

## PPA avec contrainte et Lagrangien augmenté

05 janvier 2021

# Plan du cours

- 1 Principe du Problème Auxiliaire avec contrainte
  - Rappel du PPA dans le cas sans contrainte
  - Application du PPA dans le cas avec contrainte
  - Algorithmes du PPA pour la recherche de point selle
- 2 Régularisation et Lagrangien augmenté
  - Transformée de Moreau-Yosida
  - Application à la minimisation et méthodes proximales
  - Application à la recherche de point selle
- 3 Travaux dirigés sur le Principe du Problème Auxiliaire
  - Un peu de calcul sous-différentiel. . .
  - Étude d'une vallée hydraulique
  - Réseau de distribution d'eau

## 1 Principe du Problème Auxiliaire avec contrainte

- Rappel du PPA dans le cas sans contrainte
- Application du PPA dans le cas avec contrainte
- Algorithmes du PPA pour la recherche de point selle

## 2 Régularisation et Lagrangien augmenté

- Transformée de Moreau-Yosida
- Application à la minimisation et méthodes proximales
- Application à la recherche de point selle

## 3 Travaux dirigés sur le Principe du Problème Auxiliaire

- Un peu de calcul sous-différentiel...
- Étude d'une vallée hydraulique
- Réseau de distribution d'eau

## 1 Principe du Problème Auxiliaire avec contrainte

- Rappel du PPA dans le cas sans contrainte
- Application du PPA dans le cas avec contrainte
- Algorithmes du PPA pour la recherche de point selle

## 2 Régularisation et Lagrangien augmenté

- Transformée de Moreau-Yosida
- Application à la minimisation et méthodes proximales
- Application à la recherche de point selle

## 3 Travaux dirigés sur le Principe du Problème Auxiliaire

- Un peu de calcul sous-différentiel...
- Étude d'une vallée hydraulique
- Réseau de distribution d'eau

# Rappel du cours précédent

I

Pour présenter le **Principe du Problème Auxiliaire (PPA)**, on a considéré des problèmes d'optimisation dans lesquels ne figure **pas de contrainte explicite** :

$$\min_{u \in U^{\text{ad}} \subset \mathcal{U}} J(u) + J^{\Sigma}(u) , \quad (P_0)$$

avec

- $U^{\text{ad}}$  convexe fermé non vide de  $\mathcal{U}$ ,
- $J$  et  $J^{\Sigma}$  convexes, s.c.i., propres,
- $J$  différentiable,
- $J + J^{\Sigma}$  coercive.

On ne fait pas d'hypothèse spécifique quant à la structure de la fonction  $J$ , mais on suppose que l'ensemble  $U^{\text{ad}}$  et la fonction de coût  $J^{\Sigma}$  « ne posent pas de problème en décomposition » :

$$U^{\text{ad}} = U_1^{\text{ad}} \times \cdots \times U_N^{\text{ad}} , \quad J^{\Sigma}(u) = \sum_{i=1}^N J_i^{\Sigma}(u_i) .$$

# Rappel du cours précédent

II

Le **Principe du Problème Auxiliaire** appliqué à  $(P_0)$  consiste à **remplacer** la fonction  $J$  par son **approximation linéaire** autour d'un point  $u^{(k)}$ , et à ajouter un terme assurant la **coercivité** de la fonction ainsi obtenue. Pour cela, on **choisit** :

- une suite de fonctions  $\{K^{(k)}\}_{k \in \mathbb{N}}$  **fortement convexes**,
- une suite de coefficients  $\{\epsilon^{(k)}\}_{k \in \mathbb{N}}$  positifs,

et on remplace la résolution du problème  $(P_0)$  par la résolution d'une **suite de problèmes auxiliaires**  $\{(PA^{(k)})\}_{k \in \mathbb{N}}$ . L'expression du problème auxiliaire  $(PA^{(k)})$  est

$$\min_{u \in U^{\text{ad}}} K^{(k)}(u) + \langle \epsilon^{(k)} \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}), u \rangle + \epsilon^{(k)} J^{\Sigma}(u),$$

dont l'**unique solution**  $u^{(k+1)}$  permet de construire  $(PA^{(k+1)})$ .

# Rappel du cours précédent



On a alors vu :

- ① un **théorème** de convergence de la suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$ ,
- ② comment prendre les coefficients  $\epsilon^{(k)}$  égaux à 1 ( $\epsilon^{(k)} < \frac{2b^{(k)}}{A}$ ),
- ③ différents **algorithmes** issus de choix des fonctions  $K^{(k)}$ ,
- ④ comment le choix d'un **noyau additif** :

$$K^{(k)}(u) = \sum_{i=1}^N K_i^{(k)}(u_i) .$$

permet de **décomposer** le problème  $(PA^{(k)})$   $u_i$  par  $u_i$  :

$$\min_{u_i \in U_i^{\text{ad}}} K_i^{(k)}(u_i) + \langle \epsilon^{(k)} \nabla_{u_i} J(u^{(k)}) - \nabla K_i^{(k)}(u_i^{(k)}), u_i \rangle + \epsilon^{(k)} J_i^{\Sigma}(u_i) .$$

## 1 Principe du Problème Auxiliaire avec contrainte

- Rappel du PPA dans le cas sans contrainte
- Application du PPA dans le cas avec contrainte
- Algorithmes du PPA pour la recherche de point selle

## 2 Régularisation et Lagrangien augmenté

- Transformée de Moreau-Yosida
- Application à la minimisation et méthodes proximales
- Application à la recherche de point selle

## 3 Travaux dirigés sur le Principe du Problème Auxiliaire

- Un peu de calcul sous-différentiel...
- Étude d'une vallée hydraulique
- Réseau de distribution d'eau



# Cadre du problème

On s'intéresse aujourd'hui au **cas général** :

$$\min_{u \in U^{\text{ad}}} J(u) + J^{\Sigma}(u) \quad \text{sous} \quad \Theta(u) + \Theta^{\Sigma}(u) \in -C. \quad (P_1)$$

Sous une hypothèse de **qualification** des contraintes, le problème est équivalent à celui de trouver un **point selle** du Lagrangien associé dans l'ensemble  $U^{\text{ad}} \times C^*$ . Ce **Lagrangien** s'écrit :

$$L(u, p) + L^{\Sigma}(u, p),$$

avec  $L(u, p) = J(u) + \langle p, \Theta(u) \rangle$  et  $L^{\Sigma}(u, p) = J^{\Sigma}(u) + \langle p, \Theta^{\Sigma}(u) \rangle$ .

Comme dans le cas sans contrainte, on suppose que la partie  $L^{\Sigma}$  du Lagrangien « ne pose pas de problème en décomposition »<sup>12</sup>.

12. dans un sens qu'il faudra préciser. . .

