $\varepsilon_{\text{opt}} = 10^{-4}$ and is capped at 3600 seconds or 10^6 iterations; sPDLP is run with three seeds per instance. The iteration ratio of an instance, and likewise its time ratio, is the geometric mean over seeds of the sPDLP count divided by the PDLP count, so that a ratio below one favours sPDLP; the overall figure is the geometric mean over instances. On `map18` sPDLP reaches the time limit on all three seeds while PDLP terminates; on `rail02` PDLP reaches the time limit while sPDLP terminates on one seed; ratios are therefore reported on the remaining 66 instances. The hyperparameters, $\kappa = 2.5$, $d = 3$, $L_0 = 200$, the exponent 0.6 in (20), $p_{\max} = 0.7$ and $p_{\min} = 0.003$, were chosen by a grid search on fourteen of the 68 instances, listed in Appendix G, and then fixed for all.

Results. Table 3 in the appendix gives the full results and Table 2 their distribution; Figure 2 shows the per-instance iteration ratios with their variability across seeds. In iterations the rule is close to neutral: the geometric-mean ratio is 1.074, sPDLP needs fewer iterations on 27 of the 66 instances and more on 39, and 58 of the 66 ratios lie within a factor of two of one. A two-sided sign test on the 27 wins gives $p \approx 0.18$, so the benchmark does not distinguish the two methods. The one large effect is `app1-2`, the instance on which the baseline is by far the slowest, where sPDLP needs 4.5 per cent of the baseline's iterations; without it the geometric mean is 1.128. On the losing side, `glass4` is the only pronounced outlier, with a ratio of 4.421; the remaining large losses are factors between 2.0 and 2.7. In wall-clock time the geometric-mean ratio is 1.078 and sPDLP is faster on 33 of the 66 instances. On several small instances the time ratio and the iteration ratio differ widely, which suggests that timing on them is dominated by fixed overheads.

Relation to the theory. Section 4 concerns geometric restarts superimposed on a fixed base method, whereas sPDLP replaces the restart of PDLP by a state-dependent rule, so the benchmark is not a direct test of the theory. Two observations are consistent with it: the losses are far smaller than the exponential penalty that Proposition 1 allows for restarts to fresh draws, and the one large gain occurs on the instance on which the baseline is slowest, the regime that Corollary 1 and Remark 1 point to. On `danoint`, a controlled test with PDHG as the base method, the renewal formula predicts correctly that a constant restart rate slows the method, though it overestimates by how much (Appendix G.1).

8 Conclusion

Randomised restarting can beat PDLP by a margin that grows with the dimension, as the plateau-and-wall family shows, but only when a restart can leave the convex hull of the trajectory, as snapping does. When a run that has not yet finished can expect to finish no later than a fresh run would, no constant restart rate helps; when the iteration count varies little from one restart point to another, as under a two-sided linear rate, none helps appreciably; and in a reduced-state model with monotone dynamics the best state-dependent policy never restarts near the target. The pilot runs of Section 4.4 estimate which case an instance is in; running them and sPDLP with snapping across the benchmark is the natural next step.

Ethics statement

This work concerns algorithms for linear programming and uses only publicly available benchmark instances. It involves no human subjects, personal data or foreseeable harmful application, and we are not aware of ethical concerns specific to it.

Reproducibility statement

Source code is in the supplementary material, together with instructions for use. Proofs of all results are given in Appendices D, E and B, with the diffusion surrogate in Appendix F. Algorithm 1 states the rule in full, and Section 7 gives every hyperparameter, the tolerance, the limits and the number of seeds. Appendix G lists the tuning instances, the hardware and the per-instance results. The instances of Figure 1 are fully specified in Section 1.1, with the family parameters in the caption. The baseline is the publicly released `FirstOrderLp.jl` code.

The use of large language models

A large language model was used to revise the wording and structure of the manuscript, to check the consistency of its notation, to help search the literature on first passage under restart, and to help formulate the example of Theorem 1. The authors have checked all text and all proofs and take full responsibility for the content of the paper.

References

Ben Adcock, Matthew J. Colbrook, and Maksym Neyra-Nesterenko. Restarts subject to approximate sharpness: A parameter-free and optimal scheme for first-order methods. *Foundations of Computational Mathematics*, 26(2):1137–1192, 2025. doi: 10.1007/s10208-024-09673-8.

Saeed Ahmad, Indrani Nayak, Ajay Bansal, Amitabha Nandi, and Dibyendu Das. First passage of a particle in a potential under stochastic resetting: A vanishing transition of optimal resetting rate. *Physical Review E*, 99(2), February 2019. ISSN 2470-0053. doi: 10.1103/physreve.99.022130. URL http://dx.doi.org/10.1103/PhysRevE.99.022130.

Erling D Andersen, Jacek Gondzio, Csaba Mészáros, and Xiaojie Xu. Implementation of interior-point methods for large scale linear programs. In *Interior Point Methods of Mathematical Programming*, volume 5 of *Applied Optimization*, pp. 189–252. Kluwer Academic Publishers, 1996. doi: 10.1007/978-1-4613-3449-1_6.

David Applegate, Oliver Hinder, Haihao Lu, and Miles Lubin. Faster first-order primal-dual methods for linear programming using restarts and sharpness. *Mathematical Programming*, 201(1–2):133–184, October 2022. ISSN 1436-4646. doi: 10.1007/s10107-022-01901-9. URL http://dx.doi.org/10.1007/s10107-022-01901-9.

David Applegate, Mateo Díaz, Haihao Lu, and Miles Lubin. Infeasibility detection with primal-dual hybrid gradient for large-scale linear programming. *SIAM Journal on Optimization*, 34(1):459–484, 2024. doi: 10.1137/22M1510467.

David Applegate, Mateo Díaz, Oliver Hinder, Haihao Lu, Miles Lubin, Brendan O'Donoghue, and Warren Schudy. PDLP: a practical first-order method for large-scale linear programming. *Mathematical Programming Computation*, 2026. doi: 10.1007/s12532-026-00309-2.

David L. Applegate, Mateo D'iaz, Oliver Hinder, Haihao Lu, Miles Lubin, Brendan O'Donoghue, and Warren Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. In *Neural Information Processing Systems*, 2021. URL https://api.semanticscholar.org/CorpusID:235376806.

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Johannes Aspman, Daniel Mastropietro, and Jakub Mareček. First-passage time in space-dependent stochastic resetting. *Physical Review E*, 114(2), 2026. doi: 10.1103/w2yj-91m8. arXiv:2509.11809.

Richard E. Barlow and Frank Proschan. *Statistical Theory of Reliability and Life Testing: Probability Models*. Holt, Rinehart and Winston, New York, 1975.

Dimitri P. Bertsekas and Steven E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Academic Press, New York, 1978.

Daniel Bienstock, Gonzalo Muñoz, and Sebastian Pokutta. Principled deep neural network training through linear programming. *Discrete Optimization*, 49, 2023.

Ofek Lauber Bonomo and Arnab Pal. First passage under restart for discrete space and time: Application to one-dimensional confined lattice random walks. *Physical Review E*, 103(5):052129, 2021. doi: 10.1103/PhysRevE.103.052129.

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(1):120–145, 2011.

Kaihuang Chen, Defeng Sun, Yancheng Yuan, Guojun Zhang, and Xinyuan Zhao. On the relationships among GPU-accelerated first-order methods for solving linear programming. *Journal of Computational Mathematics*, 2026. doi: 10.4208/jcm.2606-m2025-0260.

George B Dantzig, Alex Orden, Philip Wolfe, et al. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific Journal of Mathematics*, 5(2):183–195, 1955. doi: 10.2140/pjm.1955.5.183.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632: 768–774, 2024.

Iddo Eliazar and Shlomi Reuveni. Mean-performance of sharp restart I: statistical roadmap. *Journal of Physics A: Mathematical and Theoretical*, 53(40):405004, 2020. doi: 10.1088/1751-8121/abae8c.

Martin R Evans and Satya N Majumdar. Diffusion with optimal resetting. *Journal of Physics A: Mathematical and Theoretical*, 44(43):435001, October 2011a. ISSN 1751-8121. doi: 10.1088/1751-8113/44/43/435001. URL http://dx.doi.org/10.1088/1751-8113/44/43/435001.

Martin R. Evans and Satya N. Majumdar. Diffusion with stochastic resetting. *Physical Review Letters*, 106(16):160601, 2011b. doi: 10.1103/PhysRevLett.106.160601.

