Randomized Feasibility Methods for Constrained Optimization with Adaptive Step Sizes

Abhishek Chakraborty and Angelia Nedić

School of Electrical, Computer, and Energy Engineering
Arizona State University

ICML 2026

Problem Statement

Consider the constrained optimization problem

$$\min_{x} f(x)$$
$$\text{subject to } x \in X \cap Y$$

Problem Statement

Consider the constrained optimization problem

$$\min_{x} f(x)$$
$$\text{subject to } x \in X \cap Y$$

- set $Y \subseteq \mathbb{R}^n$ is closed, convex, and easy to project on (e.g., ball or box constraint)

Problem Statement

Consider the constrained optimization problem

$$\min_x f(x)$$
$$\text{subject to } x \in X \cap Y$$

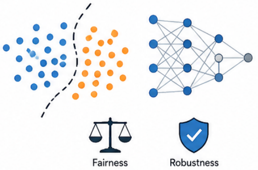
- set $Y \subseteq \mathbb{R}^n$ is closed, convex, and easy to project on (e.g., ball or box constraint)
- Set $X \subseteq \mathbb{R}^n$ is not easy to project on and is represented as

$$X := \cap_{i=1}^m \{x \in \mathbb{R}^n \mid g_i(x) \leq 0\}$$

- $g_i : \mathbb{R}^n \to \mathbb{R}$ for all $i = 1, \ldots, m$ are convex, and m can be a **large number**.

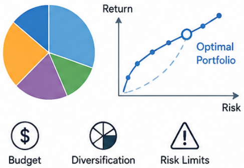
Motivating Applications

Fair & Robust Machine Learning



Optimize model performance while enforcing fairness and robustness constraints.

Portfolio Optimization

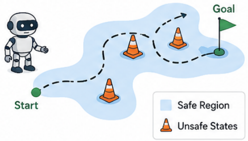


Maximize return subject to budget, risk, and regulatory constraints.

Energy Systems & Power Networks

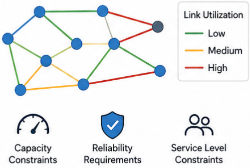
Optimize operation while satisfying capacity, reliability, and safety constraints.

Safe Reinforcement Learning



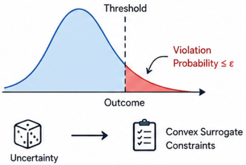
Learn policies that maximize reward while satisfying safety constraints.

Network Capacity & Reliability



Ensure reliable and efficient network operation under capacity constraints.

Chance-Constrained Optimization



Handle uncertainty via convex approximations of chance constraints.

Related Works and Motivation

Random Projections

- Sample a constraint and project onto it.
- Projection may be costly or unavailable.
- Can handle one constraint at a time.

Related Works and Motivation

Random Projections

- Sample a constraint and project onto it.
- Projection may be costly or unavailable.
- Can handle one constraint at a time.

Penalty / Interior-Point / ADMM (Projection free)

- Uses penalties or augmented Lagrangians.
- Requires tuning or solving subproblems.
- Handles all constraints jointly.

Primal-Dual / Frank–Wolfe (Projection free)

- Requires solving an LMO/subproblem.
- Scalability to infinite constraints is unclear.
- Handles all constraints jointly.

Related Works and Motivation

Random Projections

- Sample a constraint and project onto it.
- Projection may be costly or unavailable.
- Can handle one constraint at a time.

Penalty / Interior-Point / ADMM (Projection free)

- Uses penalties or augmented Lagrangians.
- Requires tuning or solving subproblems.
- Handles all constraints jointly.

Primal-Dual / Frank–Wolfe (Projection free)

- Requires solving an LMO/subproblem.
- Scalability to infinite constraints is unclear.
- Handles all constraints jointly.

Randomized Feasibility Updates (Projection free)

- Subsample constraints for feasibility updates.
- Projection oracle not required.
- Scales for large/infinite constraints.

Related Works and Motivation

Random Projections

- Sample a constraint and project onto it.
- Projection may be costly or unavailable.
- Can handle one constraint at a time.