# Extension du PPA au cas des points selle

I

L'extension du PPA au problème avec contrainte  $(P_1)$  consiste, en supposant la partie  $L$  du Lagrangien différentiable, à prendre l'approximation linéaire de  $L$  autour d'un point  $(u^{(k)}, p^{(k)})$  et à ajouter un opérateur auxiliaire  $\Lambda(u, p)$  convexe en  $u$  et concave en  $p$ . Le problème auxiliaire  $(PA^{(k)})$  consiste alors à trouver un point selle de l'opérateur  $\mathcal{L}^{(k)}$  :

$$\underbrace{\Lambda(u, p) + \langle \epsilon \nabla L(u^{(k)}, p^{(k)}) - \nabla \Lambda(u^{(k)}, p^{(k)}), (u, p) \rangle + \epsilon L^\Sigma(u, p)}_{\mathcal{L}^{(k)}(u, p)},$$

avec

$$\begin{aligned} \nabla \Lambda(u, p) &= (\nabla_u \Lambda(u, p), \nabla_p \Lambda(u, p)), \\ \nabla L(u, p) &= (\nabla_u L(u, p), \nabla_p L(u, p)) \\ &= (\nabla J(u) + (\Theta'(u))^\top \cdot p, \Theta(u)). \end{aligned}$$

# Extension du PPA au cas des points selle

II

L'opérateur  $\mathfrak{L}^{(k)}$  s'écrit :

$$\begin{aligned}\mathfrak{L}^{(k)}(u, p) = & \Lambda(u, p) + \langle \epsilon \nabla_u L(u^{(k)}, p^{(k)}) - \nabla_u \Lambda(u^{(k)}, p^{(k)}), u \rangle \\ & + \langle \epsilon \nabla_p L(u^{(k)}, p^{(k)}) - \nabla_p \Lambda(u^{(k)}, p^{(k)}), p \rangle + \epsilon L^\Sigma(u, p) .\end{aligned}$$

Un **point selle**  $(u^{(k+1)}, p^{(k+1)})$  de  $\mathfrak{L}^{(k)}$  est caractérisé par :

$$\mathfrak{L}^{(k)}(u^{(k+1)}, p) \leq \mathfrak{L}^{(k)}(u^{(k+1)}, p^{(k+1)}) \leq \mathfrak{L}^{(k)}(u, p^{(k+1)}) \quad \forall (u, p) \in U^{\text{ad}} \times C^\star ,$$

ce qui s'écrit encore :

$$\begin{aligned}u^{(k+1)} & \in \arg \min_{u \in U^{\text{ad}}} \mathfrak{L}^{(k)}(u, p^{(k+1)}) , \\ p^{(k+1)} & \in \arg \max_{p \in C^\star} \mathfrak{L}^{(k)}(u^{(k+1)}, p) .\end{aligned}$$

## 1 Principe du Problème Auxiliaire avec contrainte

- Rappel du PPA dans le cas sans contrainte
- Application du PPA dans le cas avec contrainte
- Algorithmes du PPA pour la recherche de point selle

## 2 Régularisation et Lagrangien augmenté

- Transformée de Moreau-Yosida
- Application à la minimisation et méthodes proximales
- Application à la recherche de point selle

## 3 Travaux dirigés sur le Principe du Problème Auxiliaire

- Un peu de calcul sous-différentiel...
- Étude d'une vallée hydraulique
- Réseau de distribution d'eau

# Algorithme du PPA « à un niveau »

I

Une première approche consiste à **choisir**  $\Lambda$  de type **Lagrangien** :

$$\Lambda(u, p) = K(u) + \langle p, \Omega(u) \rangle .$$

Alors, l'opérateur  $\mathcal{L}^{(k)}$  est lui même un **Lagrangien** :

$$\begin{aligned} \mathcal{L}^{(k)}(u, p) = & K(u) + \langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}), u \rangle \\ & + \langle (\epsilon \Theta'(u^{(k)}) - \Omega'(u^{(k)}))^{\top} \cdot p^{(k)}, u \rangle + \epsilon J^{\Sigma}(u) \\ & + \langle p, \Omega(u) + \epsilon \Theta(u^{(k)}) - \Omega(u^{(k)}) + \epsilon \Theta^{\Sigma}(u) \rangle , \end{aligned}$$

et la recherche d'un point selle de  $\mathcal{L}^{(k)}$  correspond à la résolution du **problème auxiliaire d'optimisation sous contrainte** ( $PA^{(k)}$ ) :

$$\begin{aligned} \min_{u \in U^{\text{ad}}} & K(u) + \langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}), u \rangle \\ & + \langle (\epsilon \Theta'(u^{(k)}) - \Omega'(u^{(k)}))^{\top} \cdot p^{(k)}, u \rangle + \epsilon J^{\Sigma}(u) , \end{aligned}$$

$$\text{sous } \Omega(u) + \epsilon \Theta(u^{(k)}) - \Omega(u^{(k)}) + \epsilon \Theta^{\Sigma}(u) \in -C .$$

# Algorithme du PPA « à un niveau »

II

La résolution de  $(PA^{(k)})$  fournit un point selle  $(u^{(k+1)}, p^{(k+1)})$  qui permet de construire le problème auxiliaire  $(PA^{(k+1)})$ , d'où l'**algorithme du PPA à un niveau**.

On dispose pour cet algorithme d'un théorème de convergence. Cependant, le choix d'un noyau  $\wedge$  de type Lagrangien fait qu'il est **linéaire** en  $p$ , et donc ne peut pas être **fortement concave** en  $p$ .<sup>13</sup>

Cette non forte concavité engendre des difficultés pour prouver la convergence, qui est obtenue sous des **conditions restrictives** :

- critère  $J$  quadratique,
- contrainte  $\Theta$  linéaire,
- ensemble  $U^{\text{ad}}$  égal à tout l'espace,
- condition de compatibilité entre les opérateurs  $\Theta$  et  $\Omega$ .

13. La **forte convexité** en  $u$  peut toujours être obtenue par le choix de  $K$ .

# Décomposition du PPA « à un niveau »

III

Dans cet algorithme, la **décomposition** du problème  $(PA^{(k)})$  :

$$\min_{u \in U^{\text{ad}}} K(u) + \langle \epsilon \nabla_u L(u^{(k)}, p^{(k)}) - \nabla_u \Lambda(u^{(k)}, p^{(k)}), u \rangle + \epsilon J^\Sigma(u),$$

$$\text{sous } \Omega(u) + \epsilon \Theta(u^{(k)}) - \Omega(u^{(k)}) + \epsilon \Theta^\Sigma(u) \in -C,$$

par rapport à une décomposition de  $u$  en  $(u_1, \dots, u_N)$  signifie qu'il s'écrit comme  $N$  sous-problèmes d'optimisation **indépendants** ne dépendant chacun que d'un seul  $u_i$ . Pour cela, il faut **choisir**

- un noyau  **$K$  additif** (de même pour  $J^\Sigma$ ) :

$$K(u) = \sum_{i=1}^N K_i(u_i),$$

- un opérateur  **$\Omega$  bloc-diagonal** (de même pour  $U^{\text{ad}}$  et  $\Theta^\Sigma$ ) :

$$\Omega(u) = (\Omega_1(u_1), \dots, \Omega_N(u_N))^\top,$$

qui correspond à une **décomposition** de  $C$  en  $C_1 \times \dots \times C_N$ .