Martin R Evans and Satya N Majumdar. Diffusion with resetting in arbitrary spatial dimension. *Journal of Physics A: Mathematical and Theoretical*, 47(28):285001, June 2014. ISSN 1751-8121. doi: 10.1088/1751-8113/47/28/285001. URL http://dx.doi.org/10.1088/1751-8113/47/28/285001.

Martin R. Evans, Satya N. Majumdar, and Grégory Schehr. Stochastic resetting and applications. *Journal of Physics A: Mathematical and Theoretical*, 53(19):193001, 2020. doi: 10.1088/1751-8121/ab7cfe.

John Fearnley and Rahul Savani. The complexity of the simplex method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 201–208, 2015. doi: 10.1145/2746539.2746558.

Olivier Fercoq and Zheng Qu. Adaptive restart of accelerated gradient methods under local quadratic growth condition. *IMA Journal of Numerical Analysis*, 39(4):2069–2095, 2019. doi: 10.1093/imanum/drz007.

Philip E Gill, Walter Murray, Michael A Saunders, John A Tomlin, and Margaret H Wright. On projected newton barrier methods for linear programming and an equivalence to karmarkar's projective method. *Mathematical programming*, 36(2):183–209, 1986. doi: 10.1007/BF02592025.

Ambros Gleixner, Gregor Hendel, Gerald Gamrath, Tobias Achterberg, Michael Bastubbe, Timo Berthold, Philipp Christophel, Kati Jarck, Thorsten Koch, Jeff Linderoth, Marco Lübbecke, Hans D. Mittelmann, Derya Ozyurt, Ted K. Ralphs, Domenico Salvagnin, and Yuji Shinano. MIPLIB 2017: data-driven compilation of the 6th mixed-integer programming library. *Mathematical Programming Computation*, 13(3):443–490, 2021. doi: 10.1007/s12532-020-00194-3.

Carla P. Gomes, Bart Selman, and Henry Kautz. Boosting combinatorial search through randomization. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, Madison, WI, 1998. AAAI Press.

Carla P. Gomes, Bart Selman, Nuno Crato, and Henry Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning*, 24(1–2):67–100, 2000. doi: 10.1023/A:1006314320276.

Jacek Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012. doi: 10.1016/j.ejor.2011.09.017.

Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pp. 302–311, 1984. doi: 10.1145/800057.808695.

Zhenwei Lin, Zikai Xiong, Dongdong Ge, and Yinyu Ye. Pdcs: A primal-dual large-scale conic programming solver with gpu enhancements. *arXiv preprint arXiv:2505.00311*, 2025.

Haihao Lu and Jinwen Yang. On the geometry and refined rate of primal–dual hybrid gradient for linear programming. *Mathematical Programming*, 212(1-2):349–387, 2024a. doi: 10.1007/s10107-024-02109-9.

Haihao Lu and Jinwen Yang. Restarted Halpern PDHG for linear programming. *arXiv preprint arXiv:2407.16144*, 2024b.

Haihao Lu and Jinwen Yang. cupdlp. jl: A gpu implementation of restarted primal-dual hybrid gradient for linear programming in julia. *Operations Research*, 73(6):3440–3452, 2025a. doi: 10.1287/opre.2024.1069.

Haihao Lu and Jinwen Yang. A practical and optimal first-order method for large-scale convex quadratic programming. *Mathematical Programming*, 215(1-2):771–808, 2025b. doi: 10.1007/s10107-025-02241-0.

Michael Luby, Alistair Sinclair, and David Zuckerman. Optimal speedup of Las Vegas algorithms. *Information Processing Letters*, 47(4):173–180, 1993. doi: 10.1016/0020-0190(93)90029-9.

Brendan O'Donoghue and Emmanuel Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, 2015. doi: 10.1007/s10208-013-9150-3.

Arnab Pal and Vishwajeet Vishwakarma Prasad. Landau-like expansion for phase transitions in stochastic resetting. *Physical Review Research*, 1(3):032001, 2019. doi: 10.1103/PhysRevResearch.1.032001.

Arnab Pal and Shlomi Reuveni. First passage under restart. *Physical Review Letters*, 118(3):030603, 2017. doi: 10.1103/PhysRevLett.118.030603.

Arnab Pal, Sarah Kostinski, and Shlomi Reuveni. The inspection paradox in stochastic resetting. *Journal of Physics A: Mathematical and Theoretical*, 55(2):021001, 2022. doi: 10.1088/1751-8121/ac3cdf.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1310–1318, 2013.

George Philipp, Dawn Song, and Jaime G. Carbonell. The exploding gradient problem demystified: definition, prevalence, impact, origin, tradeoffs, and solutions. *arXiv preprint arXiv:1712.05577*, 2017.

Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 2020.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, 1994. doi: 10.1002/9780470316887.

Sidney Redner. *A Guide to First-Passage Processes*. Cambridge University Press, 2001. doi: 10.1017/CBO9780511606014.

James Renegar and Benjamin Grimmer. A simple nearly optimal restart scheme for speeding up first-order methods. *Foundations of Computational Mathematics*, 22(1):211–256, 2022. doi: 10.1007/s10208-021-09502-2.

Shlomi Reuveni. Optimal stochastic restart renders fluctuations in first passage times universal. *Physical Review Letters*, 116(17):170601, 2016. doi: 10.1103/PhysRevLett.116.170601.

Vincent Roulet and Alexandre d'Aspremont. Sharpness, restart, and acceleration. *SIAM Journal on Optimization*, 30(1):262–289, 2020. doi: 10.1137/18M1224568.

Satinder P. Singh, Tommi Jaakkola, and Michael I. Jordan. Learning without state-estimation in partially observable Markovian decision processes. In *Machine Learning Proceedings 1994 (ICML)*, pp. 284–292. Morgan Kaufmann, 1994. doi: 10.1016/B978-1-55860-335-6.50042-8.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 2023.

Stephen J Wright. *Primal-dual interior-point methods*. SIAM, 1997. doi: 10.1137/1.9781611971453.

Zikai Xiong. Accessible complexity bounds for restarted PDHG on linear programs with a unique optimizer. *Mathematics of Operations Research*, 2026. doi: 10.1287/moor.2024.0738.

Zikai Xiong and Robert M. Freund. The role of level-set geometry on the performance of PDHG for conic linear optimization. *arXiv preprint arXiv:2406.01942*, 2024.

Zikai Xiong and Robert M. Freund. Computational guarantees for restarted PDHG for LP based on "limiting error ratios" and LP sharpness. *Mathematical Programming*, 2026. doi: 10.1007/s10107-026-02358-w.

A THE CONVERGENCE SCORE

The dual of (1) is $\max_{y,\lambda}\ q^\top y + l^\top\lambda^+ - u^\top\lambda^-$ subject to $c - K^\top y = \lambda$, $y_G \geq 0$ and $\lambda \in \Lambda$, where $y = (y_G, y_A)$ is split as in Y and Λ is the set of reduced costs compatible with the bounds. The three terms of the score (3) are, as in Applegate et al. (2021),

$$r_{\mathrm{p}}^k := \frac{\big\|\big(Ax^k - b,\ (h - Gx^k)^+\big)\big\|_2}{1 + \|q\|_2}, \qquad r_{\mathrm{d}}^k := \frac{\|c - K^\top y^k - \lambda^k\|_2}{1 + \|c\|_2},$$

$$g^k := \frac{|q^\top y^k + l^\top(\lambda^k)^+ - u^\top(\lambda^k)^- - c^\top x^k|}{1 + |q^\top y^k + l^\top(\lambda^k)^+ - u^\top(\lambda^k)^-| + |c^\top x^k|}, \qquad \lambda^k = \mathrm{proj}_\Lambda(c - K^\top y^k),$$

the relative primal residual, the relative dual residual and the relative optimality gap.

B PROOF OF THEOREM 1

Write w_k for the w-coordinate of the k-th iterate and $W_k := \max_{0\leq i\leq k} w_i$. Since w enters no constraint, its component of $c - K^\top y$ equals $-\delta$ whatever y is, so a PDHG step gives $w_{k+1} = \min\{1,\ w_k + \tau\delta\} \leq W_k + \tau\delta$. A convex combination of earlier iterates has w-coordinate at most W_k. In either case $W_{k+1} \leq W_k + \tau\delta$, and from $w_0 = 0$ induction gives $W_k \leq k\tau\delta$. The target requires $w_k = 1$, hence $k\tau\delta \geq 1$ and $k \geq L$. The step-size condition gives $\tau < 1/\|\hat{K}\|_2 = 1/\sqrt{n}$, so $L \geq 1/(\tau\delta) > \sqrt{n}/\delta$. This proves (i).

