First-Order Methods for Distributionally Robust Mixed-Integer Optimization

Hubert Villuendas, Mathieu Besançon & Jérôme Malick
: hubert.villuendas@univ-grenoble.fr

University Grenoble Alpes, France

February 26th, 2026

ROADEF, Tours

Université Grenoble Alpes

Outline

From uncertainty to robustness

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad f(z, \xi) \qquad\qquad \mathcal{Z} \subset \mathbf{R}^n \times \mathbf{Z}^{d-n}$$

- $\xi \in \Xi$ possible problem data

From uncertainty to robustness

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad f(z, \xi) \qquad\qquad \mathcal{Z} \subset \mathbf{R}^n \times \mathbf{Z}^{d-n}$$

- $\xi \in \Xi$ possible problem data
- Unknown: $\mathbf{P}$ distribution on Ξ

From uncertainty to robustness

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad \mathbf{E}_{\xi \sim \mathbf{P}} \left[f(z, \xi) \right] \qquad \mathcal{Z} \subset \mathbf{R}^n \times \mathbf{Z}^{d-n}$$

- $\xi \in \Xi$ possible problem data
- Unknown: $\mathbf{P}$ distribution on Ξ
- Known: samples $\left\{ \widehat{\xi}_1, \ldots, \widehat{\xi}_N \right\} \subset \Xi$

From uncertainty to robustness

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad \mathbf{E}_{\xi \sim \mathbf{P}} \left[f(z, \xi) \right] \qquad \mathcal{Z} \subset \mathbf{R}^n \times \mathbf{Z}^{d-n}$$

- $\xi \in \Xi$ possible problem data
- Unknown: $\mathbf{P}$ distribution on Ξ
- Known: samples $\left\{ \widehat{\xi}_1, \ldots, \widehat{\xi}_N \right\} \subset \Xi$
- Empirical distribution:

$$\widehat{\mathbf{P}}_N = \frac{1}{N} \sum_{k=1}^{N} \delta_{\widehat{\xi}_k}$$

From uncertainty to robustness

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad \mathbf{E}_{\xi \sim \mathbf{P}} \left[f(z, \xi) \right] \qquad \mathcal{Z} \subset \mathbf{R}^n \times \mathbf{Z}^{d-n}$$

- $\xi \in \Xi$ possible problem data
- Unknown: $\mathbf{P}$ distribution on Ξ
- Known: samples $\left\{ \widehat{\xi}_1, \ldots, \widehat{\xi}_N \right\} \subset \Xi$
- Empirical distribution:

$$\widehat{\mathbf{P}}_N = \frac{1}{N} \sum_{k=1}^{N} \delta_{\widehat{\xi}_k}$$

- Empirical Risk Minimization (ERM):

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad \left[\frac{1}{N} \sum_{k=1}^{N} f\left(z, \widehat{\xi}_k\right) = \mathbf{E}_{\xi \sim \widehat{\mathbf{P}}_N} \left[f(z, \xi) \right] \right]$$

From uncertainty to robustness

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad \mathbf{E}_{\xi \sim \mathbf{P}} \left[f(z, \xi) \right] \qquad \mathcal{Z} \subset \mathbf{R}^n \times \mathbf{Z}^{d-n}$$

- $\xi \in \Xi$ possible problem data
- Unknown: $\mathbf{P}$ distribution on Ξ
- Known: samples $\left\{ \widehat{\xi}_1, \ldots, \widehat{\xi}_N \right\} \subset \Xi$
- Empirical distribution:

$$\widehat{\mathbf{P}}_N = \frac{1}{N} \sum_{k=1}^{N} \delta_{\widehat{\xi}_k}$$

- Empirical Risk Minimization (ERM):

Subject to distributionnal shift!

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad \left[\frac{1}{N} \sum_{k=1}^{N} f\left(z, \widehat{\xi}_k\right) = \mathbf{E}_{\xi \sim \widehat{\mathbf{P}}_N} \left[f(z, \xi) \right] \right]$$

From uncertainty to robustness

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad \mathbf{E}_{\xi \sim \mathbf{P}} \left[f(z, \xi) \right] \qquad \mathcal{Z} \subset \mathbf{R}^n \times \mathbf{Z}^{d-n}$$

- $\xi \in \Xi$ possible problem data
- Unknown: $\mathbf{P}$ distribution on Ξ
- Known: samples $\left\{ \widehat{\xi}_1, \ldots, \widehat{\xi}_N \right\} \subset \Xi$
- Empirical distribution:

$$\widehat{\mathbf{P}}_N = \frac{1}{N} \sum_{k=1}^{N} \delta_{\widehat{\xi}_k}$$

- Distributionally Robust Optimization (DRO):

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad \sup_{\substack{\mathbf{Q} \in \text{Proba}(\Xi) \\ \mathbf{Q} \text{ close to } \widehat{\mathbf{P}}_N}} \mathbf{E}_{\zeta \sim \mathbf{Q}} \left[f(z, \zeta) \right]$$

Wasserstein distributionally robust optimization

Wasserstein Distributionally Robust Optimization (WDRO)
[*Esfahani and Kuhn, 2018*]

$$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \sup_{W(\mathbf{Q},\widehat{\mathbf{P}}_N)\leq\rho} \mathbf{E}_{\zeta\sim\mathbf{Q}}\left[f(z,\zeta)\right]$$

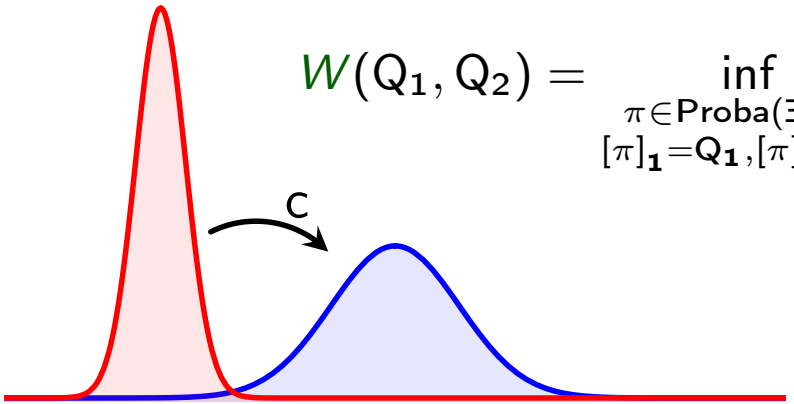
Wasserstein distributionally robust optimization

Wasserstein Distributionally Robust Optimization (WDRO)
[*Esfahani and Kuhn, 2018*]

$$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \sup_{W\left(\mathbb{Q},\widehat{\mathbb{P}}_N\right)\leq\rho} \mathbf{E}_{\zeta\sim\mathbb{Q}}\left[f(z,\zeta)\right]$$

Wasserstein distance:

$$W(\mathbb{Q}_1,\mathbb{Q}_2) = \inf_{\substack{\pi\in\mathbf{Proba}(\Xi\times\Xi)\\ [\pi]_1=\mathbb{Q}_1,[\pi]_2=\mathbb{Q}_2}} \int_{\Xi\times\Xi} c\left(\xi,\xi'\right)\mathrm{d}\pi\left(\xi,\xi'\right)$$

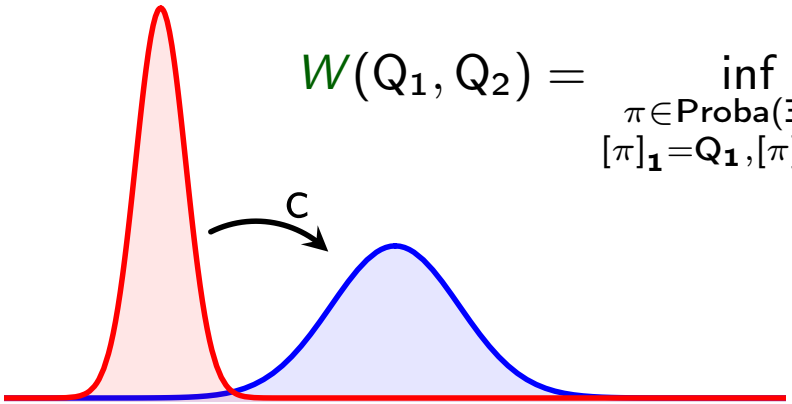


Wasserstein distributionally robust optimization

Wasserstein Distributionally Robust Optimization (WDRO)
[*Esfahani and Kuhn, 2018*]

$$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \sup_{W(\mathbb{Q},\widehat{\mathbb{P}}_N)\leq\rho} \mathbf{E}_{\zeta\sim\mathbb{Q}}\left[f(z,\zeta)\right]$$

Wasserstein distance:



$$W(\mathbb{Q}_1,\mathbb{Q}_2) = \inf_{\substack{\pi\in\mathbf{Proba}(\Xi\times\Xi)\\ [\pi]_1=\mathbb{Q}_1,[\pi]_2=\mathbb{Q}_2}} \int_{\Xi\times\Xi} c\left(\xi,\xi'\right)\mathrm{d}\pi\left(\xi,\xi'\right)$$

Existing work for DRO in Operations Research:

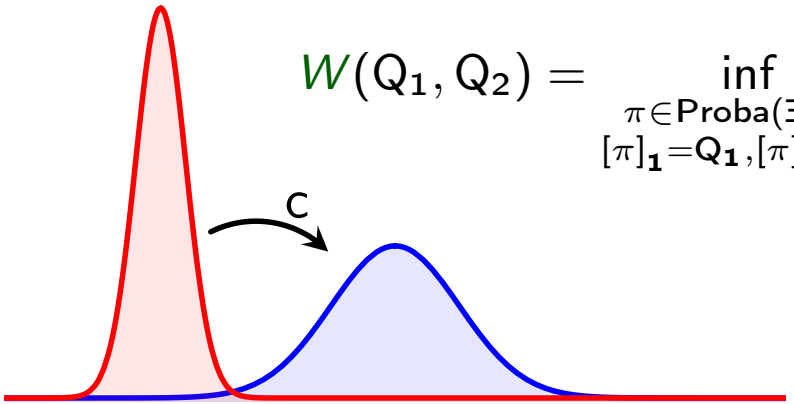
- Chance-constrained vehicle routing **[*Ghosal and Wiesemann, 2020*]**

Wasserstein distributionally robust optimization

Wasserstein Distributionally Robust Optimization (WDRO)
[*Esfahani and Kuhn, 2018*]

$$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \sup_{W(\mathbb{Q},\widehat{\mathbb{P}}_N)\leq\rho} \mathbf{E}_{\zeta\sim\mathbb{Q}}\left[f(z,\zeta)\right]$$

Wasserstein distance:



$$W(\mathbb{Q}_1,\mathbb{Q}_2) = \inf_{\substack{\pi\in\mathbf{Proba}(\Xi\times\Xi)\\ [\pi]_1=\mathbb{Q}_1,[\pi]_2=\mathbb{Q}_2}} \int_{\Xi\times\Xi} c\left(\xi,\xi'\right)\mathrm{d}\pi\left(\xi,\xi'\right)$$

Existing work for DRO in Operations Research:

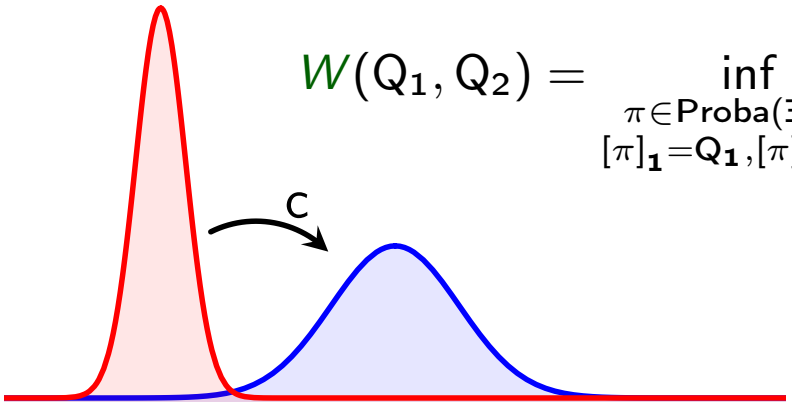
- Chance-constrained vehicle routing **[*Ghosal and Wiesemann, 2020*]**
- Surgery allocation **[*Chow et al., 2022*]**

Wasserstein distributionally robust optimization

Wasserstein Distributionally Robust Optimization (WDRO)
[*Esfahani and Kuhn, 2018*]

$$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \sup_{W(\mathbb{Q},\widehat{\mathbb{P}}_N)\leq\rho} \mathbf{E}_{\zeta\sim\mathbb{Q}}\left[f(z,\zeta)\right]$$

Wasserstein distance:



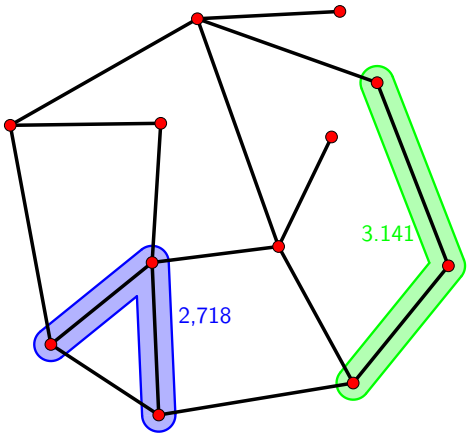
$$W(\mathbb{Q}_1,\mathbb{Q}_2) = \inf_{\substack{\pi\in\mathbf{Proba}(\Xi\times\Xi)\\ [\pi]_1=\mathbb{Q}_1,[\pi]_2=\mathbb{Q}_2}} \int_{\Xi\times\Xi} c\left(\xi,\xi'\right)\mathrm{d}\pi\left(\xi,\xi'\right)$$

Existing work for DRO in Operations Research:

- Chance-constrained vehicle routing **[*Ghosal and Wiesemann, 2020*]**
- Surgery allocation **[*Chow et al., 2022*]**
- Ressource allocation during a pandemic **[*Sun et al., 2023*]**

Running example — Quadratic Spanning Tree

$G = (V, E)$ on m edges, $\mathcal{C} \in \mathbf{R}_+^{m \times m}$ cost matrix.

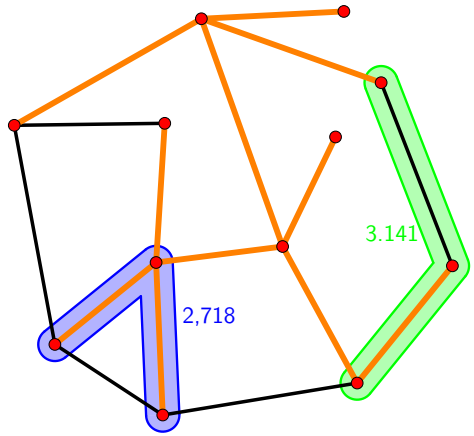


$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad z^\top \mathcal{C} z$$

$$z \in \{0,1\}^m,\ z_e = \begin{cases} \mathbf{1} & \text{if } e \in \mathscr{T} \\ 0 & \text{otherwise.} \end{cases}$$

Running example — Quadratic Spanning Tree

$G = (V, E)$ on m edges, $\mathcal{C} \in \mathbf{R}_+^{m \times m}$ cost matrix.

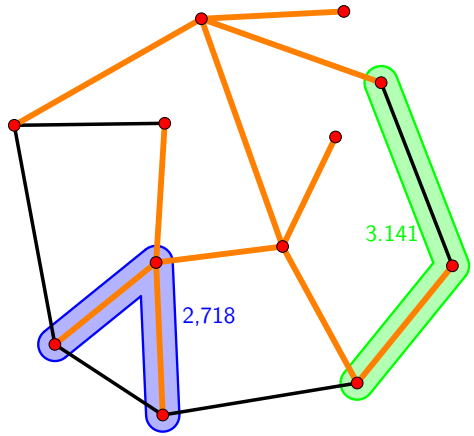


$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad z^\top \mathcal{C} z$$

$$z \in \{0,1\}^m,\ z_e = \begin{cases} \mathbf{1} & \text{if } e \in \mathscr{T} \\ 0 & \text{otherwise.} \end{cases}$$

Running example — Quadratic Spanning Tree

$G = (V, E)$ on m edges, $\mathcal{C} \in \mathbf{R}_+^{m \times m}$ cost matrix.

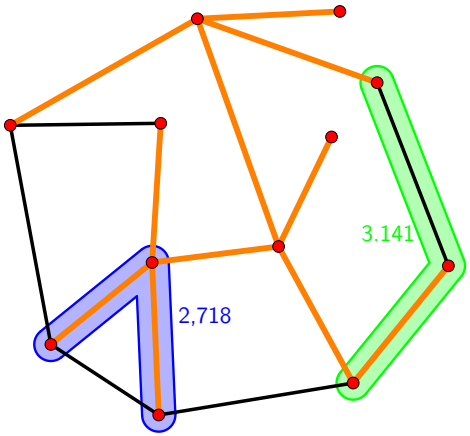


$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad z^\top \mathcal{C} z$$

$$z \in \{0,1\}^m,\ z_e = \begin{cases} \mathbf{1} & \text{if } e \in \mathscr{T} \\ 0 & \text{otherwise.} \end{cases}$$

Running example — Quadratic Spanning Tree

$G = (V, E)$ on m edges, $\mathcal{C} \in \mathbf{R}_+^{m \times m}$ cost matrix.



$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad z^\top \mathcal{C} z$$

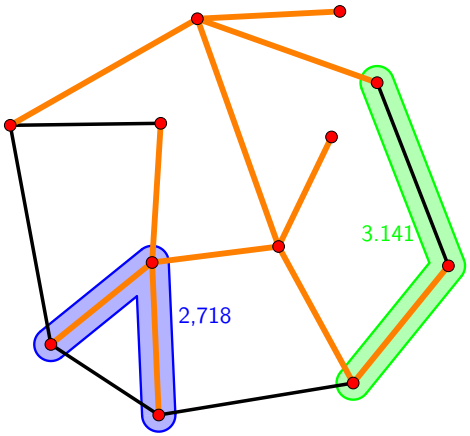
$$z \in \{0,1\}^m,\ z_e = \begin{cases} 1 & \text{if } e \in \mathscr{T} \\ 0 & \text{otherwise.} \end{cases}$$

Proposition

- *Quadratic* minimum spanning tree problem is $\mathscr{NP}$-hard [***Assad and Xu, 1992***]

Running example — Quadratic Spanning Tree

$G = (V, E)$ on m edges, $\mathcal{C} \in \mathbf{R}_+^{m \times m}$ cost matrix.