# Décomposition du PPA « à un niveau »

## IV

Avec ces **choix**, le sous-problème portant sur la variable  $i$  dans la **décomposition** du problème  $(PA^{(k)})$  s'écrit :

$$\min_{u_i \in U_i^{\text{ad}}} K_i(u_i) + \langle \epsilon \nabla_{u_i} L(u^{(k)}, p^{(k)}) - \nabla_{u_i} \Lambda(u^{(k)}, p^{(k)}), u_i \rangle + \epsilon J_i^\Sigma(u_i),$$

$$\text{sous } \Omega_i(u_i) + \epsilon \Theta_i(u^{(k)}) - \Omega_i(u_i^{(k)}) + \epsilon \Theta_i^\Sigma(u_i) \in -C_i,$$

et sa résolution fournit un point selle  $(u_i^{(k+1)}, p_i^{(k+1)})$ .

On notera la **forte proximité** entre l'algorithme de décomposition par le **PPA à un niveau** et celui de décomposition par **prédiction** :

- la **décomposition du cône**  $C$  en  $C_1 \times \cdots \times C_N$  correspond au choix d'**affection des contraintes** aux sous-problèmes ;
- dans le cas **additif**, prendre  $\epsilon = 1$ ,  $K = J$  et prendre pour  $\Omega$  la partie bloc-diagonale de  $\Theta$  suivant la décomposition du cône  $C$  en  $C_1 \times \cdots \times C_N$  conduit à l'algorithme par **prédiction**.



# Algorithme du PPA « à deux niveaux »

I

Plutôt que de résoudre le système des **deux problèmes couplés** :

$$u^{(k+1)} \in \arg \min_{u \in U^{\text{ad}}} \mathcal{L}^{(k)}(u, p^{(k+1)}) ,$$

$$p^{(k+1)} \in \arg \max_{p \in C^*} \mathcal{L}^{(k)}(u^{(k+1)}, p) ,$$

l'**approche à deux niveaux** consiste à résoudre ces deux problèmes de manière **séquentielle** :

$$u^{(k+1)} \in \arg \min_{u \in U^{\text{ad}}} \mathcal{L}^{(k)}(u, p^{(k)}) ,$$

$$p^{(k+1)} \in \arg \max_{p \in C^*} \mathcal{L}^{(k)}(u^{(k+1)}, p) ,$$

On ne peut pas choisir un opérateur  $\Lambda(u, p)$  de type **Lagrangien** car le problème de maximisation en  $p$  serait alors **linéaire**...

# Algorithme du PPA « à deux niveaux »

II

On choisit un opérateur  $\Lambda$  de la forme :

$$\Lambda(u, p) = K(u) - \frac{1}{2\alpha} \|p\|^2 ,$$

Le sous-problème de **minimisation** en  $u$  est alors :

$$\begin{aligned} \min_{u \in U^{\text{ad}}} & K(u) + \langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}) , u \rangle \\ & + \epsilon \langle (\Theta'(u^{(k)}))^{\top} \cdot p^{(k)} , u \rangle + \epsilon J^{\Sigma}(u) + \epsilon \langle p^{(k)} , \Theta^{\Sigma}(u) \rangle , \end{aligned}$$

dont la solution est notée  $u^{(k+1)}$ .

Le sous-problème de **maximisation** en  $p$  est :

$$\max_{p \in C^*} -\frac{1}{2\alpha} \|p\|^2 + \langle p , \epsilon \Theta(u^{(k+1)}) + p^{(k)} / \alpha \rangle + \epsilon \langle p , \Theta^{\Sigma}(u^{(k+1)}) \rangle ,$$

dont la solution est :

$$p^{(k+1)} = \text{proj}_{C^*} \left( p^{(k)} + \epsilon \alpha (\Theta(u^{(k+1)}) + \Theta^{\Sigma}(u^{(k+1)})) \right) .$$

# Décomposition du PPA « à deux niveaux »

III

Dans cet algorithme du PPA à deux niveaux, la **décomposition** du problème de minimisation en  $u$  :

$$\min_{u \in U^{\text{ad}}} K(u) + \langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}), u \rangle \\ + \epsilon \langle (\Theta'(u^{(k)}))^{\top} \cdot p^{(k)}, u \rangle + \epsilon J^{\Sigma}(u) + \epsilon \langle p^{(k)}, \Theta^{\Sigma}(u) \rangle ,$$

par rapport à une décomposition de  $u$  en  $(u_1, \dots, u_N)$  signifie qu'il s'écrit comme  $N$  sous-problèmes d'optimisation **indépendants** ne dépendant chacun que d'un seul  $u_i$ . Pour cela, il faut **choisir**

- un noyau  **$K$  additif** (de même pour  $J^{\Sigma}$  et  $\Theta^{\Sigma}$ ) :

$$K(u) = \sum_{i=1}^N K_i(u_i) ,$$

- un ensemble  $U^{\text{ad}}$  qui s'écrit comme un produit cartésien  $U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}$  d'ensembles admissibles, avec  $u_i \in U_i^{\text{ad}}$ .

# Décomposition du PPA « à deux niveaux »

## IV

Avec ce **choix**, le sous-problème portant sur la variable  $i$  dans la **décomposition** du problème s'écrit :

$$\begin{aligned} \min_{u_i \in U_i^{\text{ad}}} & K_i(u_i) + \langle \epsilon \nabla_{u_i} J(u^{(k)}) - \nabla_{u_i} K(u^{(k)}), u_i \rangle \\ & + \epsilon \langle (\Theta'_{u_i}(u^{(k)}))^{\top} \cdot p^{(k)}, u_i \rangle + \epsilon J_i^{\Sigma}(u_i) + \epsilon \langle p^{(k)}, \Theta_i^{\Sigma}(u_i) \rangle, \end{aligned}$$

et sa résolution fournit une solution  $u_i^{(k+1)}$ . La mise à jour du **multiplicateur** est donnée par la relation :

$$p^{(k+1)} = \text{proj}_{C^*} \left( p^{(k)} + \epsilon \alpha (\Theta(u^{(k+1)}) + \Theta^{\Sigma}(u^{(k+1)})) \right).$$

On notera la **forte proximité** entre l'algorithme de décomposition par le **PPA à deux niveaux** et celui de décomposition par les **prix** : prendre  $\epsilon = 1$ ,  $K = J$ ,  $J^{\Sigma} = 0$  et  $\Theta = 0$  conduit à la **généralisation** au cas non additif de l'algorithme de décomposition par les **prix**.

- 1 Principe du Problème Auxiliaire avec contrainte
  - Rappel du PPA dans le cas sans contrainte
  - Application du PPA dans le cas avec contrainte
  - Algorithmes du PPA pour la recherche de point selle
- 2 Régularisation et Lagrangien augmenté
  - Transformée de Moreau-Yosida
  - Application à la minimisation et méthodes proximales
  - Application à la recherche de point selle
- 3 Travaux dirigés sur le Principe du Problème Auxiliaire
  - Un peu de calcul sous-différentiel...
  - Étude d'une vallée hydraulique
  - Réseau de distribution d'eau

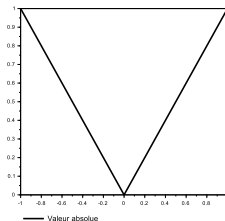
- 1 Principe du Problème Auxiliaire avec contrainte
  - Rappel du PPA dans le cas sans contrainte
  - Application du PPA dans le cas avec contrainte
  - Algorithmes du PPA pour la recherche de point selle
- 2 Régularisation et Lagrangien augmenté
  - Transformée de Moreau-Yosida
  - Application à la minimisation et méthodes proximales
  - Application à la recherche de point selle
- 3 Travaux dirigés sur le Principe du Problème Auxiliaire
  - Un peu de calcul sous-différentiel...
  - Étude d'une vallée hydraulique
  - Réseau de distribution d'eau

