



**HAL**  
open science

## Optimisation des grands systèmes

Pierre Carpentier

► **To cite this version:**

Pierre Carpentier. Optimisation des grands systèmes : Méthodes de décomposition/coordination dans le cadre déterministe. Master. France. 2023, pp.81. hal-04072326

**HAL Id: hal-04072326**

**<https://hal.science/hal-04072326>**

Submitted on 18 Apr 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# OPTIMISATION DES GRANDS SYSTÈMES



## *Méthodes de décomposition/coordination dans le cadre déterministe*

Pierre Carpentier

`pierre.carpentier@free.fr`

17 avril 2023

### Résumé

Le cours « Optimisation des grands systèmes » a été donné durant de très nombreuses années à l'ENSTA comme enseignement de troisième année dans le parcours dédié à l'approfondissement en optimisation et recherche opérationnelle.

Ces notes sont données en plus du polycopié [1] ([disponible sur Internet](#)), mais ne s'y substituent pas. Elles constituent le support du cours et ont pour but :

- de permettre de suivre aisément la présentation orale du cours,
- de fournir aux étudiants l'essentiel de ce qu'il faut retenir du cours,
- de donner les énoncés et les corrigés des exercices traités en travaux dirigés.

Le livre [2] reprend et détaille l'ensemble des notions de ce cours, et présente l'extension de ces notions au cas stochastique.

### Plan du cours

- (1) *Motivations et rappels d'optimisation* (1 séance).

Dans cette introduction, on cherche à montrer l'intérêt d'un cours sur l'optimisation des grands systèmes et on effectue les rappels d'optimisation nécessaires pour la suite du cours.

- (2) *Introduction intuitive à la décomposition/coordination* (2 séances).

Dans cette partie du cours, on cherche à introduire les idées de décomposition/coordination à l'aide d'un modèle simple, sans préoccupation de rigueur mathématique ni de généralité. Le but est de faire ressortir d'un point de vue pratique les différences entre les divers schémas de résolution et de développer les interprétations de ces schémas.

- (3) *Formalisation : le principe du problème auxiliaire* (2 séances).

Dans cette partie, on présente la théorie permettant de mettre en œuvre les méthodes de décomposition/coordination dans un cadre général et d'étudier la convergence des algorithmes associés. On présente enfin le Lagrangien augmenté et comment le décomposer.

- (4) *Contrôle des connaissances* (1 séance).

Il consiste en un examen écrit dans lequel on demande de savoir mettre en œuvre sur un problème (plus ou moins) concret les différents schémas de décomposition/coordination étudiés durant le cours.

# Table des matières

<b>1</b>	<b>Motivation du cours et rappels</b>	<b>3</b>
1.1	Présentation des “grands systèmes”	3
1.2	Problèmes prototypes	4
1.3	Rappels d’optimisation convexe et de dualité	6
1.4	Quelques références bibliographiques	13
1.5	Exercices	15
1.6	Corrigés	17
<b>2</b>	<b>Point de vue intuitif en décomposition/coordination</b>	<b>25</b>
2.1	Problème étudié	25
2.2	Décomposition par les prix	25
2.3	Décomposition par allocation de ressources	27
2.4	Décomposition par prédiction	31
2.5	Exercices	40
2.6	Corrigés	42
<b>3</b>	<b>Principe du problème auxiliaire et applications</b>	<b>52</b>
3.1	Introduction	52
3.2	Optimisation sur un ensemble admissible	53
3.3	Optimisation sous contraintes explicites	62
3.4	Exercices	70
3.5	Corrigés	72
<b>4</b>	<b>Régularisation et Lagrangien augmenté</b>	<b>74</b>
4.1	Régularisée de Moreau-Yosida	74
4.2	Application à la minimisation	75
4.3	Mise en œuvre sur le problème dual	76

## Références

- [1] G. Cohen. *Optimisation des grands systèmes. DEA MMME de l’Université Paris I, 2004.*  
<https://www.rocq.inria.fr/metalau/cohen/documents/ParisIcours-A4-NB.pdf>.
- [2] P. Carpentier and G. Cohen. *Décomposition-coordination en optimisation déterministe et stochastique. Springer-Verlag, Berlin, 2017.*

# 1 Motivation du cours et rappels

## 1.1 Présentation des “grands systèmes”

### 1.1.1 Qu’est-ce qu’un grand système ?

Du point de vue de l’*optimisation*, un système sera qualifié de “grand” si les techniques classiques de l’optimisation ne peuvent s’y appliquer directement pour des questions de taille :

- grand nombre de variables et/ou de contraintes,
- hétérogénéité spatiale et/ou temporelle,
- nombre de décideurs intervenant sur le système.