$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad z^\top \mathcal{C} z$$

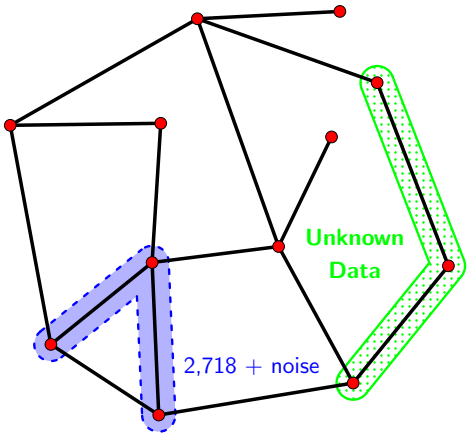
$$z \in \{0,1\}^m,\ z_e = \begin{cases} 1 & \text{if } e \in \mathscr{T} \\ 0 & \text{otherwise.} \end{cases}$$

Proposition

- *Quadratic* minimum spanning tree problem is $\mathscr{NP}$-hard **[*Assad and Xu, 1992*]**
- *Linear* minimum spanning tree problem is easy!
  Solved in $\mathcal{O}(m \log(m))$ by Kruskal algorithm **[*Schrijver, 2002*]**

Running example — Quadratic Spanning Tree

$G = (V, E)$ on m edges, $\mathcal{C} \in \mathbf{R}_+^{m \times m}$ cost matrix.



$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad z^\top \mathcal{C} z$$

$$z \in \{0,1\}^m,\ z_e = \begin{cases} 1 & \text{if } e \in \mathscr{T} \\ 0 & \text{otherwise.} \end{cases}$$

Proposition

- *Quadratic* minimum spanning tree problem is $\mathscr{NP}$-hard **[*Assad and Xu, 1992*]**
- *Linear* minimum spanning tree problem is easy!
  Solved in $\mathcal{O}(m \log(m))$ by Kruskal algorithm **[*Schrijver, 2002*]**

Outline

Regularisation for WDRO

$$\min_{z \in \mathcal{Z}} \quad \mathbf{E}_{\xi \sim \mathbf{P}} \left[f(z, \xi) \right]$$

$$\min_{z \in \mathcal{Z}} \quad \sup_{W\left(\mathbf{Q}, \widehat{\mathbf{P}}_N\right) \leq \rho} \quad \mathbf{E}_{\zeta \sim \mathbf{Q}} \left[f(z, \zeta) \right]$$

Regularisation for WDRO

$$\min_{z\in\mathcal{Z}} \quad \mathbf{E}_{\xi\sim\mathbf{P}}\left[f(z,\xi)\right]$$

$$\min_{z\in\mathcal{Z}} \sup_{W\left(\mathbf{Q},\widehat{\mathbf{P}}_N\right)\leq\rho} \mathbf{E}_{\zeta\sim\mathbf{Q}}\left[f(z,\zeta)\right]$$

$$= \min_{z\in\mathcal{Z}} \inf_{\lambda\in\mathbf{R}_+} \lambda\rho + \mathbf{E}_{\widehat{\xi}\sim\widehat{\mathbf{P}}_N}\left[\sup_{\zeta\in\Xi}\left\{f(z,\zeta) - \lambda c\left(\widehat{\xi},\zeta\right)\right\}\right]$$

[*Esfahani and Kuhn, 2018*]

Regularisation for WDRO

$$\min_{z\in\mathcal{Z}} \quad \mathbf{E}_{\xi\sim\mathbf{P}}\left[f(z,\xi)\right]$$

$$\min_{z\in\mathcal{Z}} \sup_{W\left(\mathbf{Q},\widehat{\mathbf{P}}_N\right)\leq\rho} \mathbf{E}_{\zeta\sim\mathbf{Q}}\left[f(z,\zeta)\right]$$

$$= \min_{z\in\mathcal{Z}} \inf_{\lambda\in\mathbf{R}_+} \lambda\rho + \mathbf{E}_{\widehat{\xi}\sim\widehat{\mathbf{P}}_N}\left[\sup_{\zeta\in\Xi}\left\{f(z,\zeta) - \lambda c\left(\widehat{\xi},\zeta\right)\right\}\right]$$

[*Esfahani and Kuhn, 2018*]

$$\underset{\varepsilon\to 0}{\simeq} \min_{z\in\mathcal{Z}} \inf_{\lambda\in\mathbf{R}_+} \lambda\rho + \varepsilon\mathbf{E}_{\widehat{\xi}\sim\widehat{\mathbf{P}}_N}\left[\log\left(\mathbf{E}_{\zeta\sim\mathcal{N}\left(\widehat{\xi},\sigma^2\mathbf{I}\right)}\left[\exp\left(\frac{f(z,\zeta) - \lambda c\left(\widehat{\xi},\zeta\right)}{\varepsilon}\right)\right]\right)\right]$$

[*Azizian et al., 2023*]

Regularisation for WDRO

$$\min_{z\in\mathcal{Z}} \quad \mathbf{E}_{\xi\sim\mathbf{P}}\left[f(z,\xi)\right]$$

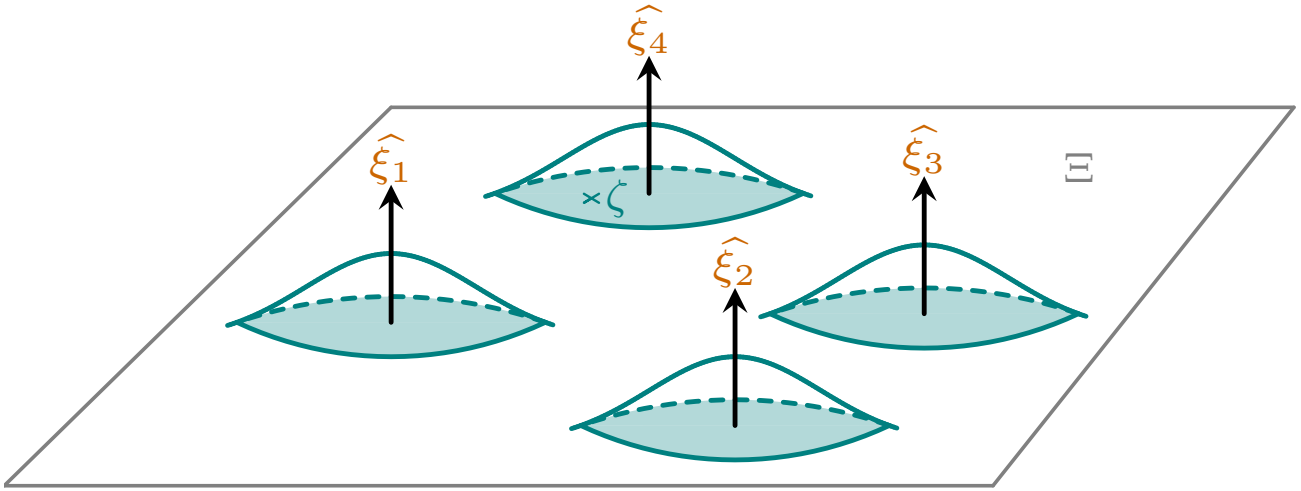
$$\min_{z\in\mathcal{Z}} \sup_{W\left(\mathbf{Q},\widehat{\mathbf{P}}_N\right)\leq\rho} \mathbf{E}_{\zeta\sim\mathbf{Q}}\left[f(z,\zeta)\right]$$

$$= \min_{z\in\mathcal{Z}} \inf_{\lambda\in\mathbf{R}_+} \lambda\rho + \mathbf{E}_{\widehat{\xi}\sim\widehat{\mathbf{P}}_N}\left[\sup_{\zeta\in\Xi}\left\{f(z,\zeta) - \lambda c\left(\widehat{\xi},\zeta\right)\right\}\right]$$

[*Esfahani and Kuhn, 2018*]

$$\underset{\varepsilon\to\mathbf{0}}{\simeq} \min_{z\in\mathcal{Z}} \inf_{\lambda\in\mathbf{R}_+} \lambda\rho + \varepsilon\mathbf{E}_{\widehat{\xi}\sim\widehat{\mathbf{P}}_N}\left[\log\left(\mathbf{E}_{\zeta\sim\mathcal{N}\left(\widehat{\xi},\sigma^2\mathbf{I}\right)}\left[\exp\left(\frac{f(z,\zeta)-\lambda c\left(\widehat{\xi},\zeta\right)}{\varepsilon}\right)\right]\right)\right]$$

[*Azizian et al., 2023*]



WDRO Objective function

$$F(z,\lambda) \overset{\text{def}}{=} \lambda\rho + \varepsilon \mathbf{E}_{\widehat{\xi}\sim\widehat{\mathsf{P}}_N}\left[\log\left(\mathbf{E}_{\zeta\sim\mathcal{N}(\widehat{\xi},\sigma^2\mathbf{I})}\left[\exp\left(\frac{f(z,\zeta)-\lambda c\left(\widehat{\xi},\zeta\right)}{\varepsilon}\right)\right]\right)\right]$$