# Régularisée de Moreau-Yosida

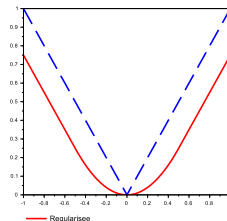
Soit  $\mathcal{U}$  un espace de Hilbert, soit une fonction  $J : \mathcal{U} \rightarrow \mathbb{R}$  et soit un coefficient  $c > 0$ . On appelle **régularisée de Moreau-Yosida** la fonction  $J_c : \mathcal{U} \rightarrow \mathbb{R}$  définie par :

$$J_c(u) = \min_{v \in \mathcal{U}} J(v) + \frac{1}{2c} \|v - u\|^2.$$

Cette définition implique que  $J_c(u) \leq J(u) \quad \forall u \in \mathcal{U}$ .



$$J(u) = |u|$$



$$J_c(u) = \begin{cases} -u - \frac{c}{2} & \text{si } u < -c \\ \frac{1}{2c} u^2 & \text{si } u \in [-c, c] \\ u - \frac{c}{2} & \text{si } u > c \end{cases}$$

# Régularisée de Moreau-Yosida

II

L'intérêt en optimisation de la **régularisée de Moreau-Yosida** d'une fonction  $J$  est donné par le théorème suivant ;

## Théorème

Soit une fonction  $J : \mathcal{U} \rightarrow \mathbb{R}$  que l'on suppose propre, convexe, s.c.i. et **sous-différentiable**, et soit un coefficient  $c > 0$ . Alors, sa **régularisée de Moreau-Yosida**  $J_c$  est telle que :

- 1 la fonction  $J_c$  est propre, convexe, s.c.i., **différentiable à gradient Lipschitzien**,
- 2 les minimums des fonctions  $J$  et  $J_c$  sont **égaux** et l'on a :

$$\arg \min_{u \in \mathcal{U}} J(u) = \arg \min_{u \in \mathcal{U}} J_c(u) ,$$

- 3 si  $J$  est fortement convexe, alors  $J_c$  est **fortement convexe**.



- 1 Principe du Problème Auxiliaire avec contrainte
  - Rappel du PPA dans le cas sans contrainte
  - Application du PPA dans le cas avec contrainte
  - Algorithmes du PPA pour la recherche de point selle
- 2 Régularisation et Lagrangien augmenté
  - Transformée de Moreau-Yosida
  - Application à la minimisation et méthodes proximales
  - Application à la recherche de point selle
- 3 Travaux dirigés sur le Principe du Problème Auxiliaire
  - Un peu de calcul sous-différentiel...
  - Étude d'une vallée hydraulique
  - Réseau de distribution d'eau

# Application à la minimisation

On considère alors le problème de minimisation :

$$\min_{u \in U^{\text{ad}} \subset \mathcal{U}} J(u) ,$$

qui s'écrit de manière équivalente :

$$\min_{u \in \mathcal{U}} J(u) + \chi_{U^{\text{ad}}}(u) ,$$

$$\text{avec } \chi_{U^{\text{ad}}}(u) = \begin{cases} 0 & \text{si } u \in U^{\text{ad}} \\ +\infty & \text{sinon} \end{cases} .$$

Par **abus de notation**, on note  $J_c$  la régularisée de  $J + \chi_{U^{\text{ad}}}$  :

$$\begin{aligned} J_c(u) &= \min_{v \in \mathcal{U}} J(v) + \chi_{U^{\text{ad}}}(v) + \frac{1}{2c} \|v - u\|^2 , \\ &= \min_{v \in U^{\text{ad}}} J(v) + \frac{1}{2c} \|v - u\|^2 . \end{aligned}$$

# Application à la minimisation

II

Par une application directe du **théorème** vu précédemment, on a

$$\begin{aligned}\min_{u \in \mathcal{U}} J_c(u) &= \min_{u \in U^{\text{ad}}} J(u) , \\ \arg \min_{u \in \mathcal{U}} J_c(u) &= \arg \min_{u \in U^{\text{ad}}} J(u) .\end{aligned}$$

- Plutôt que **minimiser** la fonction  $J$  sur  $U^{\text{ad}}$ , on préférerait donc **minimiser** sa régularisée  $J_c$  (**différentiable à gradient Lipschitzien**) sur tout l'espace  $\mathcal{U}$  (**pas de contrainte**).
- Mais une **évaluation** de  $J_c$  nécessite une **minimisation** de  $J$  sur  $U^{\text{ad}}$  ! Cette tranformation ne peut pas être utilisée de manière générale.
- Dans certains cas particuliers, elle peut être appliquée de manière **efficace**, donnant lieu aux méthodes dites **proximales**.

# Lien avec les méthodes proximales

I

Par définition, le **prox** d'une fonction  $J$  avec un coefficient  $c > 0$  est l'opérateur  $\text{prox}_{cJ} : \mathcal{U} \rightarrow \mathcal{U}$  définie par

$$\text{prox}_{cJ}(u) = \arg \min_{v \in \mathcal{U}} J(v) + \frac{1}{2c} \|v - u\|^2 .$$

Le lien avec la **régularisée de Moreau-Yosida** est immédiat. . .

Supposons alors que la fonction  $J$  soit **différentiable** et écrivons la **condition d'optimalité** du problème de minimisation définissant l'opérateur  $\text{prox}_{cJ}$  en un point  $u^{(k)}$  :

$$\nabla J(u^{(k+1)}) + \frac{1}{c} (u^{(k+1)} - u^{(k)}) = 0 ,$$

qui se réécrit sous la forme :

$$u^{(k+1)} = u^{(k)} - c \nabla J(u^{(k+1)}) .$$

# Lien avec les méthodes proximales

II

Cette dernière formule :

$$u^{(k+1)} = \text{prox}_{cJ}(u^{(k)}) = u^{(k)} - c \nabla J(u^{(k+1)}) ,$$

utilise un gradient **implicite**, et conduit à l'**algorithme de minimisation proximale**. Il est intéressant de la comparer à l'**algorithme de gradient standard** :

$$u^{(k+1)} = u^{(k)} - \rho \nabla J(u^{(k)}) ,$$

qui correspond à une formule de gradient **explicite**.

Les **méthodes proximales** s'utilisent aussi (surtout) lorsque la fonction  $J$  est **sous-différentiable**, par exemple dans les cas où :

- le calcul du **prox** de la fonction  $J$  se fait **analytiquement**,
- on utilise une **approximation linéaire** de la fonction  $J$ ,
- on minimise la somme de deux fonctions  $J$  (de **prox** connu) et  $G$  (**différentiable**) : algorithme du **gradient proximal**.