For (ii), each restart draws afresh from a fixed law, so the restart-free counts of successive epochs are independent and identically distributed, and Assumption 1 holds. With probability one half the restart sets $w = 1$; the PDHG step keeps $w = 1$, and the block reaches its target within f iterations, so $N \leq f$ on that event. Hence $\phi(p) = \mathbb{E}[(1-p)^N] \geq \frac{1}{2}(1-p)^f$, and Theorem 2 gives

$$T_p = \frac{(1-p)\big(1-\phi(p)\big)}{p\,\phi(p)} \leq \frac{1-p}{p\,\phi(p)} \leq \frac{2}{p\,(1-p)^{f-1}}.$$

At $p = 1/f$ this is $2f\,(1-1/f)^{-(f-1)} \leq 2\mathrm{e}f$, because $(1-1/f)^{f-1} \geq \mathrm{e}^{-1}$ for every $f \geq 2$. □

C MANY WALLS, PARTIAL RESTARTS, AND TRAINING NEURAL NETWORKS

This appendix extends the example of Section 1.1 to many walls. Restarting the whole iterate then loses its advantage, whereas restarting only the coordinates that have not yet settled keeps it, with a separation exponential in the number of walls. We then relate the model to vanishing and exploding gradients and to linear programmes for training ReLU networks, and we state what is not shown.

C.1 THE MODEL AND THE RESULT

Take n_w variables $w = (w_1, \ldots, w_{n_w}) \in [0,1]^{n_w}$, each with cost $-\delta$ and entering no constraint, together with the block of Section 1.1, of norm $\|\hat{K}\|_2 = \sqrt{n}$, and run PDHG with $\tau = \sigma$ and $\tau\|\hat{K}\|_2 < 1$. Since w is decoupled, each w_i moves to $\min\{1, w_i + \tau\delta\}$ at every step, independently of everything else. The target is $w = \mathbf{1}$ together with a target set $\mathcal{B}$ for the block, and we assume that from every starting point the block iterates enter $\mathcal{B}$ within f iterations and never leave it. The second half of this assumption holds when $\mathcal{B}$ is a ball about the solution set in the norm in which PDHG is Fejér monotone (Chambolle & Pock, 2011); the first half is the idealisation. Call w_i *settled* when $w_i = 1$: it sits at a bound, and its reduced cost $-\delta$ holds it there, a test that uses only the sign of the gradient. We compare three schemes, with $L := \lceil 1/(\tau\delta)\rceil$ as before.

- (T) Methods whose iterates are PDHG steps or convex combinations of earlier iterates, started at $w = \mathbf{0}$.
- (G) Global geometric restarting at rate p: with probability p at each iteration the whole iterate is replaced by a fresh draw, with $w_1, \ldots, w_{n_w}$ independent and uniform on $\{0, 1\}$ and the block from any fixed law.

(C) Coordinate-wise restarting at rate $p \in (0,1]$: after each PDHG step, each unsettled w_i is replaced, independently and with probability p, by a uniform draw from $\{0,1\}$; the block is never reset.

Proposition 2 (Many walls)**.** *Let $n_w \geq 1$ and $f \geq 1$.*

(i) Every method of type (T) needs at least $L \geq \sqrt{n}/\delta$ iterations.

(ii) Under (G), for every $p \in (0,1)$, the expected number of iterations is at least $\frac{1}{4}\min\{L,\ 2^{n_w}\}$.

(iii) Under (C), from any starting point, the expected number of iterations is at most $f + 1 + \frac{2}{p}(1 + \ln n_w)$.

Proof. (i) is Theorem 1(i) applied to w_1.

(ii) Fresh draws make the restart-free counts of successive epochs independent, so Assumption 1 holds and, by the proof of Theorem 2, $T_p = \mathbb{E}[\min(N,\Gamma)]/\mathbb{P}[\Gamma \geq N]$. Let $\varpi := 2^{-n_w} \leq \frac{1}{2}$ and $u := 1 - p$. With probability ϖ the draw has $w = \mathbf{1}$; otherwise some w_i starts at 0, and since it rises by exactly $\tau\delta$ per step it needs L steps, so $N \geq L$. Hence $\mathbb{P}[\Gamma \geq N] \leq \varpi + u^{L-1}$ and $\mathbb{E}[\min(N,\Gamma)] \geq \max\{1,\ \frac{1}{2}\mathbb{E}[\min(L,\Gamma)]\}$, where $\mathbb{E}[\min(L,\Gamma)] = \sum_{t=0}^{L-1} u^t$. If $u^{L-1} \leq \varpi$, then $T_p \geq 1/(2\varpi) = 2^{n_w-1}$. Otherwise $\mathbb{P}[\Gamma \geq N] \leq 2u^{L-1}$ and

$$T_p \ \geq\ \frac{\sum_{t=0}^{L-1} u^t}{4\,u^{L-1}} \ =\ \frac{1}{4}\sum_{i=0}^{L-1} u^{-i} \ \geq\ \frac{L}{4}.$$

(iii) Let Θ_i be the first iteration at which $w_i = 1$. A settled coordinate stays settled, since the PDHG step keeps it at 1 and it is never reset. Before Θ_i, at every iteration, conditionally on the past, w_i is reset with probability p and then set to 1 with probability $\frac{1}{2}$, so $\mathbb{P}[\Theta_i > t] \leq (1 - p/2)^t$. By the union bound, $\mathbb{P}[\max_i \Theta_i > t] \leq n_w(1 - p/2)^t$. With $t_0 := \lceil (2/p)\ln n_w \rceil$, so that $n_w(1 - p/2)^{t_0} \leq n_w \mathrm{e}^{-pt_0/2} \leq 1$,

$$\mathbb{E}\big[\max_i \Theta_i\big] = \sum_{t \geq 0} \mathbb{P}\big[\max_i \Theta_i > t\big] \leq t_0 + \sum_{t \geq t_0} n_w(1 - p/2)^t \leq \frac{2}{p}\ln n_w + 1 + \frac{2}{p}.$$

The block is never reset, so it follows plain PDHG, enters $\mathcal{B}$ within f iterations and stays there. The target is therefore reached by iteration $\max\{f, \max_i \Theta_i\} \leq f + \max_i \Theta_i$. □

Table 1: Iteration counts on the many-walls model of Proposition 2, with n_w walls, a block of norm $\sqrt{n}$ that settles within f iterations, and $p = \frac{1}{2}$ in scheme (C).

Scheme	What a restart replaces	Expected iterations
(T) PDLP-type restarts	the iterate, by a convex combination of past iterates	at least $\sqrt{n}/\delta$
(G) global restarts, any rate	the whole iterate, by a fresh draw	at least $\frac{1}{4}\min\{\sqrt{n}/\delta,\ 2^{n_w}\}$
(C) coordinate-wise restarts	each unsettled wall coordinate	at most $f + 5 + 4\ln n_w$

Table 1 collects the bounds. With n_w walls and $2^{n_w} \leq \sqrt{n}/\delta$, restarting the whole iterate is slower than restarting stuck coordinates by a factor of order $2^{n_w}/(f + \ln n_w)$. Resetting a whole iterate at random is the mechanism of the stochastic-resetting literature, but a restart that lands on all the walls at once becomes exponentially unlikely as their number grows, so the whole iterate is the wrong unit to reset.

C.2 Vanishing and exploding gradients

Each wall coordinate feels a gradient of only δ, the analogue of a vanishing gradient (Pascanu et al., 2013). The block's large norm $\sqrt{n}$ forces the step size below $1/\sqrt{n}$, much as large gradients in some directions cap the usable learning rate (Philipp et al., 2017). Neither slows the method alone: the plateau takes about $\|\hat{K}\|_2/\delta$ iterations, a ratio of the largest scale in the problem to the smallest gradient that still matters. In this convex setting that ratio is where the two gradient problems appear

together; in a deep network both arise instead from products of Jacobians across layers, which a single linear programme has no reason to contain. The wall itself is a bound active at the optimum, as for a unit that should be inactive or a weight that should be zero. Scheme (C) is close to what deep-learning practice already does when it reinitialises units that have stopped learning: the dormant-neuron resets of Sokar et al. (2023), continual backpropagation (Dohare et al., 2024), and shrink-and-perturb (Ash & Adams, 2020). Proposition 2 shows, in an idealised model, why such resets should act on individual units rather than on the whole network.