WDRO Objective function

$$F(z,\lambda) \overset{\text{def}}{=} \lambda\rho + \varepsilon \mathbf{E}_{\widehat{\xi}\sim\widehat{\mathsf{P}}_N}\left[\log\left(\mathbf{E}_{\zeta\sim\mathcal{N}(\widehat{\xi},\sigma^2\mathbf{I})}\left[h_{\widehat{\xi}}(z,\lambda;\zeta)\right]\right)\right]$$

WDRO Objective function

$$F(z,\lambda) \overset{\text{def}}{=} \lambda\rho + \varepsilon \mathbf{E}_{\widehat{\xi}\sim\widehat{\mathrm{P}}_N}\left[\log\left(\mathbf{E}_{\zeta\sim\mathcal{N}(\widehat{\xi},\sigma^2\mathbf{I})}\left[h_{\widehat{\xi}}(z,\lambda;\zeta)\right]\right)\right]$$

Theorem (*Azizian et al., 2023*)

If f is convex and $f(\cdot,\zeta)$ is differentiable for all $\zeta \in \Xi$, then F is convex and differentiable, and:

$$\nabla_z F(z,\lambda) = \frac{1}{N}\sum_{k=1}^{N} \frac{\displaystyle\int_{\Xi} \nabla_z f(z,\lambda)\frac{h_{\widehat{\xi}}(z,\lambda;\zeta)}{(2\pi)^{d/2}\sigma}\exp\left(-\frac{\|\widehat{\xi}-\zeta\|^2}{2\sigma^2}\right)\mathrm{d}\zeta}{\displaystyle\int_{\Xi}\frac{h_{\widehat{\xi}}(z,\lambda;\zeta)}{(2\pi)^{d/2}\sigma}\exp\left(-\frac{\|\widehat{\xi}-\zeta\|^2}{2\sigma^2}\right)\mathrm{d}\zeta}$$

$$\frac{\partial}{\partial\lambda}F(z,\lambda) = \rho - \frac{1}{N}\sum_{k=1}^{N} \frac{\displaystyle\int_{\Xi} c\left(\widehat{\xi}_k,\zeta\right)\frac{h_{\widehat{\xi}}(z,\lambda;\zeta)}{(2\pi)^{d/2}\sigma}\exp\left(-\frac{\|\widehat{\xi}-\zeta\|^2}{2\sigma^2}\right)\mathrm{d}\zeta}{\displaystyle\int_{\Xi}\frac{h_{\widehat{\xi}}(z,\lambda;\zeta)}{(2\pi)^{d/2}\sigma}\exp\left(-\frac{\|\widehat{\xi}-\zeta\|^2}{2\sigma^2}\right)\mathrm{d}\zeta}$$

Gradient estimate

Samples : $\zeta_1^{(k)}, \ldots, \zeta_S^{(k)} \sim \mathcal{N}\left(\widehat{\xi}_k, \sigma^2 \mathbf{I}\right) \quad \forall k \in [\![N]\!]$ **[*Vincent et al., 2024*]**

$$\mathcal{G}_z(z, \lambda) \overset{\text{def}}{=} \frac{1}{N} \sum_{k=1}^{N} \frac{\frac{1}{S} \sum_{j=1}^{S} \nabla_z f\left(z, \zeta_j^{(k)}\right) h_{\widehat{\xi}}\left(z, \lambda ; \zeta_j^{(k)}\right)}{\frac{1}{S} \sum_{j=1}^{S} h_{\widehat{\xi}}\left(z, \lambda ; \zeta_j^{(k)}\right)}$$

Gradient estimate

Samples : $\zeta_1^{(k)}, \ldots, \zeta_S^{(k)} \sim \mathcal{N}\left(\widehat{\xi}_k, \sigma^2 \mathbf{I}\right) \quad \forall k \in [\![N]\!]$ **[*Vincent et al., 2024*]**

$$\mathcal{G}_z(z, \lambda) \overset{\text{def}}{=} \frac{1}{N} \sum_{k=1}^{N} \frac{\frac{1}{S} \sum_{j=1}^{S} \nabla_z f\left(z, \zeta_j^{(k)}\right) h_{\widehat{\xi}}\left(z, \lambda ; \zeta_j^{(k)}\right)}{\frac{1}{S} \sum_{j=1}^{S} h_{\widehat{\xi}}\left(z, \lambda ; \zeta_j^{(k)}\right)}$$

Example (Quadratic Spanning Tree — sampling cost)

$f(z, \xi) = z^\top \xi z$ and $\nabla_z f(z, \xi) = (\xi + \xi^\top) z$: easy.
Evaluation of $\mathcal{G}(z, \lambda)$: takes $S \times N$ samplings of $m \times m$ matrices: not easy.

Gradient estimate

Samples : $\zeta_1^{(k)}, \dots, \zeta_S^{(k)} \sim \mathcal{N}\left(\widehat{\xi}_k, \sigma^2 \mathbf{I}\right) \quad \forall k \in [\![N]\!]$ **[*Vincent et al., 2024*]**

$$\mathcal{G}_z(z,\lambda) \overset{\text{def}}{=} \frac{1}{N}\sum_{k=1}^{N} \frac{\frac{1}{S}\sum_{j=1}^{S} \nabla_z f\left(z, \zeta_j^{(k)}\right) h_{\widehat{\xi}}\left(z,\lambda\,;\zeta_j^{(k)}\right)}{\frac{1}{S}\sum_{j=1}^{S} h_{\widehat{\xi}}\left(z,\lambda\,;\zeta_j^{(k)}\right)}$$

Example (Quadratic Spanning Tree — sampling cost)

$f(z,\xi) = z^\top \xi z$ and $\nabla_z f(z,\xi) = (\xi + \xi^\top)z$: easy.
Evaluation of $\mathcal{G}(z,\lambda)$: takes $S \times N$ samplings of $m \times m$ matrices: not easy.

How to limit sampling costs?

Gradient estimate

Let $\mathcal{I} \subseteq [\![N]\!]$ "mini-batch"

Samples : $\zeta_1^{(k)}, \ldots, \zeta_S^{(k)} \sim \mathcal{N}\left(\widehat{\xi}_k, \sigma^2 \mathbf{I}\right) \quad \forall k \in \mathcal{I}$ **[*Vincent et al., 2024*]**

$$\mathcal{G}_z^{\mathcal{I}}(z, \lambda) \overset{\text{def}}{=} \frac{1}{|\mathcal{I}|} \sum_{k \in \mathcal{I}} \frac{\frac{1}{S} \sum_{j=1}^{S} \nabla_z f\left(z, \zeta_j^{(k)}\right) h_{\widehat{\xi}}\left(z, \lambda ; \zeta_j^{(k)}\right)}{\frac{1}{S} \sum_{j=1}^{S} h_{\widehat{\xi}}\left(z, \lambda ; \zeta_j^{(k)}\right)}$$

Gradient estimate

Let $\mathcal{I} \subseteq [\![N]\!]$ "mini-batch"

Samples : $\zeta_1^{(k)}, \dots, \zeta_S^{(k)} \sim \mathcal{N}\left(\widehat{\xi}_k, \sigma^2 \mathbf{I}\right) \quad \forall k \in \mathcal{I}$ **[*Vincent et al., 2024*]**

$$\mathcal{G}_z^{\mathcal{I}}(z, \lambda) \overset{\text{def}}{=} \frac{1}{|\mathcal{I}|} \sum_{k \in \mathcal{I}} \frac{\frac{1}{S} \sum_{j=1}^{S} \nabla_z f\left(z, \zeta_j^{(k)}\right) h_{\widehat{\xi}}\left(z, \lambda ; \zeta_j^{(k)}\right)}{\frac{1}{S} \sum_{j=1}^{S} h_{\widehat{\xi}}\left(z, \lambda ; \zeta_j^{(k)}\right)}$$

Proposition

Let $\mathcal{I}$: random subset. $\forall k \in \mathcal{I}$, let $\zeta_1^{(k)}, \dots, \zeta_S^{(k)} \sim \mathcal{N}\left(\widehat{\xi}_k, \sigma^2 \mathbf{I}\right)$ independant samples. Then $\mathbf{E}\left[\mathcal{G}^{\mathcal{I}}(z, \lambda)\right] \underset{S \to +\infty}{\longrightarrow} \nabla F(z, \lambda)$ almost surely.