- 1 Principe du Problème Auxiliaire avec contrainte
  - Rappel du PPA dans le cas sans contrainte
  - Application du PPA dans le cas avec contrainte
  - Algorithmes du PPA pour la recherche de point selle
- 2 Régularisation et Lagrangien augmenté
  - Transformée de Moreau-Yosida
  - Application à la minimisation et méthodes proximales
  - Application à la recherche de point selle
- 3 Travaux dirigés sur le Principe du Problème Auxiliaire
  - Un peu de calcul sous-différentiel...
  - Étude d'une vallée hydraulique
  - Réseau de distribution d'eau

# Lagrangien et régularisation de la fonction duale

On considère le problème d'optimisation **sous contrainte** :

$$\min_{u \in U^{\text{ad}} \subset \mathcal{U}} J(u) \quad \text{sous} \quad \Theta(u) \in -C \subset \mathcal{V}.$$

Sous les hypothèses d'existence d'un **point selle** du Lagrangien  $L$  associé, on le résout en étudiant le problème **dual** :

$$\max_{p \in C^*} H(p),$$

où la fonction duale  $H$  est définie par :  $H(p) = \min_{u \in U^{\text{ad}}} L(u, p)$ .

La fonction  $H$  est propre, **concave**, **s.c.s.** et **sous-différentiable**.  
On introduit sa **régularisée de Moreau-Yosida** :

$$H_c(p) = \max_{q \in \mathcal{V}} H(q) - \chi_{C^*}(q) - \frac{1}{2c} \|q - p\|^2.$$

# Lagrangien augmenté

I

La **régularisée** de la fonction duale  $H$  s'écrit :

$$\begin{aligned} H_c(p) &= \max_{q \in \mathcal{V}} H(q) - \chi_{C^*}(q) - \frac{1}{2c} \|q - p\|^2, \\ &= \max_{q \in C^*} \min_{u \in U^{\text{ad}}} J(u) + \langle q, \Theta(u) \rangle - \frac{1}{2c} \|q - p\|^2, \\ &= \min_{u \in U^{\text{ad}}} \left( J(u) + \max_{q \in C^*} \langle q, \Theta(u) \rangle - \frac{1}{2c} \|q - p\|^2 \right). \end{aligned}$$

On définit alors le **Lagrangien augmenté**  $L_c$  par :

$$L_c(u, p) = J(u) + \max_{q \in C^*} \left( \langle q, \Theta(u) \rangle - \frac{1}{2c} \|q - p\|^2 \right),$$

de telle sorte que :

$$H_c(p) = \min_{u \in U^{\text{ad}}} L_c(u, p).$$



# Lagrangien augmenté

La fonction  $H_c$  est définie sur  $\mathcal{V}$  (alors que  $H$  l'est sur  $C^\star$ ), et le **Lagrangien augmenté**  $L_c$  est donc défini sur  $U^{\text{ad}} \times \mathcal{V}$  (alors que le Lagrangien  $L$  est défini sur  $U^{\text{ad}} \times C^\star$ ). De plus, la fonction  $H_c$  est **différentiable** à gradient Lipschitzien et on a :

$$\arg \max_{p \in C^\star} H(p) = \arg \max_{p \in \mathcal{V}} H_c(p) .$$

**Sans hypothèse de stricte convexité**, on a le résultat suivant.

## Théorème

- (a) Le **Lagrangien**  $L$  et le **Lagrangien augmenté**  $L_c$  admettent le même ensemble de points selle  $U^\# \times P^\#$ .
- (b) Le **Lagrangien augmenté**  $L_c$  est **stable** :

$$\forall p^\# \in P^\# , \arg \min_{u \in U^{\text{ad}}} L_c(u, p^\#) = U^\# .$$

# Lagrangien augmenté

Dans l'expression définissant le **Lagrangien augmenté** :

$$L_c(u, p) = J(u) + \max_{q \in C^\star} \left( \langle q, \Theta(u) \rangle - \frac{1}{2c} \|q - p\|^2 \right),$$

la maximisation en  $q$  porte sur une fonction **quadratique** : on peut la **calculer** de manière **analytique**.

- Le **max** en  $q$  est atteint en :

$$q^\# = \text{proj}_{C^\star}(p + c\Theta(u)).$$

- L'expression du **Lagrangien augmenté** est alors :

$$L_c(u, p) = J(u) + \frac{1}{2c} \left( \|\text{proj}_{C^\star}(p + c\Theta(u))\|^2 - \|p\|^2 \right),$$

son **gradient partiel** par rapport à  $p$  étant :

$$\nabla_p L_c(u, p) = \frac{1}{c} \left( \text{proj}_{C^\star}(p + c\Theta(u)) - p \right).$$

# Lagrangien augmenté

On peut alors écrire l'algorithme d'**Uzawa** appliqué à la recherche d'un point selle du problème d'optimisation sous contrainte et basé sur le **Lagrangien augmenté**. L'itération  $k$  de cet algorithme est composé des deux étapes suivantes :

- 1 **minimisation** en  $u$  à  $p = p^{(k)}$  fixé :

$$\min_{u \in U^{\text{ad}}} L_c(u, p^{(k)}) \iff \min_{u \in U^{\text{ad}}} J(u) + \frac{1}{2c} \|\text{proj}_{C^*}(p^{(k)} + c\Theta(u))\|^2 ,$$

- 2 **mise à jour** du multiplicateur  $p$  :

$$\begin{aligned} p^{(k+1)} &= p^{(k)} + \rho \nabla_p L_c(u^{(k+1)}, p^{(k)}) \\ &= \left(1 - \frac{\rho}{c}\right) p^{(k)} + \frac{\rho}{c} \text{proj}_{C^*}(p^{(k)} + c\Theta(u^{(k+1)})) . \end{aligned}$$

La **convergence** de cet algorithme est prouvée dans le cadre convexe, **sans hypothèse de stricte ou forte convexité**.

## Exemple : le cas de contrainte égalité

I

Considérons le cas de contrainte **égalité** :

$$\Theta(u) = 0 .$$

Le **Lagrangien augmenté** :

$$L_c(u, p) = J(u) + \frac{1}{2c} \left( \|\text{proj}_{C^*}(p + c\Theta(u))\|^2 - \|p\|^2 \right) ,$$

s'écrit alors :

$$L_c(u, p) = J(u) + \langle p, \Theta(u) \rangle + \frac{c}{2} \|\Theta(u)\|^2 .$$

Les deux premiers termes correspondent au **Lagrangien ordinaire**.  
Le dernier terme est parfois interprété comme une **pénalisation**  
de la contrainte égalité : c'est une interprétation **trompeuse**  
car il correspond en fait à une **régularisation**.

## Exemple : le cas de contrainte égalité

II

Dans le cas d'un problème à **structure additive** :

$$J(u) = \sum_{i=1}^N J_i(u_i) \quad , \quad \Theta(u) = \sum_{i=1}^N \Theta_i(u_i) \quad ,$$

Le **Lagrangien augmenté** s'écrit :

$$\sum_{i=1}^N \left( J_i(u_i) + \langle p, \Theta_i(u_i) \rangle \right) + \frac{c}{2} \left\| \sum_{i=1}^N \Theta_i(u_i) \right\|^2 .$$

Il n'est pas **additif** et sa minimisation en  $u$  ne se **décompose**  $u_i$  par  $u_i$  : la **décomposition par les prix** ne fonctionne donc pas. . .

Mais on sait **décomposer** un problème non additif grâce au **PPA**,  
et on peut donc décomposer la recherche d'un point selle du  
**Lagrangien augmenté** !