C.3 Walls in linear programmes for training ReLU networks

Bienstock et al. (2023) construct, for a given architecture, loss and sample size, a single polytope with one face for each data set, such that optimising a linear function over the face of a data set trains the network to within an additive ϵ. The polytope comes from a discretisation of the parameters into a binary problem of low treewidth. Its size is linear in the number of samples and exponential in the size of the architecture, which puts it beyond simplex and interior-point codes and within the reach of first-order methods such as PDLP. Two features of this formulation match the model. At an optimal vertex the lifted binary variables take the values 0 and 1, so that every such coordinate sits at a wall. Neighbouring points of the parameter grid differ in loss by an amount controlled by the grid spacing, so near-optimal alternatives should have small reduced costs, which are plateaus. A coordinate-wise restart then sets an undecided binary variable to a random bound, a randomised rounding of one variable. The resulting point may violate the coupling constraints, which PDHG tolerates, since its iterates need not be feasible.

A second source of walls is the convex reformulation of two-layer ReLU training by Pilanci & Ergen (2020), a group-sparse programme over the activation patterns of the data. With absolute loss and ℓ_1 penalties in place of the squared loss and the group norm, it becomes a linear programme of the same shape; we use this analogue only as an illustration and do not claim that it is equivalent to training a network. At an optimum most pattern blocks are zero; a pattern that is nearly useful has a small reduced cost and drifts slowly to zero; and a coordinate-wise restart that sets a block to zero is the pruning of one neuron.

C.4 What is not shown

Proposition 2 is proved for decoupled walls, which presolve would remove, and under an assumption on the block. In both formulations above the walls are coupled through the constraints, so the proposition does not apply to them directly; we have neither proved that their reduced costs are small nor measured the resulting iteration counts. The factor $\sqrt{n}$ comes from the operator norm and can be reduced by diagonal preconditioning, and a plateau delays termination under the score (3) only while δ is large compared with the tolerance. What the proposition does establish is the direction a realistic test should take: restart schemes that act on individual coordinates, triggered by the sign test that identifies settled coordinates, and compared with global restarts on linear programmes of the kinds above.

D Proofs for Section 4

D.1 Proof of Theorem 2

The formula is eq. (20) of Bonomo & Pal (2021); we give a proof in our notation. Let Γ be the index of the first restart, so that Γ is geometric on $\{1, 2, \dots\}$ with $\mathbb{P}[\Gamma > t] = (1-p)^t$, independent of N. Decompose the run into *epochs*, an epoch being a maximal stretch of iterations between two consecutive restarts. By Assumption 1 the epochs are independent and identically distributed; let $(N_i, \Gamma_i)_{i \geq 1}$ be the corresponding independent copies.

Consider a single epoch. Under our accounting convention a restart is applied *after* an iteration, so the epoch ends by absorption in $\mathcal{T}$ if and only if no restart occurs after any of the iterations $1, \dots, N-1$, that is, if and only if $\Gamma \geq N$; and the number of iterations consumed by the epoch is $\ell := \min(N, \Gamma)$ in either case. Hence the absorption probability is

$$\phi_{\mathrm{abs}} := \mathbb{P}[\Gamma \geq N] = \mathbb{E}[(1-p)^{N-1}] = \phi(p)/(1-p), \tag{23}$$

which is strictly positive because $\mathbb{P}[N < \infty] > 0$ and at most 1, with equality only in the degenerate case $\mathbb{P}[N = 1] = 1$, where $T_p \equiv 1$. For the epoch length, summing the tail and using independence,

$$\mathbb{E}[\ell] = \sum_{t \geq 0} \mathbb{P}[N > t]\, \mathbb{P}[\Gamma > t] = \sum_{t \geq 0} \mathbb{P}[N > t]\, (1-p)^t = \frac{1 - \mathbb{E}[(1-p)^N]}{p} = \frac{1 - \phi(p)}{p},$$

where the third equality is the identity $\sum_{t \geq 0} u^t\, \mathbb{P}[N > t] = \big(1 - \mathbb{E}[u^N]\big)/(1-u)$ for $u \in [0, 1)$, obtained by exchanging the order of summation. In particular $\mathbb{E}[\ell] \leq \mathbb{E}[\Gamma] = 1/p < \infty$.

Let $J := \min\{i \geq 1 : \Gamma_i \geq N_i\}$ be the index of the absorbing epoch. Then J is geometric with success probability ϕ_{abs}, so $\mathbb{E}[J] = 1/\phi_{\mathrm{abs}} < \infty$, and $\{J \geq i\}$ is determined by epochs $1, \ldots, i-1$, hence independent of ℓ_i. The total iteration count is $\sum_{i=1}^{J} \ell_i$, and Wald's identity gives

$$T_p = \mathbb{E}\Big[\sum_{i=1}^{J} \ell_i\Big] = \mathbb{E}[J]\, \mathbb{E}[\ell] = \frac{1-p}{\phi(p)} \cdot \frac{1 - \phi(p)}{p},$$

which is (6), and is finite. Finally, $\phi(p) \to \mathbb{P}[N < \infty]$ and $(1-\phi(p))/p = \sum_{t \geq 0} \mathbb{P}[N > t](1-p)^t \to \mathbb{E}[N]$ as $p \downarrow 0$ by monotone convergence, which gives the last claim. □

D.2 Two extreme cases

Remark 2 (Two extreme cases). If $N \equiv t$ is deterministic then (6) telescopes: with $u = 1 - p$,

$$T_p = \frac{u\,(1 - u^t)}{(1-u)\,u^t} = \sum_{k=0}^{t-1} u^{-k} \;=:\; T_p^{\det}(t) \;\geq\; t, \tag{24}$$

and T_p is strictly increasing in p for $t \geq 2$: restarting a method whose running time is exactly predictable can only hurt, at every p. If instead N is geometric, so that the restart-free method is memoryless, then (6) gives $T_p = \mathbb{E}[N]$ for all p, and (8) holds with equality.

D.3 An arbitrary starting point

Theorem 2 assumes that the run begins at a fresh restart point; the version for a solve that begins at a user-supplied z^0 follows from the same one-epoch decomposition.

Corollary 6 (Arbitrary starting point)**.** *Let N_0 be the restart-free iteration count from z^0 and $\phi_0(p) := \mathbb{E}[(1-p)^{N_0}]$. Then the expected iteration count from z^0 is*

$$T_p(z^0) \;=\; \frac{1 - \phi_0(p)}{p} \;+\; \Big(1 - \frac{\phi_0(p)}{1-p}\Big)\, T_p. \tag{25}$$

Proof. The first epoch consumes $\mathbb{E}[\min(N_0, \Gamma)] = (1 - \phi_0(p))/p$ iterations and absorbs with probability $\phi_0(p)/(1-p)$, as in (23). Otherwise the process starts afresh from μ and contributes T_p. □

D.4 Proof of Corollary 2

The criterion is (25) of Bonomo & Pal (2021); we give a proof in our notation. Let $\psi(u) := \mathbb{E}[u^N]$ be the probability generating function of N, so that $\phi(p) = \psi(1-p)$. Since $\mathbb{E}[N^2] < \infty$, Abel's theorem gives $\psi(1) = 1$, $\psi'(1^-) = \mathbb{E}[N]$ and $\psi''(1^-) = \mathbb{E}[N(N-1)]$, hence

$$\phi(p) = 1 - pm + p^2 A + o(p^2), \qquad m := \mathbb{E}[N], \quad A := \tfrac{1}{2}\mathbb{E}[N(N-1)] = \tfrac{1}{2}\big(v + m^2 - m\big),$$

with $v := \mathrm{Var}(N)$. Substituting into (6),

$$\frac{1 - \phi(p)}{p} = m - pA + o(p), \qquad \frac{1}{\phi(p)} = 1 + pm + o(p),$$

so that $\frac{1-\phi}{p\phi} = m + p\,(m^2 - A) + o(p)$ and therefore

$$T_p = (1-p)\Big(m + p(m^2 - A)\Big) + o(p) = m + p\big(m^2 - m - A\big) + o(p) = m + \frac{p}{2}\big(m^2 - m - v\big) + o(p).$$

This is (7), and both implications follow from the sign of the first-order term. □

D.5 Proofs of Theorem 3 and Corollary 3

Write $\bar{F}(j) := \mathbb{P}[N > j]$ for the tail of N, so that $\mathbb{E}[(N-j)^+] = \sum_{i \geq j} \bar{F}(i)$ and $m(j) = \mathbb{E}[(N-j)^+]/\bar{F}(j)$ whenever $\bar{F}(j) > 0$. Throughout, a term $\bar{F}(j)\, m(j)$ with $\bar{F}(j) = 0$ is read as $\mathbb{E}[(N-j)^+] = 0$.