Stochastic Frank-Wolfe algorithm

Algorithm 1: *Momentum stochastic Frank-Wolfe* **[Braun et al., 2022]**

Input: Starting point $y_0 \in \mathcal{Y}$, step sizes γ_t and momentum terms $\beta_t \in [0, 1]$.

for $t = 0$ to $\ldots$ **do**

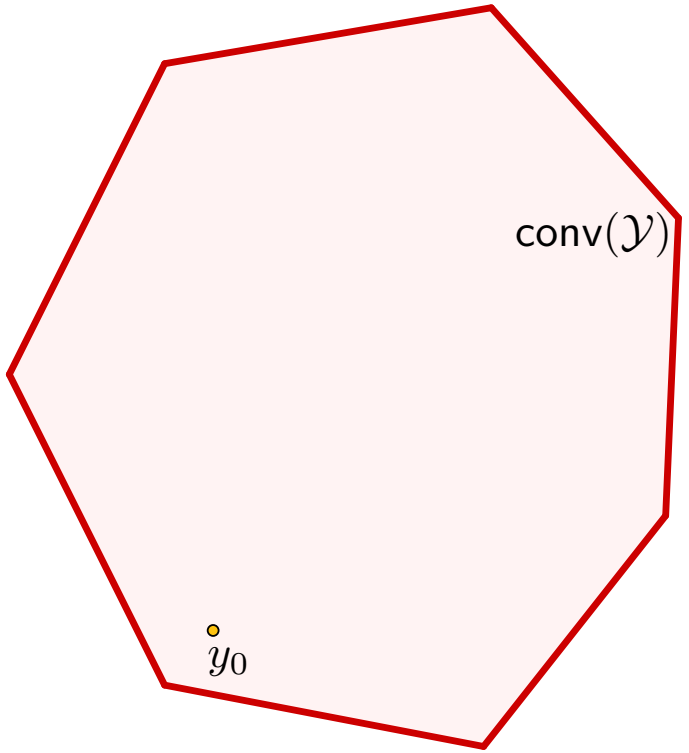
Take $\mathcal{I}_t \subseteq [\![N]\!]$ random

$\widehat{\nabla} F(y_t) \leftarrow \beta_t \mathcal{G}^{\mathcal{I}_t}(y_t) + (1 - \beta_t)\widehat{\nabla} F(y_{t-1})$

$v_t \leftarrow \operatorname{argmin}_{v \in \operatorname{conv}(\mathcal{Y})} \mathrm{LMO}\left(\widehat{\nabla} F(y_t), v\right)$

$y_{t+1} \leftarrow z_t + \gamma_t (v_t - y_t)$

end for



Stochastic Frank-Wolfe algorithm

Algorithm 1: *Momentum stochastic Frank-Wolfe* **[Braun et al., 2022]**

Input: Starting point $y_0 \in \mathcal{Y}$, step sizes γ_t and momentum terms $\beta_t \in [0, 1]$.

for $t = 0$ to $\ldots$ **do**

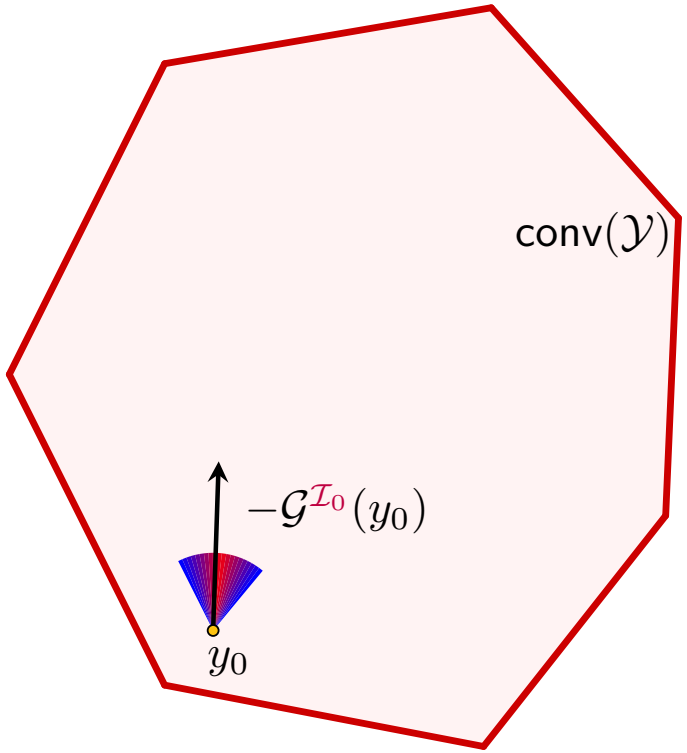
Take $\mathcal{I}_t \subseteq [\![N]\!]$ random

$\widehat{\nabla} F(y_t) \leftarrow \beta_t \mathcal{G}^{\mathcal{I}_t}(y_t) + (1 - \beta_t)\widehat{\nabla} F(y_{t-1})$

$v_t \leftarrow \operatorname{argmin}_{v \in \operatorname{conv}(\mathcal{Y})} \mathrm{LMO}\left(\widehat{\nabla} F(y_t), v\right)$

$y_{t+1} \leftarrow z_t + \gamma_t (v_t - y_t)$

end for



Stochastic Frank-Wolfe algorithm

Algorithm 1: *Momentum stochastic Frank-Wolfe* **[*Braun et al., 2022*]**

Input: Starting point $y_0 \in \mathcal{Y}$, step sizes γ_t and momentum terms $\beta_t \in [0, 1]$.

for $t = 0$ to $\ldots$ **do**

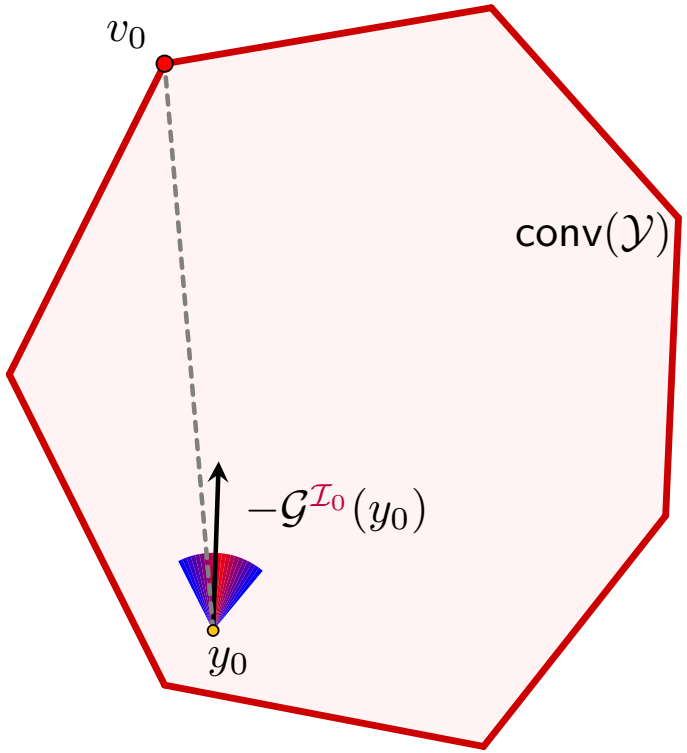
Take $\mathcal{I}_t \subseteq [\![N]\!]$ random

$\widehat{\nabla} F(y_t) \leftarrow \beta_t \mathcal{G}^{\mathcal{I}_t}(y_t) + (1 - \beta_t)\widehat{\nabla} F(y_{t-1})$

$v_t \leftarrow \operatorname{argmin}_{v \in \mathrm{conv}(\mathcal{Y})} \mathrm{LMO}\left(\widehat{\nabla} F(y_t), v\right)$

$y_{t+1} \leftarrow z_t + \gamma_t (v_t - y_t)$

end for



Stochastic Frank-Wolfe algorithm

Algorithm 1: *Momentum stochastic Frank-Wolfe* **[Braun et al., 2022]**

Input: Starting point $y_0 \in \mathcal{Y}$, step sizes γ_t and momentum terms $\beta_t \in [0, 1]$.

for $t = 0$ to $\ldots$ **do**

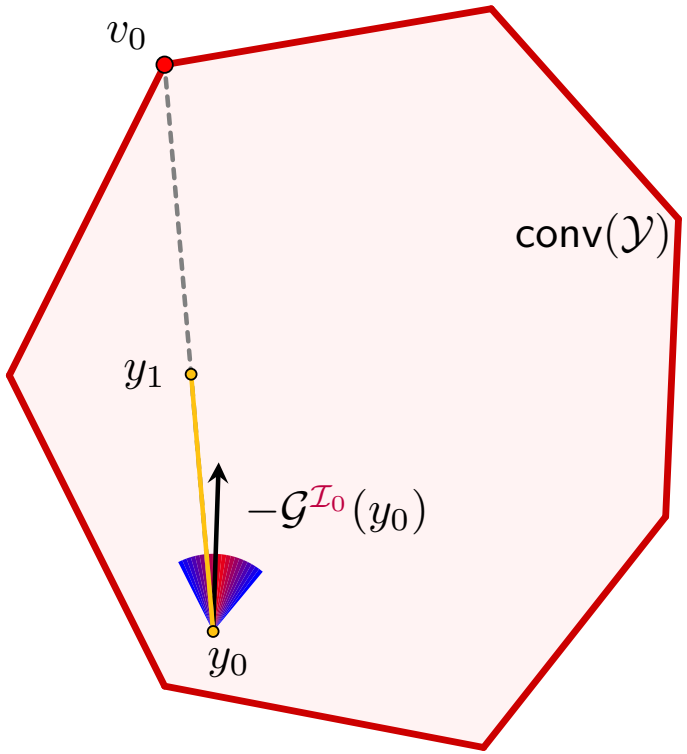
Take $\mathcal{I}_t \subseteq [\![N]\!]$ random

$\widehat{\nabla} F(y_t) \leftarrow \beta_t \mathcal{G}^{\mathcal{I}_t}(y_t) + (1 - \beta_t)\widehat{\nabla} F(y_{t-1})$