# Interprétation géométrique

I

On a vu que, pour une contrainte **égalité**, l'**algorithme d'Uzawa** :

$$u^{(k+1)} = \arg \min_{u \in U^{\text{ad}}} J(u) + \langle p^{(k)}, \Theta(u) \rangle ,$$

$$p^{(k+1)} = p^{(k)} + \rho \Theta(u^{(k+1)}) ,$$

se réécrit à l'aide de la **fonction de perturbation**  $G$  :<sup>14</sup>

$$v^{(k+1)} = \arg \min_{v \in \mathcal{V}} G(v) + \langle p^{(k)}, v \rangle ,$$

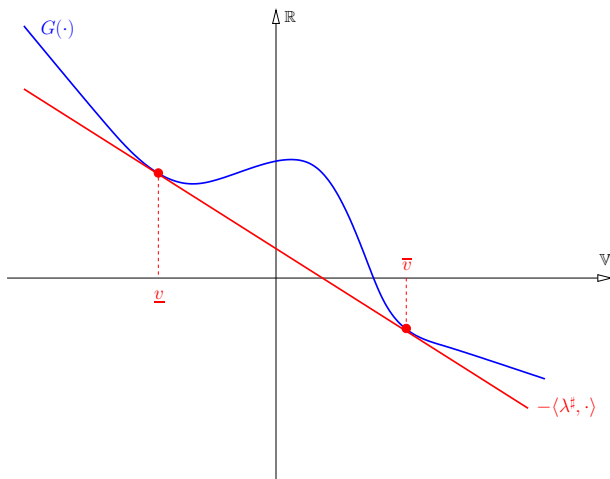
$$p^{(k+1)} = p^{(k)} + \rho v^{(k+1)} ,$$

ce qui permet d'en donner une **interprétation géométrique**.

14. avec  $G(v) = \min_{u \in U^{\text{ad}}} J(u)$  sous  $\Theta(u) - v = 0$

# Interprétation géométrique

II



# Interprétation géométrique



Dans le cas du **Lagrangien augmenté**, l'**algorithme d'Uzawa** est :

$$\begin{aligned} u^{(k+1)} &= \arg \min_{u \in U^{\text{ad}}} J(u) + \langle p^{(k)}, \Theta(u) \rangle + \frac{c}{2} \|\Theta(u)\|^2, \\ p^{(k+1)} &= p^{(k)} + \rho \Theta(u^{(k+1)}), \end{aligned}$$

et il se réécrit à l'aide de la **fonction de perturbation G** :

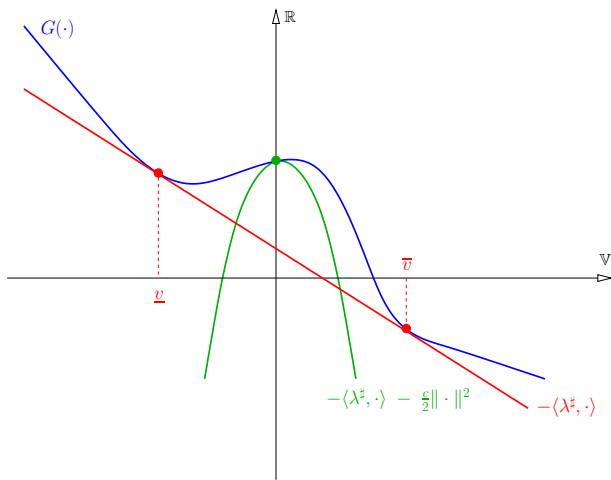
$$\begin{aligned} v^{(k+1)} &= \arg \min_{v \in \mathcal{V}} G(v) + \langle p^{(k)}, v \rangle + \frac{c}{2} \|v\|^2, \\ p^{(k+1)} &= p^{(k)} + \rho v^{(k+1)}, \end{aligned}$$

d'où son **interprétation géométrique**.



## Interprétation géométrique

## IV



## Conclusion du cours

Dans ce cours, on a vu beaucoup d'outils et de méthodes permettant de **décomposer** les grands problèmes d'optimisation dans le cadre classique **convexe déterministe**. Dans ce cadre, la décomposition porte sur des **vecteurs**  $u$  d'un espace  $\mathcal{U}$ .

Mais ces méthodes ne s'appliquent pas à tous les problèmes d'optimisation : ainsi, dans le cas du **contrôle optimal stochastique**, les variables de décision sont des feedbacks :

$$u : \mathcal{X} \longrightarrow \mathcal{U},$$

et la **décomposition** devrait porter **à la fois** sur l'espace  $\mathcal{U}$  et l'espace  $\mathcal{X}$  sur lequel les feedbacks sont définis : cela engendre des difficultés très importantes, hors du cadre de ce cours.

Ceci étant, les mécanismes vus sont génériques et permettent d'attaquer de nombreux problèmes, aussi bien dans le cadre déterministe que dans le cadre stochastique.

- 1 Principe du Problème Auxiliaire avec contrainte
  - Rappel du PPA dans le cas sans contrainte
  - Application du PPA dans le cas avec contrainte
  - Algorithmes du PPA pour la recherche de point selle
- 2 Régularisation et Lagrangien augmenté
  - Transformée de Moreau-Yosida
  - Application à la minimisation et méthodes proximales
  - Application à la recherche de point selle
- 3 Travaux dirigés sur le Principe du Problème Auxiliaire
  - Un peu de calcul sous-différentiel...
  - Étude d'une vallée hydraulique
  - Réseau de distribution d'eau

- 1 Principe du Problème Auxiliaire avec contrainte
  - Rappel du PPA dans le cas sans contrainte
  - Application du PPA dans le cas avec contrainte
  - Algorithmes du PPA pour la recherche de point selle
- 2 Régularisation et Lagrangien augmenté
  - Transformée de Moreau-Yosida
  - Application à la minimisation et méthodes proximales
  - Application à la recherche de point selle
- 3 Travaux dirigés sur le Principe du Problème Auxiliaire
  - Un peu de calcul sous-différentiel...
  - Étude d'une vallée hydraulique
  - Réseau de distribution d'eau

# Régularisée de la fonction valeur absolue

E<sub>1</sub>

On considère la fonction réelle **valeur absolue** :

$$\begin{aligned} J : \mathbb{R} &\longrightarrow \mathbb{R} \\ u &\longrightarrow |u| , \end{aligned}$$

et sa **régularisée de Moreau-Yosida** :

$$J_c(u) = \min_{v \in \mathbb{R}} |v| + \frac{1}{2c} \|v - u\|^2 .$$

- 1 Calculer analytiquement la régularisée  $J_c$ .
- 2 Donner l'expression de l'opérateur  $\text{prox}_{cJ}$ .

# Régularisée de la fonction valeur absolue

R<sub>1</sub>

On note :

$$F(v) = |v| + \frac{1}{2c} \|v - u\|^2 .$$

On calcule le **sous-différentiel** de la fonction  $F$  :

$$\partial F(v) = \begin{cases} -1 + \frac{1}{c}(v - u) & \text{si } v < 0 , \\ [-1, 1] - \frac{u}{c} & \text{si } v = 0 , \\ 1 + \frac{1}{c}(v - u) & \text{si } v > 0 . \end{cases}$$

La solution  $v^\#$  associée au calcul de la régularisée  $J_c$  est caractérisée par la **condition d'optimalité** :  $0 \in \partial F(v^\#)$ .