Proof of Theorem 3. Let Γ be geometric on $\{1, 2, \dots\}$ with $\mathbb{P}[\Gamma = j] = p(1-p)^{j-1}$, independent of N. From the proof of Theorem 2, $T_p = \mathbb{E}[\min(N, \Gamma)]/\mathbb{P}[\Gamma \geq N]$ with $\mathbb{P}[\Gamma \geq N] = \phi(p)/(1-p)$. Since $N = \min(N, \Gamma) + (N - \Gamma)^+$ and $m < \infty$,

$$\mathbb{E}[\min(N,\Gamma)] = m - \mathbb{E}[(N-\Gamma)^+] = m - \sum_{j \geq 1} \mathbb{P}[\Gamma = j]\, \mathbb{E}[(N-j)^+] = m - \sum_{j \geq 1} \mathbb{P}[\Gamma = j]\, \bar{F}(j)\, m(j).$$

Also $\mathbb{P}[\Gamma < N] = \sum_{j \geq 1} \mathbb{P}[\Gamma = j]\, \bar{F}(j)$, so $m = m\, \mathbb{P}[\Gamma \geq N] + \sum_{j \geq 1} \mathbb{P}[\Gamma = j]\, \bar{F}(j)\, m$. Subtracting,

$$\mathbb{E}[\min(N,\Gamma)] - m\, \mathbb{P}[\Gamma \geq N] = \sum_{j \geq 1} \mathbb{P}[\Gamma = j]\, \bar{F}(j)\, \big(m - m(j)\big),$$

and dividing by $\mathbb{P}[\Gamma \geq N] = \phi(p)/(1-p) > 0$ gives (9). □

A memorylessness lemma. The strictness claim of Corollary 3 rests on the following elementary fact.

Lemma 2. *Let N take values in $\{1, 2, \dots\}$ with $m = \mathbb{E}[N] < \infty$. Then $m(j) = m$ for every $j \geq 0$ with $\bar{F}(j) > 0$ if and only if N is geometric with parameter $1/m$.*

Proof. A geometric N is memoryless, so $m(j) = m$. Conversely, suppose $m(j) = m$ whenever $\bar{F}(j) > 0$, that is, $\sum_{i \geq j} \bar{F}(i) = m\, \bar{F}(j)$. Fix j with $\bar{F}(j) > 0$. If $\bar{F}(j+1) = 0$ the identity reads $\bar{F}(j) = m\, \bar{F}(j)$, so $m = 1$ and $N \equiv 1$, which is geometric with parameter 1. Otherwise subtract the identity at $j+1$ from the one at j to obtain $\bar{F}(j) = m\big(\bar{F}(j) - \bar{F}(j+1)\big)$, that is, $\bar{F}(j+1) = (1 - 1/m)\, \bar{F}(j)$. Starting from $\bar{F}(0) = 1$, induction gives $\bar{F}(j) = (1-1/m)^j$ for all j, the tail of a geometric law with parameter $1/m$. □

Proof of Corollary 3. (i) and (ii): every weight $\frac{1-p}{\phi(p)}\, p(1-p)^{j-1} \bar{F}(j)$ in (9) is non-negative, and strictly positive whenever $\bar{F}(j) > 0$, so the sign of $T_p - T_0$ is that of the terms $m - m(j)$ when these all have the same sign. For strictness in (i): if N is not geometric, Lemma 2 gives some $j \geq 1$ with $\bar{F}(j) > 0$ and $m(j) \neq m$; under NBUE this means $m(j) < m$, so that term is strictly positive and $T_p > T_0$ for every $p \in (0, 1)$. If N is geometric, Remark 2 gives $T_p = T_0$.

(iii): For $j \geq 0$ and $k \geq 1$ with $\bar{F}(j) > 0$,

$$\mathbb{P}[N > j + k \mid N > j] = \prod_{t=j+1}^{j+k} \mathbb{P}[N > t \mid N \geq t] = \prod_{t=j+1}^{j+k} \big(1 - H(t)\big),$$

using $\{N > t-1\} = \{N \geq t\}$. If H is non-decreasing, each factor $1 - H(j+i)$, $i = 1, \dots, k$, is non-increasing in j, so the conditional tail is non-increasing in j for every k, and therefore $m(j) = \sum_{k \geq 1} \mathbb{P}[N > j + k \mid N > j]$ is non-increasing in j. In particular $m(j) \leq m(0) = m$, which is NBUE. This is the discrete form of the classical chain IFR ⇒ DMRL ⇒ NBUE (Barlow & Proschan, 1975). □

Recovering Corollary 2 from Theorem 3. Assume $\mathbb{E}[N^2] < \infty$. Then $\sum_{j \geq 1} \bar{F}(j) = m - 1$ and $\sum_{j \geq 1} \bar{F}(j)\, m(j) = \sum_{j \geq 1} \mathbb{E}[(N-j)^+] = \mathbb{E}\big[\frac{1}{2} N(N-1)\big]$, so $\sum_{j \geq 1} \bar{F}(j)\, |m - m(j)| < \infty$. As $p \downarrow 0$, $(1-p)/\phi(p) \to 1$ and, by dominated convergence, $\sum_{j \geq 1} (1-p)^{j-1} \bar{F}(j) \big(m - m(j)\big) \to \sum_{j \geq 1} \bar{F}(j) \big(m - m(j)\big) = m(m-1) - \frac{1}{2}\mathbb{E}[N(N-1)]$. Hence, with $v = \mathrm{Var}(N)$,

$$T_p - T_0 = p\Big(m(m-1) - \tfrac{1}{2}\big(v + m^2 - m\big)\Big) + o(p) = \frac{p}{2}\big(m^2 - m - v\big) + o(p),$$

which is (7).

D.6 Proofs of Proposition 1 and Corollary 4

Let Γ be as above. Since $t_- \le N \le t_+$ almost surely, $\min(t_-,\Gamma) \le \min(N,\Gamma) \le \min(t_+,\Gamma)$ pointwise and $\mathbb{P}[\Gamma \ge t_+] \le \mathbb{P}[\Gamma \ge N] \le \mathbb{P}[\Gamma \ge t_-]$, all three probabilities being positive. Hence

$$\frac{\mathbb{E}[\min(t_-,\Gamma)]}{\mathbb{P}[\Gamma \ge t_-]} \;\le\; \frac{\mathbb{E}[\min(N,\Gamma)]}{\mathbb{P}[\Gamma \ge N]} = T_p \;\le\; \frac{\mathbb{E}[\min(t_+,\Gamma)]}{\mathbb{P}[\Gamma \ge t_+]}.$$

The outer expressions are the values of (6) for the deterministic counts t_- and t_+, which Remark 2 evaluates as $T_p^{\det}(t_\pm) = \sum_{k=0}^{t_\pm - 1}(1-p)^{-k}$. This is the sandwich in (10). Since every summand of $T_p^{\det}(t_-)$ is at least 1, $T_p \ge t_-$ for every p, while $T_0 = \mathbb{E}[N] \le t_+$; dividing gives the bound on the speed-up.