$v_t \leftarrow \operatorname{argmin}_{v \in \operatorname{conv}(\mathcal{Y})} \mathrm{LMO}\left(\widehat{\nabla} F(y_t), v\right)$

$y_{t+1} \leftarrow z_t + \gamma_t (v_t - y_t)$

end for



Stochastic Frank-Wolfe algorithm

Algorithm 1: *Momentum stochastic Frank-Wolfe* **[Braun et al., 2022]**

Input: Starting point $y_0 \in \mathcal{Y}$, step sizes γ_t and momentum terms $\beta_t \in [0, 1]$.

for $t = 0$ to $\ldots$ **do**

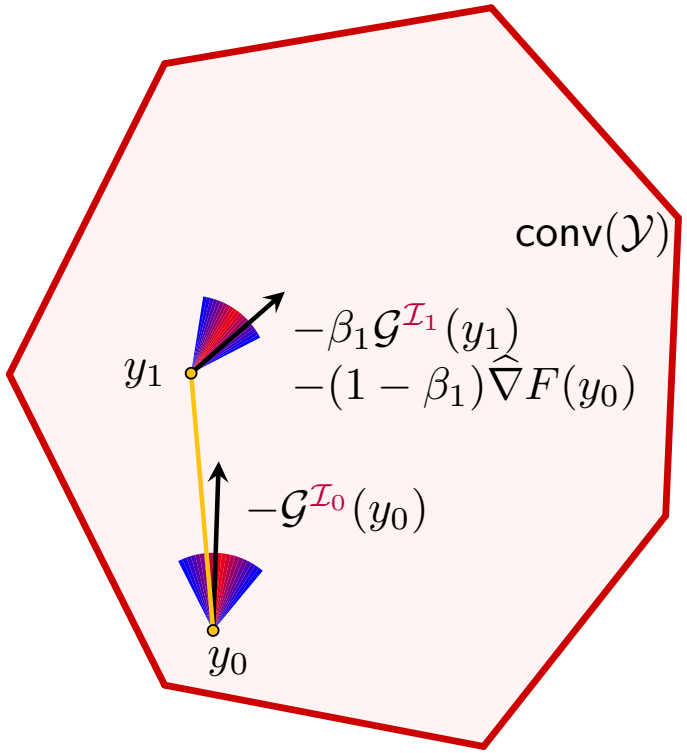
Take $\mathcal{I}_t \subseteq [\![N]\!]$ random

$\widehat{\nabla} F(y_t) \leftarrow \beta_t \mathcal{G}^{\mathcal{I}_t}(y_t) + (1 - \beta_t)\widehat{\nabla} F(y_{t-1})$

$v_t \leftarrow \operatorname{argmin}_{v \in \operatorname{conv}(\mathcal{Y})} \mathrm{LMO}\left(\widehat{\nabla} F(y_t), v\right)$

$y_{t+1} \leftarrow z_t + \gamma_t (v_t - y_t)$

end for



Stochastic Frank-Wolfe algorithm

Algorithm 1: *Momentum stochastic Frank-Wolfe* **[*Braun et al., 2022*]**

Input: Starting point $y_0 \in \mathcal{Y}$, step sizes γ_t and momentum terms $\beta_t \in [0, 1]$.

for $t = 0$ to $\ldots$ **do**

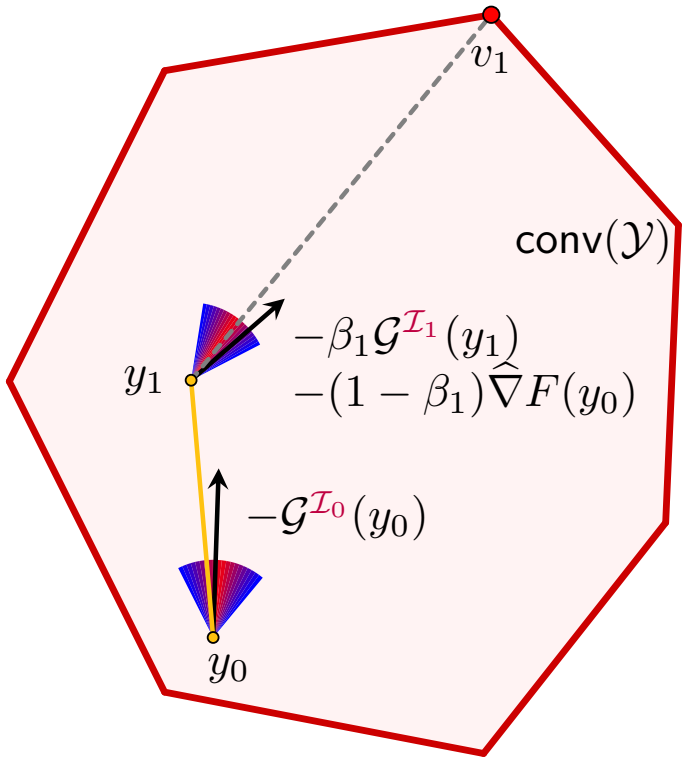
Take $\mathcal{I}_t \subseteq [\![N]\!]$ random

$\widehat{\nabla} F(y_t) \leftarrow \beta_t \mathcal{G}^{\mathcal{I}_t}(y_t) + (1 - \beta_t)\widehat{\nabla} F(y_{t-1})$

$v_t \leftarrow \operatorname{argmin}_{v \in \operatorname{conv}(\mathcal{Y})} \mathrm{LMO}\left(\widehat{\nabla} F(y_t), v\right)$

$y_{t+1} \leftarrow z_t + \gamma_t (v_t - y_t)$

end for



Stochastic Frank-Wolfe algorithm

Algorithm 1: *Momentum stochastic Frank-Wolfe* **[*Braun et al., 2022*]**

Input: Starting point $y_0 \in \mathcal{Y}$, step sizes γ_t and momentum terms $\beta_t \in [0, 1]$.

for $t = 0$ to $\ldots$ **do**

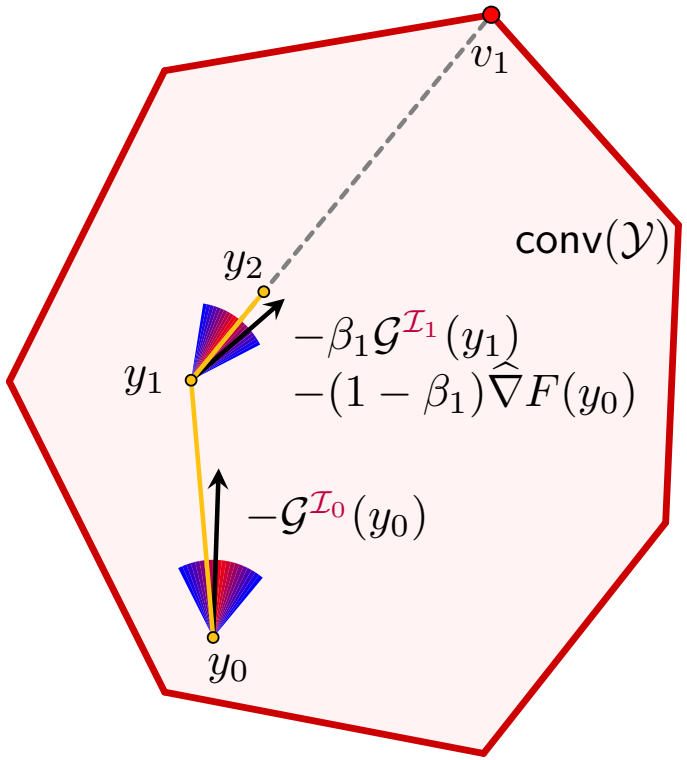
Take $\mathcal{I}_t \subseteq [\![N]\!]$ random

$\widehat{\nabla} F(y_t) \leftarrow \beta_t \mathcal{G}^{\mathcal{I}_t}(y_t) + (1 - \beta_t)\widehat{\nabla} F(y_{t-1})$

$v_t \leftarrow \operatorname{argmin}_{v \in \mathrm{conv}(\mathcal{Y})} \mathrm{LMO}\left(\widehat{\nabla} F(y_t), v\right)$

$y_{t+1} \leftarrow z_t + \gamma_t (v_t - y_t)$

end for



Stochastic Frank-Wolfe algorithm

Algorithm 1: *Momentum stochastic Frank-Wolfe* [***Braun et al., 2022***]

Input: Starting point $y_0 \in \mathcal{Y}$, step sizes γ_t and momentum terms $\beta_t \in [0, 1]$.