Primal-Dual / Frank–Wolfe (Projection free)

- Requires solving an LMO/subproblem.
- Scalability to infinite constraints is unclear.
- Handles all constraints jointly.

Penalty / Interior-Point / ADMM (Projection free)

- Uses penalties or augmented Lagrangians.
- Requires tuning or solving subproblems.
- Handles all constraints jointly.

Randomized Feasibility Updates (Projection free)

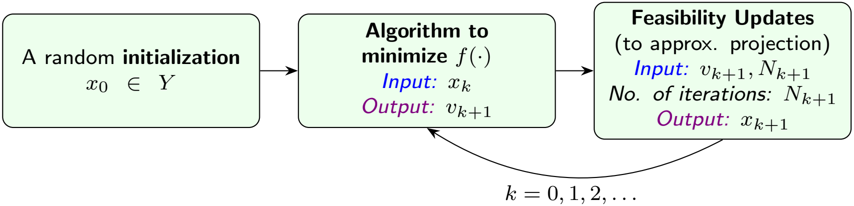
- Subsample constraints for feasibility updates.
- Projection oracle not required.
- Scales for large/infinite constraints.

Motivation

Combine **randomized feasibility updates** with **adaptive step-size schemes** to:

- Achieve exponential convergence (up to a tolerance) for strongly convex objective function.
- Develop a fully parameter-free and line-search-free algorithm for convex optimization.

Our Algorithmic Overview



Optimization Algorithms

- Gradient Method
- Distance over Weighted (sub)Gradients[a] (DoWS)
- Tamed-Distance over Weighted (sub)Gradients (T-DoWS)

[a] Ahmed Khaled, Konstantin Mishchenko, and Chi Jin. "DoWG unleashed: An efficient universal parameter-free gradient descent method". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 6748–6769.

Randomized Feasibility Updates

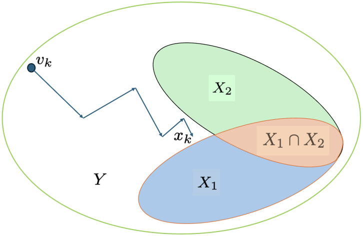
- Inner loop consists of N_k feasibility iterations.
- N_k is specified at every outer iteration $k \geq 1$.
- Sequential or parallel updates are possible.
- We consider sequential updates[a].

[a] Angelia Nedić and Ion Necoara. "Random minibatch subgradient algorithms for convex problems with functional constraints". In: *Applied Mathematics & Optimization* 80.3 (2019), pp. 801–833.

Randomized Feasibility Updates (Sequential Implementation)[1]

At iteration k of the Gradient Method

- **Require:** v_k, N_k constant step size $0 < \beta < 2$
- **Set:** $z_k^0 = v_k$
- **for** $i = 1, \ldots, N_k$
  - **Sample:** Choose a random index $\omega_k^i \in \{1, \ldots, m\}$
  - **Compute:** Subgradient d_k^i of $g^+_{\omega_k^i}(z)$ at $z = z_k^{i-1}$
  - **Update:**
    $$z_k^i = \Pi_Y \left[z_k^{i-1} - \beta \, \frac{g^+_{\omega_k^i}(z_k^{i-1})}{\|d_k^i\|^2} \, d_k^i \right]$$
- end for
- **Output** $x_k = z_k^{N_k}$



[1] Boris T. Polyak. "Minimization of unsmooth functionals". In: *USSR Computational Mathematics and Mathematical Physics* 9.3 (1969), pp. 14–29.

Gradient Method for Strongly Convex Objective Function

Assumptions: The function f is L-Lipschitz smooth and μ-strongly convex.

$$f(y) - f(x) \leq \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \text{ for all } x, y \in Y.$$

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2 \text{ for all } x, y \in Y.$$

Gradient Method for Strongly Convex Objective Function

Assumptions: The function f is L-Lipschitz smooth and μ-strongly convex.

$$f(y) - f(x) \leq \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \text{ for all } x, y \in Y.$$
$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2 \text{ for all } x, y \in Y.$$