For the last claim, Popoviciu's inequality for a random variable supported in $[t_-,t_+]$ gives $\mathrm{Var}(N) \le (t_+ - t_-)^2/4$, and $x \mapsto x^2 - x$ is increasing on $[1,\infty)$ with $m \ge t_- \ge 1$, so $m^2 - m \ge t_-(t_- - 1)$. Under the stated condition, $\mathrm{Var}(N) \le (t_+ - t_-)^2/4 < t_-(t_- - 1) \le m^2 - m$, that is, the inequality in (8) is reversed and strict; $\mathbb{E}[N^2] < \infty$ holds trivially, so Corollary 2 gives $T_p > T_0$ for all sufficiently small $p > 0$. □

Proof of Corollary 4. The difference $t_+(\varepsilon) - t_-(\varepsilon)$ is bounded by $\Delta := \log(\rho_+/\rho_-)/\log(1/\gamma) + t_0 + 1$ independently of ε, while $t_-(\varepsilon) \to \infty$ as $\varepsilon \downarrow 0$. Claim (i) is the speed-up bound of Proposition 1, since $t_+/t_- \le 1 + \Delta/t_- \to 1$. Claim (ii) is the last sentence of the proposition, whose condition $(t_+ - t_-)^2 < 4t_-(t_- - 1)$ holds once $\Delta^2 < 4t_-(t_- - 1)$, that is, for all ε below some ε_0. □

D.7 Charging the restart

Remark 3 (Charging the restart). If each restart is charged one iteration, the same argument with $\mathbb{P}[\Gamma > N]$ in place of $\mathbb{P}[\Gamma \ge N]$ in (23) yields $T_p = (1-\phi(p))/(p\,\phi(p))$ and the criterion $\mathrm{CV}(N)^2 > 1 + 1/\mathbb{E}[N]$. Restarting is then strictly harmful in the memoryless case, by exactly the wasted iteration. The structure of every statement in Section 4 is unaffected; the constants are not. As Remark 4 notes, this convention yields the same formula as letting a restart pre-empt an absorption, which is the convention of Section 5.

E Proofs for Section 5

Remark 4 (A restart may pre-empt absorption). The Bellman recursion (14) applies the restart action with probability a at every state outside $\mathcal{T}$, irrespective of where the accompanying iteration would have landed. Repeating the one-epoch argument of Appendix D.1 with $\mathbb{P}[\Gamma > N] = \phi(p)$ in place of $\mathbb{P}[\Gamma \ge N]$ gives, for a constant rate p,

$$\tilde{T}_p = \frac{1-\phi(p)}{p\,\phi(p)} = \frac{T_p}{1-p}, \qquad \tilde{T}_p(z^0) = \frac{1-\phi_0(p)}{p} + \big(1-\phi_0(p)\big)\tilde{T}_p \;\le\; \frac{1}{p} + \tilde{T}_p, \tag{26}$$

which is the expression of Remark 3; the second formula is the counterpart of Corollary 6 and gives the uniform bound (13). All statements of Section 5 are qualitative and hold under either convention.

Proof of Lemma 1. Equation (14) is the Bellman equation of the dynamic programme: one iteration is always performed, after which, with probability $1-a$, the state is distributed as $P(\cdot \mid R)$ and, with probability a, as μ. The map $a \mapsto Q^\star(R) + a\big(C^\star - Q^\star(R)\big)$ is affine, so its minimum over the compact set $\mathcal{A}$ is attained at $\max\mathcal{A}$ when the slope is negative, at $\min\mathcal{A}$ when it is positive, and everywhere when it vanishes. Measurability of the optimal selector is elementary, since it is determined by the sign of $Q^\star(R) - C^\star$. □

Proof of Theorem 4. (i) Set $T^{(0)} \equiv 0$ and, for $j \ge 0$,

$$T^{(j+1)}(R) := \mathbb{1}\{R > \varepsilon\}\Big(1 + \min_{a\in\mathcal{A}}\big[(1-a)\,Q^{(j)}(R) + a\,C^{(j)}\big]\Big),$$

where $Q^{(j)}(R) = \int T^{(j)}\,dP(\cdot \mid R)$ and $C^{(j)} = \int T^{(j)}\,d\mu$. Under Assumption 2, $T^{(j)} \uparrow T^\star$ pointwise, with $T^\star \le \bar{T}$ by (13) (Bertsekas & Shreve, 1978, Ch. 8–9). We show by induction

that each $T^{(j)}$ is non-negative and non-decreasing. This holds for $T^{(0)}$. If it holds for $T^{(j)}$, then $T^{(j)}$ is bounded and non-decreasing, so Assumption 3 makes $Q^{(j)}$ non-decreasing, while $C^{(j)}$ is a constant. By affineness the minimum over $\mathcal{A}$ equals the minimum of the two values at $\underline{a} = \min \mathcal{A}$ and $\bar{a} = \max \mathcal{A}$, each a non-decreasing affine function of $Q^{(j)}(R)$ with slope $1 - a \geq 0$, and a minimum of non-decreasing functions is non-decreasing. Multiplying by $\mathbb{1}\{R > \varepsilon\}$ preserves monotonicity, because the indicator is non-decreasing and the other factor non-negative. Letting $j \to \infty$, $T^\star$ is non-decreasing, and by Assumption 3 so is $Q^\star$.

(ii) If $R \leq R'$ and $Q^\star(R) > C^\star$, then $Q^\star(R') \geq Q^\star(R) > C^\star$ by (i); an up-set of an interval of $\mathbb{R}$ is an interval of the stated form.

(iii) Combine (ii) with Lemma 1: on $\{R > \beta^\star\}$ the action $\max \mathcal{A}$ is optimal; on $\{R \leq \beta^\star\}$, $Q^\star(R) \leq C^\star$ and $\min \mathcal{A}$ is optimal, uniquely if the inequality is strict and among all actions if it is an equality. A stationary policy attaining the minimum in (14) at every state is optimal for a dynamic programme with non-negative costs (Bertsekas & Shreve, 1978, Ch. 8–9). □

Proof of Corollary 5. The first claim is (16) with $\min \mathcal{A} = 0$. For the second, $T^\star$ is non-decreasing and vanishes on $\mathcal{T}$, so $C^\star \geq \omega\, T^\star(\xi) \geq \omega$, where $T^\star(\xi) \geq 1$ by (14) since $\xi > \varepsilon$. On the other side, $T^\star \leq \bar{T}$ and $T^\star = 0$ on $\mathcal{T}$, so $Q^\star(R) \leq \bar{T}\, \mathbb{P}[R_1 > \varepsilon \mid R_0 = R]$. By (17) this is $< \omega\, T^\star(\xi) \leq C^\star$ on a right neighbourhood of ε, so $\beta^\star > \varepsilon$ by (15). Since $T^\star(\xi) \geq 1$, the condition $\limsup_{R \downarrow \varepsilon} \mathbb{P}[R_1 > \varepsilon \mid R_0 = R] < \omega/\bar{T}$ suffices. □

Remark 5 (Interpretation and calibration of the threshold). By (15) and Lemma 1, the optimal threshold $\beta^\star$ is the level of the progress variable at which the expected remaining count under the optimal policy, $Q^\star(\beta^\star)$, equals that from a fresh restart, $C^\star$. The cutoff $\kappa\varepsilon_{\mathrm{opt}}$ of (18) plays the role of $\beta^\star$, although it is imposed on the score rather than on R, and no single global value can be right for every instance, since $\beta^\star$ depends on the instance through both $Q^\star$ and $C^\star$. Calibrating it exactly is circular, since $Q^\star$ and $C^\star$ are defined through $T^\star$. Replacing $T^\star$ by T_p for a constant $p \in \mathcal{A}$, computable from pilot runs by Theorem 2, gives a proxy free of this circularity, which over-estimates $C^\star$ and $Q^\star$ because constant policies form a subset of all policies and so $T^\star \leq T_p$ pointwise.

Remark 6 (Why randomise at all?). Whenever $1 \in \mathcal{A}$, Theorem 4 exhibits an optimal policy that is deterministic: restart at every state above $\beta^\star$. Randomisation confers no benefit in the idealised model, and the cap $p_{\max} < 1$ and the factor $2^{0.6\lfloor \nu_k/d \rfloor}$ in (20) bound the rate at which progress can be discarded. The situation changes once one recalls that the merit M_k is a lossy statistic of the state z^k: a policy measurable with respect to M_k is a restricted class in a partially observed problem, and there randomised memoryless policies can be strictly better than deterministic ones (Singh et al., 1994). This offers one reason for the stochastic form of (19); we have not shown that it applies here.

F THE DIFFUSION SURROGATE

As a heuristic guide, and as a concrete model in which Assumption 3 can be verified, we consider the one-dimensional surrogate

$$dR_t = -V'(R_t)\, dt + \sqrt{2D}\, dW_t, \qquad V(R) = \tfrac{1}{2}R^2, \tag{27}$$

where $D > 0$ is an effective diffusion coefficient and W a Brownian motion. The quadratic potential encodes mean reversion. With a state-dependent restart rate $\varrho(R)$ and restart law μ_R, the mean first passage time of (27) satisfies the backward equation $D\,T''(R) - R\,T'(R) - \varrho(R)\big(T(R) - C(R)\big) = -1$ with $C(R) = \int T\, d\mu_R$ (Evans & Majumdar, 2011a), so that restarting at R is beneficial precisely when $C(R) < T(R)$. We do not use the surrogate for quantitative predictions.