Cette impossibilité est d’ordre méthodologique. Elle correspond souvent à une augmentation rapide des ressources (temps-calcul, mémoire) nécessaires à la résolution du problème en fonction de la dimension de ce dernier. Ainsi, la complexité de la méthode de Programmation Dynamique croît *exponentiellement* avec le nombre de variables d’état du système dynamique associé, ce qui limite en pratique à 4 ou 5 la dimension des systèmes que l’on est capable résoudre par cette méthode. Il en va de même pour la résolution directe des conditions d’optimalité sous contraintes de type inégalité.

**Quelques exemples :** réseaux de production/distribution (eau, énergie, ...) à l’échelle locale, nationale, européenne ; grands ensembles de production (raffinerie, ...).

### 1.1.2 Pourquoi optimiser un grand système ?

En général, les grands systèmes sont tels que :

- ils sont difficilement (ou mal) “gérables à la main”,
- on peut dans la plupart des cas leur associer un vrai critère économique,
- un faible gain en pourcentage sur leur coût peut s’avérer très important en valeur absolue.

### 1.1.3 Comment optimiser un grand système ?

On se limite aux systèmes correspondant à la situation suivante :

- toute l’information est disponible,
- un seul critère doit être minimisé,
- un seul décideur agit sur le système.

On notera que le dernier point correspond en particulier à la situation *déterministe* (pas d’aléas dans le fonctionnement du système). Ce n’est pas très réaliste dans la mesure où les grands systèmes sont “dans la vraie vie” presque toujours stochastiques, mais il faut commencer par le commencement. . .

Dans ce cadre, on cherche à décomposer le système initial en sous-systèmes de taille plus petite (du point de vue de l’optimisation). On se propose alors :

- de formuler pour chacun des sous-systèmes un sous-problème d’optimisation, et de le résoudre à l’aide des techniques de l’optimisation classique : c’est la phase de *décomposition*, dont les avantages sont multiples (taille et homogénéité de chaque sous-problème, adaptation de la méthode de résolution, parallélisation),
- de mettre en place un processus d’échange d’informations entre les sous-systèmes, basé sur les solutions de ces sous-problèmes : c’est la phase de *coordination* dont le but est d’assurer que la juxtaposition des solutions des sous-systèmes fournit la solution du système initial.

On notera les deux points suivants.

1. La décomposition/coordination est un processus *itératif*. Une itération consiste, d’une part à formuler et à résoudre les problèmes d’optimisation des sous-systèmes en fonction des in-

formations disponibles à cette itération, et d'autre part à remettre à jour les informations disponibles en fonction des solutions de l'ensemble des sous-systèmes.

2. Du point de vue de la coordination, seule compte la mise à jour des informations disponibles, et non la manière de les obtenir. On est donc libre du choix des méthodes utilisées lors de l'optimisation des sous-systèmes (résolution directe, énumération, méthodes de gradient, ...).

Pour ce type d'algorithme, l'expérience montre qu'il faut disposer d'un cadre théorique approprié, faute de quoi on risque de mettre en œuvre des processus fournissant des solutions incorrectes (voir les exercices 1.1, 1.2 et 1.3 du polycopié du cours). Il en est d'ailleurs de même pour l'optimisation en général, dont les applications pratiques vont bien au delà du cadre dans lequel cette discipline est enseignée. La théorie n'est alors pas le moule dans lequel doit se couler l'application, mais plutôt le guide permettant de choisir parmi des méthodes celle qui sera le mieux adaptée à cette application.