Gradient Method with Randomized Feasibility:

$$v_{k+1} = \Pi_Y[x_k - \alpha_k \nabla f(x_k)]$$
$$x_{k+1} = \texttt{Randomized Feasibility}(v_{k+1}, N_{k+1})$$

Gradient Method for Strongly Convex Objective Function

Assumptions: The function f is L-Lipschitz smooth and μ-strongly convex.

$$f(y) - f(x) \leq \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \text{ for all } x, y \in Y.$$
$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2 \text{ for all } x, y \in Y.$$

Gradient Method with Randomized Feasibility:

$$v_{k+1} = \Pi_Y[x_k - \alpha_k \nabla f(x_k)]$$
$$x_{k+1} = \texttt{Randomized Feasibility}(v_{k+1}, N_{k+1})$$

Question: How to choose α_k?

Main Result: Strongly Convex Case

Theorem

- f is L-smooth and μ-strongly convex
- Sufficient sample selection:

$$\mathbb{E}\big[(1-q)^{N_k/2}\big] \leq \frac{\epsilon}{(B_0 + \|x^*\|)M_f}.$$

- Adaptive step size

$$\alpha_k = \min\left\{\frac{1}{2(L-\mu)}, \frac{1}{L}, \frac{\epsilon}{2\|\nabla f(x_k)\|^2}\right\}.$$

- Algorithm 1 produces an iterate $\bar{x}_k$ satisfying $\mathbb{E}[\,|f(\bar{x}_k) - f(x^*)|\,] \leq \epsilon$ after $k = O\big(\log \frac{1}{\epsilon}\big)$ iterations.

If $N_k = N$ for all $k \geq 1$, then sample selection to reach ϵ tolerance is

$$N \geq \frac{2\ln\left(\frac{(B_0+\|x^*\|)M_f}{\epsilon}\right)}{\ln\left(\frac{1}{1-q}\right)}.$$

Parameter-Free Step Sizes for Convex Nonsmooth Optimization

Convex Nonsmooth Setting

$$f(y) - f(x) \geq \langle s_f(x), y - x \rangle, \qquad s_f(x) \in \partial f(x).$$

Convex Nonsmooth Setting

$$f(y) - f(x) \geq \langle s_f(x), y - x \rangle, \qquad s_f(x) \in \partial f(x).$$

Algorithm Ingredients

- Adaptive radius estimate: $\bar{r}_k = \max\{\|x_k - x_0\|, \bar{r}_{k-1}\}$
- (Sub)gradient accumulation: $p_k = p_{k-1} + \bar{r}_k^2 \|s_f(x_k)\|^2$

Convex Nonsmooth Setting

$$f(y) - f(x) \geq \langle s_f(x), y - x \rangle, \qquad s_f(x) \in \partial f(x).$$

Algorithm Ingredients

- Adaptive radius estimate: $\bar{r}_k = \max\{\|x_k - x_0\|, \bar{r}_{k-1}\}$
- (Sub)gradient accumulation: $p_k = p_{k-1} + \bar{r}_k^2 \|s_f(x_k)\|^2$
- (Sub)gradient method: $v_{k+1} = \Pi_Y \left[x_k - \alpha_k s_f(x_k)\right]$
- Feasibility Update: $x_{k+1} = \texttt{Randomized Feasibility}(v_{k+1}, N_{k+1})$

Parameter-Free Step Sizes for Convex Nonsmooth Optimization

Convex Nonsmooth Setting

$$f(y) - f(x) \geq \langle s_f(x), y - x \rangle, \qquad s_f(x) \in \partial f(x).$$

Algorithm Ingredients

- Adaptive radius estimate: $\bar{r}_k = \max\{\|x_k - x_0\|, \bar{r}_{k-1}\}$
- (Sub)gradient accumulation: $p_k = p_{k-1} + \bar{r}_k^2 \|s_f(x_k)\|^2$
- (Sub)gradient method: $v_{k+1} = \Pi_Y \left[x_k - \alpha_k s_f(x_k)\right]$
- Feasibility Update: $x_{k+1} = \texttt{Randomized Feasibility}(v_{k+1}, N_{k+1})$

DoWS + Randomized Feasibility

$$\alpha_k = \frac{\bar{r}_k^2}{\sqrt{p_k}}$$

Assumption

- Y is bounded.