for $t = 0$ to $\ldots$ **do**

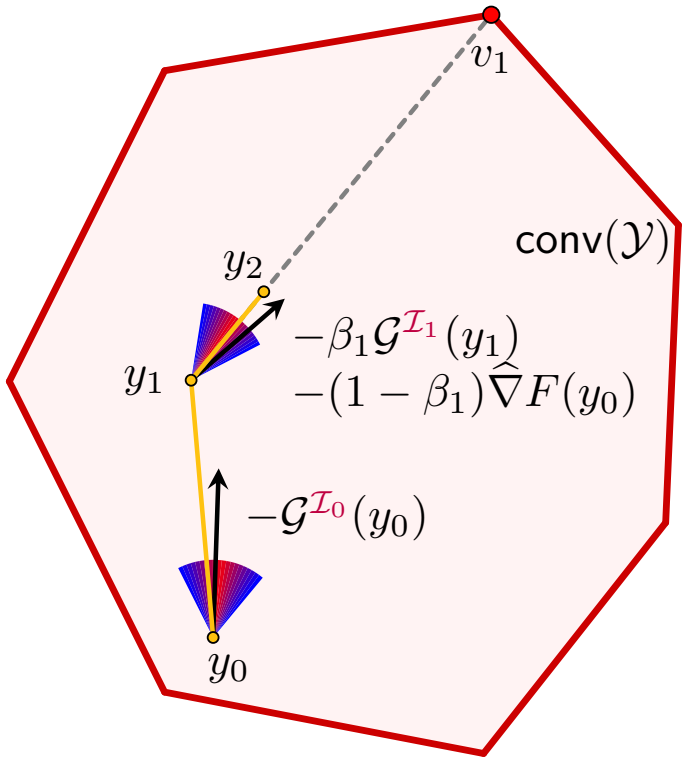
Take $\mathcal{I}_t \subseteq [\![N]\!]$ random

$\widehat{\nabla} F(y_t) \leftarrow \beta_t \mathcal{G}^{\mathcal{I}_t}(y_t) + (1 - \beta_t)\widehat{\nabla} F(y_{t-1})$

$v_t \leftarrow \operatorname{argmin}_{v \in \mathrm{conv}(\mathcal{Y})} \mathrm{LMO}\left(\widehat{\nabla} F(y_t), v\right)$

$y_{t+1} \leftarrow z_t + \gamma_t (v_t - y_t)$

end for



Remarks

- LMO offers flexible modelling: add constraints, cuts, ... [***Besançon et al., 2022***]

Stochastic Frank-Wolfe algorithm

Algorithm 1: *Momentum stochastic Frank-Wolfe* **[*Braun et al., 2022*]**

Input: Starting point $y_0 \in \mathcal{Y}$, step sizes γ_t and momentum terms $\beta_t \in [0, 1]$.

for $t = 0$ to $\ldots$ **do**

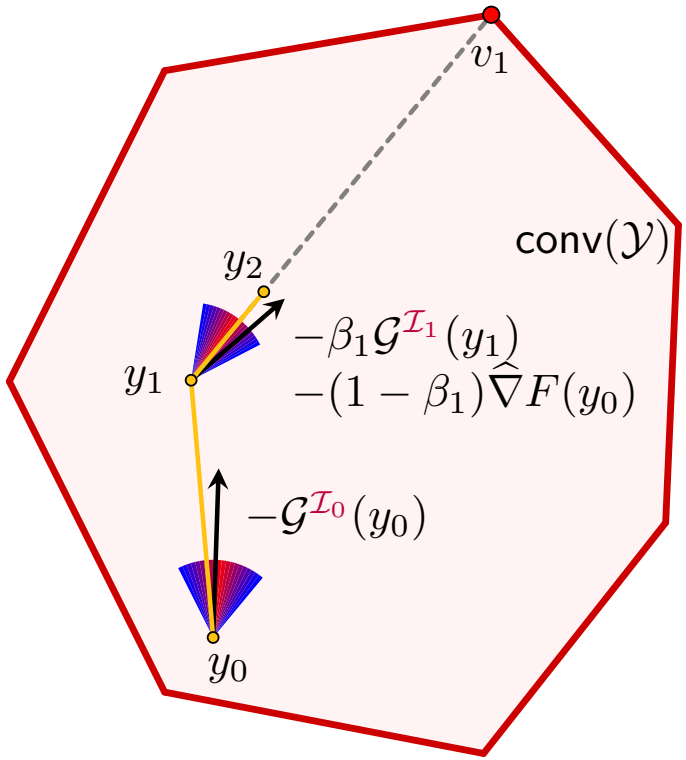
Take $\mathcal{I}_t \subseteq [\![N]\!]$ random

$\widehat{\nabla} F(y_t) \leftarrow \beta_t \mathcal{G}^{\mathcal{I}_t}(y_t) + (1 - \beta_t)\widehat{\nabla} F(y_{t-1})$

$v_t \leftarrow \operatorname{argmin}_{v \in \mathsf{conv}(\mathcal{Y})} \mathsf{LMO}\left(\widehat{\nabla} F(y_t), v\right)$

$y_{t+1} \leftarrow z_t + \gamma_t (v_t - y_t)$

end for



Remarks

- LMO offers flexible modelling: add constraints, cuts, ... **[*Besançon et al., 2022*]**
- Quadratic spanning tree: LMO = Kruskal$\rightarrow$ solved efficiently!

Outline

Experimental setup

	ERM	Robust
Objective	$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \frac{1}{N}\sum_{k=1}^{N} z^\top \widehat{\xi}_k z$	$\underset{z\in\mathcal{Z},\lambda\geq 0}{\text{minimize}} \quad F(z,\lambda)$

Experimental setup

	ERM	Robust
Objective	$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \frac{1}{N}\sum_{k=1}^{N} z^\top \widehat{\xi}_k z$	$\underset{z\in\mathcal{Z},\lambda\geq 0}{\text{minimize}} \quad F(z,\lambda)$
Gradient	$\frac{1}{N}\sum_{k=1}^{N}\left(\widehat{\xi}_k + \widehat{\xi}_k{}^\top\right) z$	$\mathcal{G}^{\mathcal{I}}(z,\lambda)$

Experimental setup

	ERM	Robust
Objective	$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \frac{1}{N}\sum_{k=1}^{N} z^\top \widehat{\xi}_k z$	$\underset{z\in\mathcal{Z},\lambda\geq 0}{\text{minimize}} \quad F(z,\lambda)$
Gradient	$\frac{1}{N}\sum_{k=1}^{N}\left(\widehat{\xi}_k + \widehat{\xi}_k{}^\top\right) z$	$\mathcal{G}^{\mathcal{I}}(z,\lambda)$
Solution	z_{erm}	z_{robust}

Experimental setup

	ERM	Robust
Objective	$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \frac{1}{N}\sum_{k=1}^{N} z^\top \widehat{\xi}_k z$	$\underset{z\in\mathcal{Z},\lambda\geq 0}{\text{minimize}} \quad F(z,\lambda)$
Gradient	$\frac{1}{N}\sum_{k=1}^{N}\left(\widehat{\xi}_k + \widehat{\xi}_k^\top\right) z$	$\mathcal{G}^{\mathcal{I}}(z,\lambda)$
Solution	z_{erm}	z_{robust}
Train Loss	$\left\{ z_{\text{erm}}{}^\top \widehat{\xi}_k z_{\text{erm}} \;\middle|\; k \in [\![N]\!] \right\}$	$\left\{ z_{\text{robust}}{}^\top \widehat{\xi}_k z_{\text{robust}} \;\middle|\; k \in [\![N]\!] \right\}$

Experimental setup

	ERM	Robust
Objective	$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \frac{1}{N}\sum_{k=1}^{N} z^\top \widehat{\xi}_k z$	$\underset{z\in\mathcal{Z},\lambda\geq 0}{\text{minimize}} \quad F(z,\lambda)$
Gradient	$\frac{1}{N}\sum_{k=1}^{N}\left(\widehat{\xi}_k + \widehat{\xi}_k^\top\right) z$	$\mathcal{G}^{\mathcal{I}}(z,\lambda)$
Solution	z_{erm}	z_{robust}
Train Loss	$\left\{ z_{\text{erm}}{}^\top \widehat{\xi}_k z_{\text{erm}} \;\middle|\; k \in [\![N]\!] \right\}$	$\left\{ z_{\text{robust}}{}^\top \widehat{\xi}_k z_{\text{robust}} \;\middle|\; k \in [\![N]\!] \right\}$

Build $\tilde{\mathbf{P}}$ a shifted distribution, *e.g.* $\tilde{\mathbf{P}} = \mathbf{P} + \text{noise}$