## 1.2 Problèmes prototypes

### 1.2.1 Exemple canonique

Une entreprise possède  $N$  unités de production. Les variables de décision de l'unité de production  $i$  sont notées  $u_i$ . Chaque variable  $u_i$  doit respecter un certain nombre de contraintes se traduisant par la relation  $u_i \in U_i^{\text{ad}}$ . La production et le coût associés à  $u_i$  sont notés respectivement  $\Theta_i(u_i)$  et  $J_i(u_i)$ . L'objectif de l'entreprise est de minimiser son coût total de production, sous la contrainte que sa production totale soit égale à  $\theta$ . La formulation du problème est donc :

$$\min_{u_i \in U_i^{\text{ad}}} \sum_{i=1}^N J_i(u_i), \quad (1a)$$

sous la contrainte :

$$\sum_{i=1}^N \Theta_i(u_i) = \theta. \quad (1b)$$

Dans ce premier exemple, le critère et la contrainte sont *additifs* en  $i$ . On verra que c'est là une situation particulièrement favorable pour les méthodes de décomposition/coordination.

### 1.2.2 Constitution d'un grand système

On considère le cas de l'optimisation du coût de fonctionnement d'un réseau de distribution d'eau sur une période de 24 heures. On distingue dans cette période 2 pas de temps, chacun d'eux correspondant à un tarif différent de l'énergie électrique (jour/nuit).

À l'origine, ce réseau est constitué de  $N$  sous-réseaux *indépendants*. Chaque sous-réseau  $i$  comporte une usine de refoulement injectant la quantité d'eau  $v_{i,t}$  au pas de temps  $t$  pour un coût quadratique  $\frac{1}{2}a_{i,t}v_{i,t}^2$ , la constante  $a_{i,t}$  (strictement positive) représentant le coût unitaire de l'énergie pour le sous-réseau  $i$  au pas de temps  $t$ . Chaque sous-réseau comporte de plus un réservoir, et on impose que le niveau du réservoir reste positif à la fin du premier pas de temps, et soit égal à son niveau initial à la fin du second pas de temps. Le problème d'optimisation associé se met sous la forme :

$$\min_{(v_{i,1}, v_{i,2})} \frac{1}{2} (a_{i,1}v_{i,1}^2 + a_{i,2}v_{i,2}^2), \quad (2a)$$

sous les contraintes :

$$v_{i,1} \geq \bar{v}_{i,1} , \quad (2b)$$

$$v_{i,1} + v_{i,2} = \bar{v}_{i,2} . \quad (2c)$$

Pour des raisons de cohérence, on supposera que les constantes  $\bar{v}_{i,1}$  et  $\bar{v}_{i,2}$  vérifient :

$$\bar{v}_{i,2} \geq \bar{v}_{i,1} \geq 0 .$$

Pour optimiser le réseau global, il suffit de résoudre le problème de minimisation associé à chacun des sous-réseaux, car il n'y a aucun couplage.

Pour des raisons diverses (coût, qualité, sécurité), on décide de construire une nouvelle usine de refoulement permettant d'alimenter *simultanément* les  $N$  sous-réseaux précédents. Notant  $u_{i,t}$  la quantité d'eau fournie au sous-réseau  $i$  pour le pas de temps  $t$  par cette nouvelle usine, le problème d'optimisation du sous-réseau  $i$  devient :

$$\min_{(v_{i,1}, v_{i,2})} \frac{1}{2} (a_{i,1} v_{i,1}^2 + a_{i,2} v_{i,2}^2) , \quad (3a)$$

sous les contraintes :

$$u_{i,1} + v_{i,1} \geq \bar{v}_{i,1} , \quad (3b)$$

$$u_{i,1} + u_{i,2} + v_{i,1} + v_{i,2} = \bar{v}_{i,2} . \quad (3c)$$

Le *résultat* de la minimisation en  $(v_{i,1}, v_{i,2})$  du problème (3) dépend des variables  $(u_{i,1}, u_{i,2})$ , et on note  $J_i(u_{i,1}, u_{i,2})$  la valeur du minimum. La quantité globale d'eau fournie au pas de temps  $t$  par la nouvelle usine est égale à  $\sum_{i=1}^N u_{i,t}$ . On note  $J_{N+1}$  la fonction de coût associée à cette usine, et le problème de l'optimisation du réseau s'écrit :

$$\min_{\{(u_{i,1}, u_{i,2})\}_{i=1}^N} J_{N+1} \left( \sum_{i=1}^N u_{i,1}, \sum_{i=1}^N u_{i,2} \right) + \sum_{i=1}^N J_i(u_{i,1}, u_{i,2}) . \quad (4)$$