T-DoWS + Randomized Feasibility

$$\alpha_k = \begin{cases} \dfrac{\bar{r}_k^2}{2\sqrt{p_k}\ln\left(\frac{e p_k}{p_1}\right)} & p_0 = 0, \\ \dfrac{\bar{r}_k^2}{\sqrt{2p_k}\ln\left(\frac{e p_k}{p_0}\right)} & p_0 > 0. \end{cases}$$

Assumption

- Level set $\{y \in Y \mid f(y) \leq f^*\}$ is bounded

Convergence Rates of the Parameter-Free Methods

Let **RF** denotes Randomized Feasibility Algorithm.

Theorem

For any $T \geq 1$, define $\tau = \arg\min_{1 \leq k \leq T} \frac{\bar{r}_{k+1}^2}{\sum_{i=1}^{k} \bar{r}_i^2}$ and $\bar{x}_\tau = \frac{\sum_{k=1}^{\tau} \bar{r}_k^2 x_k}{\sum_{i=1}^{\tau} \bar{r}_i^2}$. Then

$$\mathbb{E}\left[|f(\bar{x}_\tau) - f^*|\right] \leq \max\{A_1(T), \min\{A_2(T), A_3(T)\}\}.$$

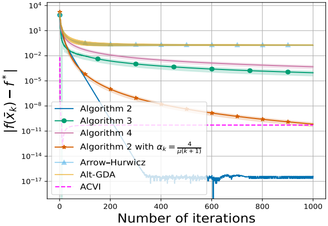
Term	**DoWS + RF**	**T-DoWS + RF**
$A_1(T)$	$O(1/\sqrt{T})$	$O(\ln(T)/\sqrt{T})$
$A_2(T)$	$O\left(\max_{1\leq k\leq T} \mathbb{E}\left[(1-q)^{N_k/2}\right]\right)$	$O\left(\max_{1\leq k\leq T} \mathbb{E}\left[(1-q)^{N_k/2}\right]\right)$
$A_3(T)$	$O\left(\frac{\sum_{k=1}^{T} \mathbb{E}\left[(1-q)^{N_k/2}\right]}{T}\right)$	$O\left(\frac{\sum_{k=1}^{T} \mathbb{E}\left[(1-q)^{N_k/2}\right]}{T}\right)$

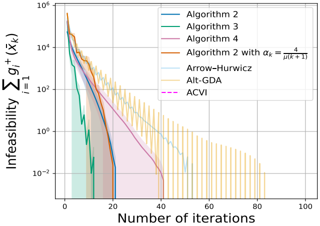
- **E.g.:** If $N_k = \lceil k^{\frac{1}{p}} \rceil$ and $p > 0$, then $A_3(T) = O(1/T)$ for both algorithms.

Simulation (Quadratically Constrained Quadratic Programming)

$$\min_{x\in[-10,10]^{10}} \langle x, Ax\rangle + \langle b, x\rangle$$
$$\text{s.t. } \langle x, C_i x\rangle + \langle u_i, x\rangle - e_i \le 0 \quad \text{for } i = 1,\dots,1000.$$

Strongly convex objective function with known f^*:

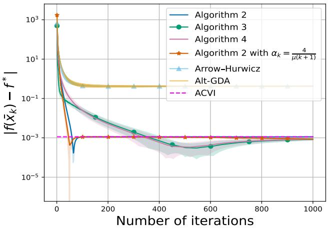


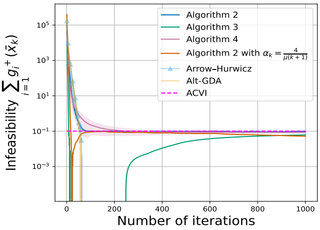


Algorithm 2: Gradient Method + Randomized Feasibility
Algorithm 3: DoWS + Randomized Feasibility
Algorithm 4: T-DoWS + Randomized Feasibility