## Régularisée de la fonction valeur absolue

R<sub>2</sub>

- Cas  $v^\sharp < 0$  :

$$0 \in \partial F(v^\sharp) \iff v^\sharp = u + c, \text{ et donc } u < -c.$$

- Cas  $v^\sharp = 0$  :

$$0 \in \partial F(v^\sharp) \iff u \in [-c, c].$$

- Cas  $v^\sharp > 0$  :

$$0 \in \partial F(v^\sharp) \iff v^\sharp = u - c, \text{ et donc } u > c.$$

On obtient donc :

$$\text{prox}_{cJ}(u) = \begin{cases} u + c & \text{si } u < -c \\ 0 & \text{si } u \in [-c, c] \\ u - c & \text{si } u > c \end{cases}, \quad J_c(u) = \begin{cases} -u - \frac{c}{2} & \text{si } u < -c \\ \frac{u^2}{2c} & \text{si } u \in [-c, c] \\ u - \frac{c}{2} & \text{si } u > c \end{cases}.$$

- 1 Principe du Problème Auxiliaire avec contrainte
  - Rappel du PPA dans le cas sans contrainte
  - Application du PPA dans le cas avec contrainte
  - Algorithmes du PPA pour la recherche de point selle
- 2 Régularisation et Lagrangien augmenté
  - Transformée de Moreau-Yosida
  - Application à la minimisation et méthodes proximales
  - Application à la recherche de point selle
- 3 Travaux dirigés sur le Principe du Problème Auxiliaire
  - Un peu de calcul sous-différentiel...
  - Étude d'une vallée hydraulique
  - Réseau de distribution d'eau



# Optimisation d'une vallée hydraulique

E<sub>1</sub>

Le problème de l'optimisation d'une **vallée hydraulique** a été étudié lors des séances d'exercices précédentes et s'écrit :

$$\min_{(x_{i,t}, u_{i,t})_{(i,t) \in \mathcal{I} \times \mathcal{T}}} \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} L_{i,t}(x_{i,t}, u_{i,t}) ,$$

sous  $x_{i,t+1} = f_{i,t}(x_{i,t}, u_{i,t}, u_{i-1,t}) \quad \forall i \in \mathcal{I}, \quad \forall t \in \mathcal{T} .$

Pour obtenir une **formulation compacte**, on introduit les vecteurs :

$$x_i = (x_{i,t})_{t \in \mathcal{T}} \quad , \quad u_i = (u_{i,t})_{t \in \mathcal{T}} .$$

L'ensemble des équations décrivant la **dynamique** du barrage  $i$  au cours du temps s'écrit sous forme vectorielle :

$$x_i = F_i(u_i, u_{i-1}) .$$

Substituant cette expression de  $x_i$  dans le **critère**, on définit :

$$J_i(u_i, u_{i-1}) = \sum_{t \in \mathcal{T}} L_{i,t}(x_{i,t}, u_{i,t})$$

# Optimisation d'une vallée hydraulique

E<sub>2</sub>

Finalement, le **problème de minimisation de la vallée hydraulique** se met sous la forme compacte suivante :

$$\min_{(u_i)_{i \in \mathcal{I}}} \sum_{i \in \mathcal{I}} J_i(u_i, u_{i-1}) .$$

On souhaite appliquer la **décomposition par le PPA** à ce problème, avec une décomposition **barrage par barrage** ( $i$  par  $i$ ).

- 1 Écrire l'algorithme du PPA associé à ce problème avec le choix de **noyaux additifs** suivant :

$$K^{(k)}((u_i)_{i \in \mathcal{I}}) = \sum_{i \in \mathcal{I}} J_i(u_i, u_{i-1}^{(k)}) ,$$

et donner la formulation **compacte** du sous-problème  $i$ .

- 2 À quelle condition peut-on choisir les  $\epsilon^{(k)}$  **égaux** à 1 ?
- 3 Donner la formulation **détaillée** du problème auxiliaire associé au barrage  $i$  lorsque  $\epsilon^{(k)} = 1$ .

# Optimisation d'une vallée hydraulique

R<sub>1</sub>

On écrit le **problème auxiliaire** à l'itération  $k$  :

$$\min_{u \in U^{\text{ad}}} K^{(k)}(u) + \langle \epsilon^{(k)} \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}), u \rangle + \epsilon^{(k)} J^{\Sigma}(u),$$

avec  $J^{\Sigma} \equiv 0$ , ce qui donne :

$$\begin{aligned} \min_{(u_i)_{i \in \mathcal{I}}} \quad & \sum_{i \in \mathcal{I}} J_i(u_i, u_{i-1}^{(k)}) \\ & + \sum_{i \in \mathcal{I}} \left\langle (\epsilon^{(k)} - 1) \nabla_{u_i} J_i(u_i^{(k)}, u_{i-1}^{(k)}), u_i \right\rangle \\ & + \sum_{i \in \mathcal{I}} \left\langle \epsilon^{(k)} \nabla_{u_i} J_{i+1}(u_{i+1}^{(k)}, u_i^{(k)}), u_i \right\rangle. \end{aligned}$$

La solution de ce **problème auxiliaire** :

$$\left( (u_i^{(k+1)})_{i \in \mathcal{I}} \right),$$

permet de formuler le problème auxiliaire à l'itération  $k + 1$ .

# Optimisation d'une vallée hydraulique

R<sub>2</sub>

On constate que le choix de **noyaux additifs** que l'on a fait conduit à la minimisation d'une somme de fonctions ne dépendant chacune que d'un seul  $i$ , et donc à la **décomposition** du problème auxiliaire. Le **sous-problème** associé au barrage  $i$  est :

$$\min_{u_i} J_i(u_i, u_{i-1}^{(k)}) + \left\langle (\epsilon^{(k)} - 1) \nabla_{u_i} J_i(u_i^{(k)}, u_{i-1}^{(k)}), u_i \right\rangle \\ + \left\langle \epsilon^{(k)} \nabla_{u_i} J_{i+1}(u_{i+1}^{(k)}, u_i^{(k)}), u_i \right\rangle .$$

Pouvoir choisir  $\epsilon^{(k)} = 1$  dépend de la **forte convexité** des  $J_i$ , puisque on doit satisfaire la condition :  $\epsilon^{(k)} < 2b^{(k)}/A$ . Si cette condition n'est pas remplie, ajouter dans le **sous-problème**  $i$  un terme quadratique  $\gamma_i \|u_i - u_i^{(k)}\|^2$  permettra de prendre  $\epsilon^{(k)} = 1$  pour un  $\gamma_i$  suffisamment grand.