Sample Test Set $= \left\{\tilde{\xi} \sim \tilde{\mathbf{P}}\right\}$

Experimental setup

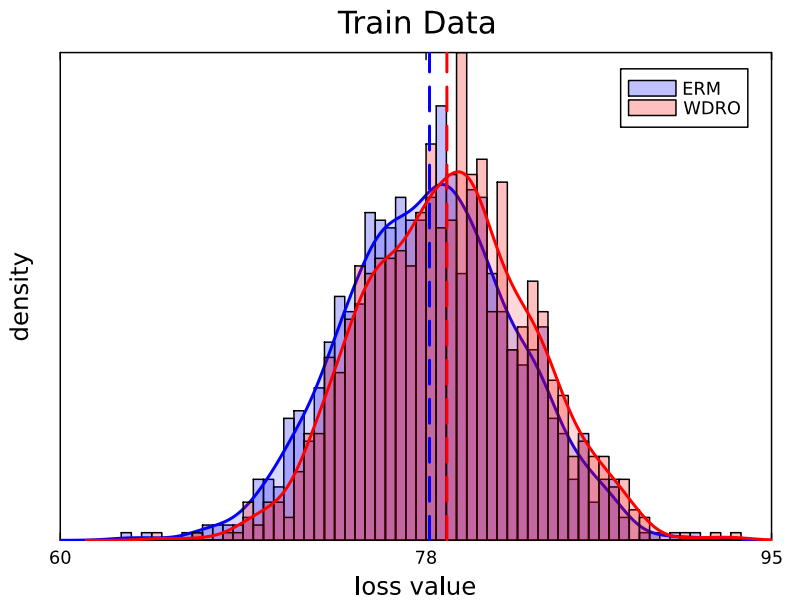
	ERM	Robust
Objective	$\underset{z\in\mathcal{Z}}{\text{minimize}} \quad \frac{1}{N}\sum_{k=1}^{N} z^\top \widehat{\xi}_k z$	$\underset{z\in\mathcal{Z},\lambda\geq 0}{\text{minimize}} \quad F(z,\lambda)$
Gradient	$\frac{1}{N}\sum_{k=1}^{N}\left(\widehat{\xi}_k + \widehat{\xi}_k^\top\right) z$	$\mathcal{G}^{\mathcal{I}}(z,\lambda)$
Solution	z_{erm}	z_{robust}
Train Loss	$\left\{ z_{\text{erm}}^\top \widehat{\xi}_k z_{\text{erm}} \;\middle|\; k \in [\![N]\!] \right\}$	$\left\{ z_{\text{robust}}^\top \widehat{\xi}_k z_{\text{robust}} \;\middle|\; k \in [\![N]\!] \right\}$

Build $\tilde{\mathbf{P}}$ a shifted distribution, *e.g.* $\tilde{\mathbf{P}} = \mathbf{P} + \text{noise}$

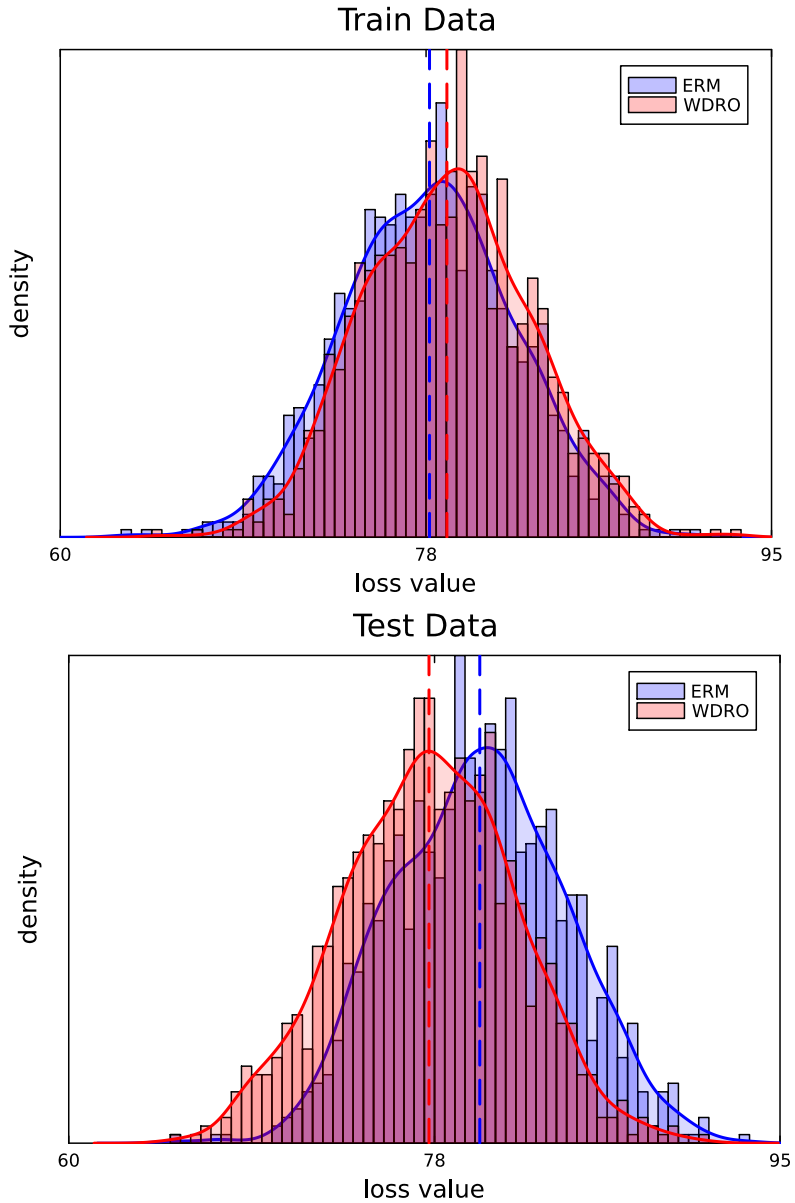
Sample Test Set $= \left\{\tilde{\xi} \sim \tilde{\mathbf{P}}\right\}$

Test Loss	$\left\{ z_{\text{erm}}^\top \tilde{\xi} z_{\text{erm}} \;\middle|\; \tilde{\xi} \in \text{Test Set} \right\}$	$\left\{ z_{\text{robust}}^\top \tilde{\xi} z_{\text{robust}} \;\middle|\; \tilde{\xi} \in \text{Test Set} \right\}$

Experiments on Quadratic Minimum Spanning Tree



Experiments on Quadratic Minimum Spanning Tree



Conclusion and perspectives

- Entropic regularization of WDRO:
  - ➞ Convex & differentiable objective function
  - ➞ Explicit gradient formulation

Conclusion and perspectives

- Entropic regularization of WDRO:
  - ➞ Convex & differentiable objective function
  - ➞ Explicit gradient formulation
- Stochastic estimator of the gradient:
  - ➞ Adapted to large dimensions

Conclusion and perspectives

- Entropic regularization of WDRO:
  - ➞ Convex & differentiable objective function
  - ➞ Explicit gradient formulation
- Stochastic estimator of the gradient:
  - ➞ Adapted to large dimensions
- Momentum stochastic Frank-Wolfe algorithm:
  - ➞ Handle the noise on the gradient estimator
  - ➞ Allows for constant batch size
  - ➞ Flexible modelling through LMO

Conclusion and perspectives

- Entropic regularization of WDRO:
  - ➞ Convex & differentiable objective function
  - ➞ Explicit gradient formulation
- Stochastic estimator of the gradient:
  - ➞ Adapted to large dimensions
- Momentum stochastic Frank-Wolfe algorithm:
  - ➞ Handle the noise on the gradient estimator
  - ➞ Allows for constant batch size
  - ➞ Flexible modelling through LMO

} for generic convex differentiable loss function f

Conclusion and perspectives

- Entropic regularization of WDRO:
  - ➞ Convex & differentiable objective function
  - ➞ Explicit gradient formulation
- Stochastic estimator of the gradient:
  - ➞ Adapted to large dimensions
- Momentum stochastic Frank-Wolfe algorithm:
  - ➞ Handle the noise on the gradient estimator
  - ➞ Allows for constant batch size
  - ➞ Flexible modelling through LMO

} for generic convex differentiable loss function f

- Illustration on the Quadratic Minimum Spanning Tree Problem:
  - ➞ Robust solutions to distributional shift

Conclusion and perspectives

- Entropic regularization of WDRO:
  - → Convex & differentiable objective function
  - → Explicit gradient formulation
- Stochastic estimator of the gradient:
  - → Adapted to large dimensions
- Momentum stochastic Frank-Wolfe algorithm:
  - → Handle the noise on the gradient estimator
  - → Allows for constant batch size
  - → Flexible modelling through LMO

} for generic convex differentiable loss function f

- Illustration on the Quadratic Minimum Spanning Tree Problem:
  - → Robust solutions to distributional shift
- Adaptation to two-stage problems:
  - → *e.g.* Facility Location problems

References I


Assad, A. and Xu, W. (1992).
The quadratic minimum spanning tree problem.
Naval Research Logistics.


Azizian, W., Iutzeler, F., and Malick, J. (2023).
Regularization for Wasserstein distributionally robust optimization.
ESAIM: Control, Optimisation and Calculus of Variations.


Besançon, M., Carderera, A., and Pokutta, S. (2022).
`FrankWolfe.jl`: A high-performance and flexible toolbox for Frank–Wolfe algorithms and conditional gradients.
INFORMS Journal on Computing.


Braun, G., Carderera, A., Combettes, C., Hassani, H., Karbasi, A., Mokhtari, A., and Pokutta, S. (2022).
Conditional gradient methods.
arXiv preprint arXiv:2211.14103.


Chow, V. T. F., Cui, Z., and Long, D. Z. (2022).
Target-oriented distributionally robust optimization and its applications to surgery allocation.
INFORMS Journal on Computing.

References II


Esfahani, P. and Kuhn, D. (2018).
Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations.
Mathematical Programming.


Ghosal, S. and Wiesemann, W. (2020).
The distributionally robust chance-constrained vehicle routing problem.
Operations Research.


Schrijver, A. (2002).
Combinatorial optimization.


Sun, L., Xie, W., and Witten, T. (2023).
Distributionally robust fair transit resource allocation during a pandemic.
Transportation science.


Vincent, F., Azizian, W., Iutzeler, F., and Malick, J. (2024).
`skwdro`: A library for Wasserstein distributionally robust machine learning.