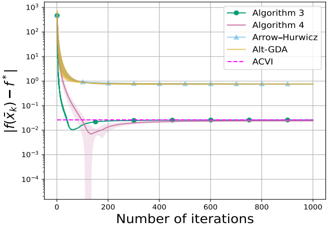
Simulation (Quadratically Constrained Quadratic Programming (QCQP))

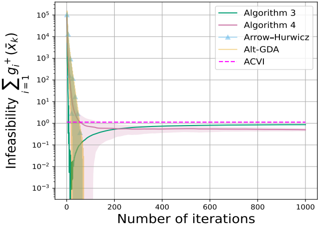
Strongly convex objective function with unknown f^*:



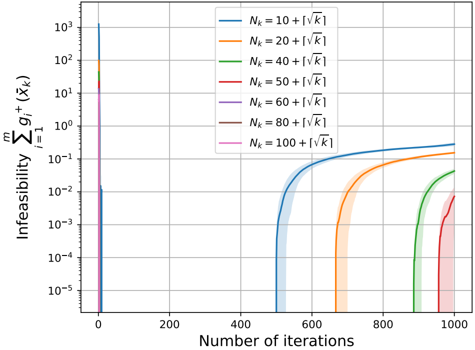


Convex objective function with unknown f^*:





QCQP: Effect of Feasibility Sample Size N_k



Setup

- Convex objective
- DoWS + Randomized Feasibility

Observation

- Larger N_k yields faster infeasibility reduction.
- The solution lies near the constraint boundary.
- Small N_k may drive infeasibility close to zero initially, but violations can reappear later.
- Too few sampled constraints may miss active constraints.

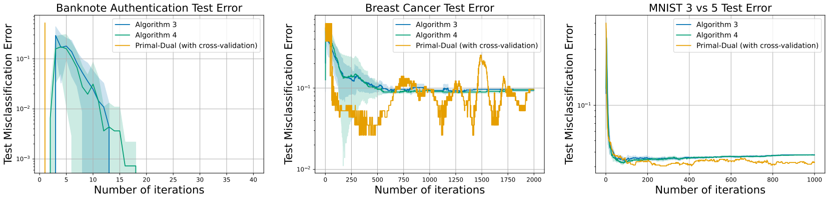
Takeaway

Increasing N_k improves feasibility accuracy at the cost of additional computation.

Simulation (Support Vector Machine)

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{subject to} \quad 1 - \xi_i - y_i(w^\top z_i + b) \leq 0\ , \quad \xi_i \geq 0, \quad i = 1, \ldots, m.$$



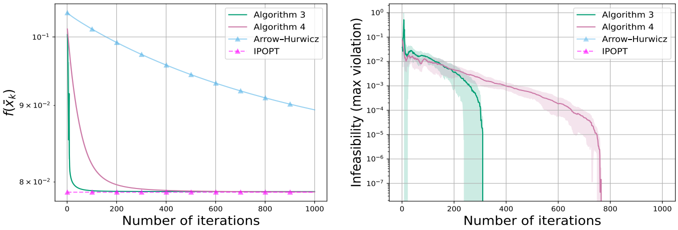
Algorithm 3: DoWS + Randomized Feasibility
Algorithm 4: T-DoWS + Randomized Feasibility

Simulation (Logistic Regression with Group Fairness Constraints)

$$\min_{x\in\mathbb{R}^n} \quad f(x) := \frac{1}{\widehat{N}} \sum_{j=1}^{\widehat{N}} \ln(1+\exp(-y_j \langle a_j, x\rangle))$$

$$\text{s.t.} \quad x \in \{x \in \mathbb{R}^n \mid \ell \le x \le u\} \bigcap \left\{ \bigcap_{i=1}^{m} \left\{ x \in \mathbb{R}^n \;\middle|\; \left|c_i^\top x\right| \le \tau_i \right\} \right\}$$

- See the manuscript for construction details



(a) Objective function value

(b) Maximum Infeasibility violation

Algorithm 3: DoWS + Randomized Feasibility
Algorithm 4: T-DoWS + Randomized Feasibility

More results and insights can be found in our work.

Check it out below!

Thank You!

Scan the QR code to read the full paper on arXiv.

Acknowledgements: This work is partially supported by the NSF award CIF 2134256.