Lemma 3 (The diffusion surrogate is monotone)**.** *Let R_k evolve by the Euler–Maruyama discretisation of* (27)*, $R_{k+1} = R_k + \Delta t\, f(R_k) + \sqrt{2D\Delta t}\, w_k$, with w_k independent and identically distributed and $f = -V'$ Lipschitz with constant* $\mathrm{Lip}(f)$*. If $\Delta t\, \mathrm{Lip}(f) \leq 1$ then the transition kernel P satisfies Assumption 3. In particular this holds for $V(R) = \frac{1}{2}R^2$ and any $\Delta t \leq 1$.*

Proof. Couple two copies driven by the same noise, started at $R_0 \leq R_0'$. Then $R_1' - R_1 = (R_0' - R_0) + \Delta t\big(f(R_0') - f(R_0)\big) \geq \big(1 - \Delta t\, \mathrm{Lip}(f)\big)(R_0' - R_0) \geq 0$. A monotone pathwise coupling is equivalent to stochastic monotonicity of P. □

The lemma concerns a kernel driven by additive noise independent of the state. It says nothing about the kernel induced on R_k by one projected PDHG iteration (2), for which, as noted in Section 5, we know neither a proof nor a counterexample.

Remark 7 (Condition (17) is needed). In the surrogate at a fixed step size the one-step kernel does not concentrate into the target: for the kernel of Lemma 3, $\mathbb{P}[R_1 > \varepsilon \mid R_0 = R] \to \mathbb{P}[\sqrt{2D\Delta t}\, w_0 > \varepsilon\Delta t]$ as $R \downarrow \varepsilon$, a fixed positive constant. Whether $\beta^\star > \varepsilon$ then depends on the size of this constant relative to $\omega\, T^\star(\xi)/\bar{T}$, which is what (17) asks. The argument that restarts become useless near the target because $T(R) \to 0$ as $R \downarrow \varepsilon$ is valid for the continuous-time surrogate and false in discrete time, where $T^\star \geq 1$ outside the target.

G Further computational results

Hardware and tuning. All experiments were run on a 64-bit Windows machine with an 11th-generation Intel Core i5-1145G7 processor at 2.60 GHz and 16 GB of memory. The hyperparameters of Algorithm 1 were chosen by a grid search on the fourteen instances `30n20b8`, `aflow40b`, `air04`, `danoint`, `dfn-gwin-UUM`, `eil33-2`, `eilB101`, `lectsched-4-obj`, `macrophage`, `mcsched`, `ran16x16`, `reblock67`, `rmatr100-p10` and `rmine6`, which are among the 68 instances reported; the resulting single setting was then used for every instance.

Full results. Table 3 lists every instance of the benchmark. On `map18`, sPDLP reaches the time limit on all three seeds while PDLP terminates in 9 400 iterations; on `rail02`, PDLP reaches the time limit at 50 080 iterations without terminating, while sPDLP terminates on one of the three seeds, in 75 481 iterations, and reaches the time limit on the other two.

Table 2 summarises it.

Table 2: Summary of Table 3: sPDLP against PDLP at $\varepsilon_{\text{opt}} = 10^{-4}$, three seeds. Ratios are sPDLP over PDLP, so values below one favour sPDLP.

Among the 66 instances with a ratio	Iterations	Time
ratio < 0.5	1	2
ratio in $[0.5, 1)$	26	31
ratio in $[1, 1.5)$	27	14
ratio in $[1.5, 2)$	5	7
ratio ≥ 2	7	12
Geometric-mean ratio	1.074	1.078

Table 3: Full results on the Mittelmann benchmark at $\varepsilon_{\text{opt}} = 10^{-4}$: sPDLP (Algorithm 1, three seeds) against PDLP with its default adaptive normalised-duality-gap restart. Ratios are geometric means, over the seeds that reached the tolerance on both sides, of the sPDLP figure divided by the PDLP figure; values below one favour sPDLP. n and m are the numbers of variables and constraints after presolve.

Instance	n	m	Iter. ratio	Time ratio
`30n20b8`	18,380	576	0.711	1.189
`aflow40b`	2,728	1,442	2.237	1.108
`air04`	8,904	823	1.311	3.943
`app1-2`	26,871	53,467	0.045	0.078
`bab5`	21,600	4,964	0.918	1.483
`beasleyC3`	2,500	1,750	0.847	1.780
`bienst2`	505	576	1.121	1.038
`binkar10_1`	2,298	1,026	0.771	1.913
`bley_xl1`	5,831	175,620	1.051	1.184
`buildingenergy`	154,978	277,594	0.668	0.754
`csched010`	1,758	351	1.111	1.846
`danoint`	521	664	0.966	4.022

continued on next page

Instance	n	m	Iter. ratio	Time ratio
datt256	262,144	11,077	1.221	2.205
dfn-gwin-UUM	938	158	0.612	3.314
eil33-2	4,516	32	1.622	2.516
eilB101	2,818	100	0.912	2.678
ex10	17,680	69,608	0.937	2.444
glass4	322	396	4.421	1.755
gmu-35-40	1,205	424	1.344	2.193
iis-100-0-cov	100	3,831	1.121	1.174
iis-bupa-cov	345	4,803	1.123	1.494
iis-pima-cov	768	7,201	1.206	1.672
irish-e	61,728	104,259	0.939	0.972
lectsched-4-obj	7,901	14,163	1.295	2.656
macrophage	2,260	3,164	1.528	2.361
map18	164,547	328,818	time limit	time limit
mcsched	1,747	2,107	2.575	2.388
mik-250-1-100-1	251	151	0.557	1.333
mine-166-5	830	8,429	2.246	1.121
mine-90-10	900	6,270	1.138	0.877
n3div36	22,120	4,484	1.102	0.902
neos-1109824	1,520	28,979	2.007	0.914
neos-1337307	2,840	5,687	0.703	0.557
neos-1396125	1,161	1,494	1.284	0.621
neos-1601936	4,446	3,131	0.925	0.730
neos-686190	3,660	3,664	2.466	0.672
neos-849702	1,737	1,041	1.253	0.575
neos-916792	1,474	1,909	2.712	1.439
neos-934278	23,123	11,495	0.818	0.532
neos13	1,827	20,852	0.844	0.607
neos18	3,312	11,402	0.738	0.701
net12	14,115	14,021	0.855	0.702
netdiversion	129,180	119,589	1.109	0.686
newdano	505	576	1.121	0.573
ns1208400	2,883	4,289	1.312	0.726
ns1688347	2,685	4,191	0.796	0.548
ns1830653	1,629	2,932	0.720	0.771
opm2-z7-s2	2,023	31,798	1.591	0.829
qap15	22,275	6,330	1.085	0.654
rail02	270,869	95,791	time limit	time limit
rail507	63,019	509	0.876	0.629
ran16x16	512	288	1.172	0.440
reblock67	670	2,523	0.931	1.080
rmatr100-p10	7,359	7,260	0.526	0.689
rmatr100-p5	8,784	8,685	0.847	0.678
rmine6	1,096	7,078	1.204	1.733
rocII-4-11	9,234	21,738	0.713	0.728
rococoC10-001000	3,117	1,293	1.012	0.767
satellites1-25	9,013	5,996	1.103	0.698
sp98ic	10,894	825	1.711	1.294
sp98ir	1,680	1,531	0.923	1.121
stp3d	204,880	159,488	1.690	1.489
support10	14,770	165,684	1.451	0.827
tanglegram2	4,714	8,980	1.120	0.782
timtab1	397	171	1.442	0.517
unitcal_7	25,755	48,939	0.929	0.779
vpphard	51,471	47,280	1.375	1.941
zib54-UUE	5,150	1,809	1.046	3.152
Geometric mean over the 66 instances with a ratio			1.074	1.078

G.1 THE CONTROLLED TEST ON `DANOINT`

The fifteen restart-free counts behind the test of Section 7 are

$$N = 172,\ 170,\ 172,\ 174,\ 170,\ 171,\ 176,\ 170,\ 169,\ 169,\ 175,\ 170,\ 172,\ 173,\ 171.$$

Table 4 gives the plug-in curve of (12) computed from them, together with the mean hitting time of independent runs of restarted PDHG at four values of p. The renewal formula predicts a monotone increase over the whole range, with its minimum at $p = 0$. The independent runs agree in direction: at $p = 0.0005$ the difference from the control is not significant and at $p = 0.001$ it is borderline at this sample size, while at $p = 0.01$ the slowdown is plain, though smaller than the formula predicts. A likely reason for the gap is that coordinates without two finite bounds keep their pre-restart values, so post-restart states are not independent draws from μ and retain part of the progress made; in a replay experiment the actual post-restart states reached the target in 154.8 iterations on average against 171.6 for fresh draws from μ. Here μ denotes the calibration law used to generate restart-free initial states: each primal coordinate with two finite bounds is sampled independently and uniformly over its interval, coordinates without two finite bounds are set to 0, and the dual variable is initialized at 0 (followed by projection). The observed means at $p \in \{0, 5 \times 10^{-4}, 10^{-3}\}$ are based on 15 independent runs each, while the result at $p = 10^{-2}$ uses 50 independent runs.