# Optimisation d'une vallée hydraulique

R<sub>3</sub>

On se place dans le cas  $\epsilon^{(k)} = 1$ . Alors, le **sous-problème auxiliaire** associé au barrage  $i$  se simplifie et devient :

$$\min_{u_i \in \mathbb{R}^{|\mathcal{T}|}} J_i(u_i, u_{i-1}^{(k)}) + \langle \nabla_{u_i} J_{i+1}(u_{i+1}^{(k)}, u_i^{(k)}), u_i \rangle .$$

Remplaçant  $J_i$  par son expression **détaillée**, il se réécrit :

$$\min_{(x_{i,t}, u_{i,t})_{t \in \mathcal{T}}} \sum_{t \in \mathcal{T}} \left( L_{i,t}(x_{i,t}, u_{i,t}) + \langle p_{i+1,t}^{(k)}, u_{i,t} \rangle \right) ,$$

$$\text{sous } x_{i,t+1} = f_{i,t}(x_{i,t}, u_{i,t}, u_{i-1,t}^{(k)}) \quad \forall t \in \mathcal{T} .$$

avec :  $p_{i+1,t}^{(k)} = \nabla_{u_{i,t}} J_{i+1}(u_{i+1}^{(k)}, u_i^{(k)})$ . Le calcul du gradient donne :

$$\nabla_{u_{i,t}} J_{i+1}(u_{i+1}^{(k)}, u_i^{(k)}) = \left[ \frac{\partial}{\partial u_{i,t}} f_{i+1,t}(x_{i+1,t}^{(k)}, u_{i+1,t}^{(k)}, u_{i,t}^{(k)}) \right]^T \zeta_{i+1,t}^{(k)} ,$$

où  $\zeta_{i+1,t}^{(k)}$  est le **multiplicateur** de la dynamique  $i+1$ . On retrouve ainsi exactement la méthode de **décomposition par prédiction**.

- 1 Principe du Problème Auxiliaire avec contrainte
  - Rappel du PPA dans le cas sans contrainte
  - Application du PPA dans le cas avec contrainte
  - Algorithmes du PPA pour la recherche de point selle
- 2 Régularisation et Lagrangien augmenté
  - Transformée de Moreau-Yosida
  - Application à la minimisation et méthodes proximales
  - Application à la recherche de point selle
- 3 Travaux dirigés sur le Principe du Problème Auxiliaire
  - Un peu de calcul sous-différentiel...
  - Étude d'une vallée hydraulique
  - Réseau de distribution d'eau

# Décomposition d'un réseau d'eau par le PPA

E<sub>1</sub>

On rappelle la **formulation compacte** du problème d'optimisation du grand **réseau d'eau connecté** :

$$\min_{(u_{i,1}, u_{i,2})_{i=1, \dots, N}} \sum_{i=1}^N J_i(u_{i,1}, u_{i,2}) + J_{N+1} \left( \sum_{i=1}^N u_{i,1}, \sum_{i=1}^N u_{i,2} \right),$$

avec les contraintes de bornes :

$$(u_{i,1}, u_{i,2}) \in U_i^{\text{ad}} = [0, \bar{v}_{i,1}] \times [0, \bar{v}_{i,2} - \bar{v}_{i,1}], \quad i = 1, \dots, N.$$

Pour  $i = 1, \dots, N$ , l'**expression détaillée** de la fonction  $J_i$  est :

$$J_i(u_{i,1}, u_{i,2}) = \min_{(v_{i,1}, v_{i,2})} \frac{1}{2} (a_{i,1} v_{i,1}^2 + a_{i,2} v_{i,2}^2),$$

$$\text{sous } u_{i,1} + v_{i,1} \geq \bar{v}_{i,1},$$

$$u_{i,1} + u_{i,2} + v_{i,1} + v_{i,2} - \bar{v}_{i,2} = 0,$$

$$\text{ainsi que : } J_{N+1}(u_{N+1,1}, u_{N+1,2}) = \frac{1}{2} (a_{N+1,1} u_{N+1,1}^2 + a_{N+1,2} u_{N+1,2}^2).$$

# Décomposition d'un réseau d'eau par le PPA

E<sub>2</sub>

On étudie les **deux choix** suivants dans l'algorithme du PPA.

## ❶ *Choix No 1*

$$J(u) = \sum_{i=1}^N J_i(u_i) + J_{N+1}\left(\sum_{i=1}^N u_i\right), \quad K^{(k)}(u) = \sum_{i=1}^N J_i(u_i), \quad \epsilon^{(k)} = 1.$$

## ❷ *Choix No 2*

$$J(u) = \sum_{i=1}^N J_i(u_i) + J_{N+1}\left(\sum_{i=1}^N u_i\right), \quad K^{(k)}(u) = \sum_{i=1}^N J_{N+1}\left(u_i + \sum_{j \neq i} u_j^{(k)}\right), \quad \epsilon^{(k)} = 1.$$

Pour chacun de ces deux choix,

- ❶ écrire l'algorithme du PPA sur la **formulation compacte**,
- ❷ dire à quelle méthode de décomposition il s'apparente,
- ❸ donner la **formulation détaillée** de chaque **sous-problème** *i*.



# Décomposition d'un réseau d'eau par le PPA

$R_1$

Avec le **choix No 1**, les différences entre les dérivées de  $J$  et  $K^{(k)}$  s'annulent pour  $\epsilon^{(k)} = 1$ , **sauf** pour les termes concernant  $J_{N+1}$ . Les dérivées partielles par rapport à  $u_i$  étant toutes égales entre elles, le **problème auxiliaire** s'écrit :

$$\min_{(u_i \in U_i^{\text{ad}})_{i=1, \dots, N}} \sum_{i=1}^N J_i(u_i) + \sum_{i=1}^N \left\langle \nabla J_{N+1} \left( \sum_{j=1}^N u_j^{(k)} \right), u_i \right\rangle .$$

Avec la notation  $p_{N+1}^{(k)} = \nabla J_{N+1} \left( \sum_{j=1}^N u_j^{(k)} \right)$ , ce problème se **décompose** en  $N$  sous-problèmes dont la  $i$ -ème instance est :

$$\min_{u_i \in U_i^{\text{ad}}} J_i(u_i) + \langle p_{N+1}^{(k)}, u_i \rangle .$$

On retrouve l'algorithme de **décomposition par prédiction** dans le cas où les contraintes couplantes sont affectées au  $(N+1)$ -ème sous-problème.

# Décomposition d'un réseau d'eau par le PPA

R<sub>2</sub>

Avec le **choix No 2** et  $\epsilon^{(k)} = 1$ , les différences entre les dérivées de  $J$  et  $K^{(k)}$  s'annulent seulement pour les termes concernant  $J_{N+1}$ .

Le **problème auxiliaire** s'écrit :

$$\min_{(u_i \in U_i^{\text{ad}})_{i=1, \dots, N}} \sum_{i=1}^N J_{N+1} \left( u_i + \sum_{j \neq i} u_j^{(k)} \right) + \sum_{i=1}^N \left\langle \nabla J_i(u_i(k)), u_i \right\rangle.$$

Introduisant la notation :  $p_i^{(k)} = \nabla J_i(u_i^{(k)})$  ce problème se **décompose** en  $N$  sous-problèmes dont la  $i$ -ème instance est :

$$\min_{u_i \in U_i^{\text{ad}}} J_{N+1} \left( u_i + \sum_{j \neq i} u_j^{(k)} \right) + \langle p_i^{(k)}, u_i \rangle.$$

Dans cet algorithme, on compare le coût du sous-réseau  $N + 1$  au coût linéarisé du sous-réseau  $i$  autour du point  $u_i^{(k)}$ . Cet algorithme est à comparer à celui obtenu avec le choix précédent.