Table 4: `danoint`: the plug-in curve $\hat{T}_p$ of (12) from fifteen restart-free counts, against the mean hitting time of independent runs of PDHG restarted at constant probability p, given as mean $\pm$ standard error with a 95 per cent confidence interval.

p	$\hat{\phi}(p)$	Predicted $\hat{T}_p$	Observed mean $\pm$ s.e.	95% interval
0	1.0000	171.6	174.20 ± 1.28	$[171.69, 176.71]$
0.0005	0.9178	179.1	172.27 ± 0.83	$[170.65, 173.88]$
0.001	0.8422	187.1	198.40 ± 12.30	$[174.29, 222.51]$
0.002	0.7093	204.6	–	–
0.005	0.4231	271.3	–	–
0.01	0.1783	456.3	361.20 ± 28.07	$[306.18, 416.22]$
0.02	0.0312	1519.3	–	–

G.2 PER-INSTANCE RATIOS AND CONVERGENCE PROFILES

Figure 2 shows the per-instance iteration ratios of Table 3 in increasing order, with the range across the three seeds. The variability across seeds is small on most instances and substantial on a few.

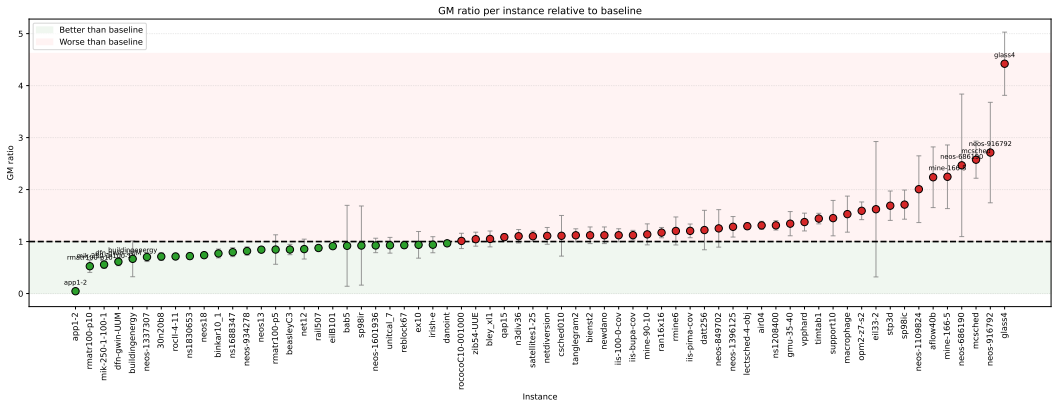


Figure 2: Per-instance geometric-mean iteration ratio of sPDLP to PDLP at $\varepsilon_{\text{opt}} = 10^{-4}$, instances ordered by ratio. The dashed line marks parity; points below it are instances on which sPDLP needs fewer iterations. Error bars show the range across the three seeds.

Figure 3 plots the convergence score s^k against the iteration count for four instances. On `app1-2` and `rmatr100-p10`, sPDLP reaches small scores much earlier than PDLP, in keeping with their

iteration ratios of 0.045 and 0.526. On `rococoC10-001000` the two methods behave alike, with a ratio of 1.012. On `glass4`, PDLP reaches the target far earlier, while sPDLP stays at a larger score for many more iterations, in keeping with its ratio of 4.421.

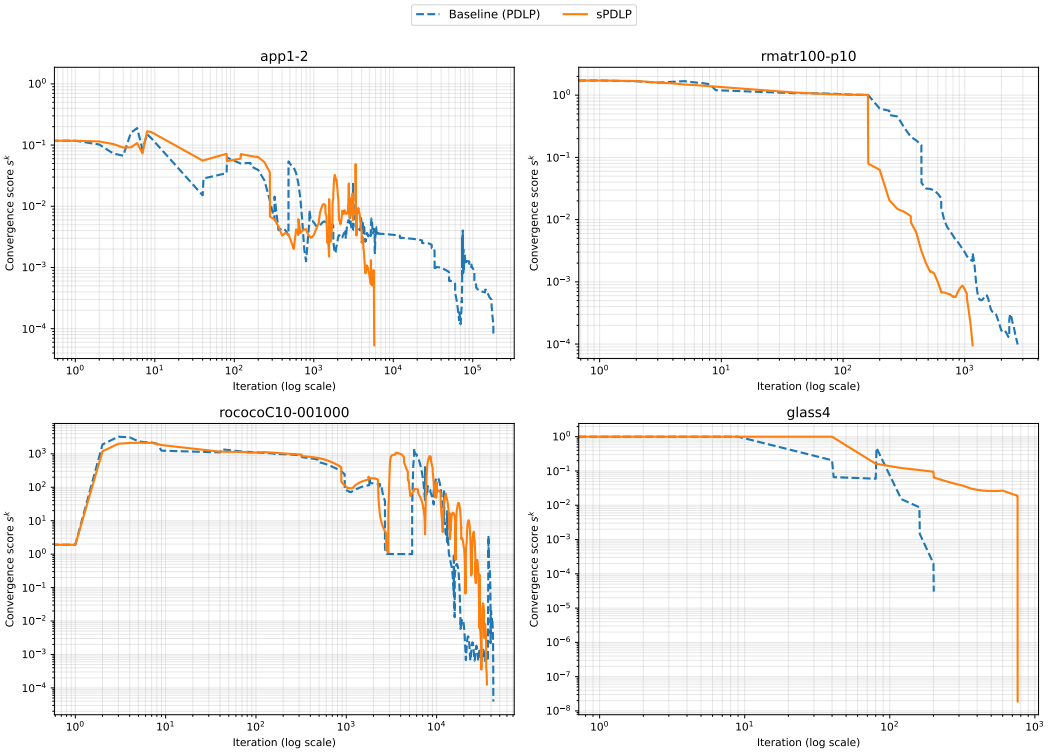


Figure 3: Convergence score s^k against iteration count on four instances, both axes logarithmic. Dashed blue curves are PDLP and solid orange curves sPDLP.

G.3 Performance by instance size and sparsity

We ask whether the size or the sparsity of an instance helps to explain the relative performance of sPDLP. For each of the 66 instances with a ratio we take its iteration ratio, the geometric mean over seeds of the sPDLP count divided by the PDLP count. Instances are grouped into terciles by the number of variables and by the density of the constraint matrix, the number of its nonzeros divided by the number of its entries; ties in density produce unequal tercile sizes. Table 5 shows a tendency towards more frequent improvements on larger and sparser instances: the large-and-sparse group has a win rate of 56.3 per cent against 33.3 per cent for the small-and-dense group. The geometric mean of the large-and-sparse group is sensitive to `app1-2`, whose ratio is 0.045; removing it moves the group's geometric mean from 0.836 to 1.016, while the win rate falls only from 56.3 to 53.3 per cent. Neither size nor sparsity yields a reliable rule for selecting instances, and the evidence does not establish an advantage in this regime.

Table 5: Iteration performance by structural subgroup. The geometric-mean ratio is taken over the iteration ratios of the group; values below one favour sPDLP.

Subgroup	Instances	Win rate	Geometric-mean ratio
Small	22	27.3%	1.236
Large	22	54.5%	0.891
Dense	23	34.8%	1.170
Sparse	22	54.5%	0.850
Small and dense	15	33.3%	1.181
Large and sparse	16	56.3%	0.836
excluding `app1-2`	15	53.3%	1.016