On constate la présence d'un couplage par le critère entre les sous-réseaux. On se ramène à la forme de l'exemple canonique (1) en introduisant des variables supplémentaires  $u_{N+1,t}$  représentant la quantité d'eau globale fournie au pas de temps  $t$  par la nouvelle usine. Le problème (4) se met alors sous la forme :

$$\min_{\{(u_{i,1}, u_{i,2})\}_{i=1}^{N+1}} \sum_{i=1}^{N+1} J_i(u_{i,1}, u_{i,2}) , \quad (5a)$$

sous les contraintes :

$$\sum_{i=1}^N u_{i,t} - u_{N+1,t} = 0 \quad , \quad t = 1, 2 . \quad (5b)$$

Cet exemple sera repris et développé durant les séances de travaux dirigés.

**Remarque 1.** Les fonctions  $J_i$  considérées dans (4) résultent elles-même d'un problème de minimisation. C'est là quelque chose d'assez courant dans l'optimisation des grands systèmes. Ainsi, lorsque l'on s'intéresse à trouver les investissements optimaux à effectuer sur un système, il faut déterminer au préalable comment le faire fonctionner au mieux avec un investissement donné afin de pouvoir comparer différents niveaux d'investissement.

### 1.3 Rappels d'optimisation convexe et de dualité

On s'intéresse aux deux problèmes suivants :

— optimisation sans contrainte explicite

$$\min_{u \in U^{\text{ad}}} J(u) ; \quad (6)$$

— optimisation avec contraintes explicites

$$\min_{u \in U^{\text{ad}}} J(u) \quad \text{sous} \quad \Theta(u) \in -C . \quad (7)$$

#### 1.3.1 Notations et propriétés élémentaires

— **Espaces et ensembles.**

—  $\mathcal{U}$  et  $\mathcal{C}$  sont des espaces de *Hilbert* (espace vectoriel normé complet dont la norme dérive d'un produit scalaire). Exemples :

— en dimension finie :  $\mathbb{R}^n$  muni du produit scalaire  $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$  ;

— en dimension infinie :  $L^2([0, T], \mathbb{R})$  le produit scalaire étant  $\langle x, y \rangle = \int_0^T x(t)y(t)dt$ .

—  $U^{\text{ad}} \subset \mathcal{U}$  est un ensemble *convexe* :

$$\forall (u_1, u_2) \in U^{\text{ad}} \times U^{\text{ad}}, \forall \alpha \in [0, 1], \alpha u_1 + (1 - \alpha)u_2 \in U^{\text{ad}},$$

que l'on suppose *fermé* et non vide.

—  $C \subset \mathcal{C}$  est un *cône* :

$$\forall \alpha > 0, v \in C \Rightarrow \alpha v \in C,$$

que l'on suppose *convexe*, *fermé* et vérifiant :  $C \cap (-C) = \{0\}$ .

Le *cône dual*  $C^*$  de  $C$  est défini par :

$$C^* = \{p \in \mathcal{C}, \langle p, v \rangle \geq 0, \forall v \in C\} .$$

Exemples dans le cas  $\mathcal{C} = \mathbb{R}^m$  :

— contraintes égalité :  $C = \{0\} \rightsquigarrow C^* = \mathbb{R}^m$  ;

— contraintes inégalité :  $C = \mathbb{R}_+^m \rightsquigarrow C^* = \mathbb{R}_+^m$ .

— **Critère.**

Soit  $J : \mathcal{U} \rightarrow \overline{\mathbb{R}}$ . On appelle domaine de  $J$  et l'on note  $\text{dom}J$  l'ensemble :

$$\text{dom}J = \{u \in \mathcal{U}, J(u) < +\infty\} .$$

On suppose que  $J$  est une fonction *propre* :

$$\text{dom}J \neq \emptyset \quad \text{et} \quad J \text{ ne prend jamais la valeur } -\infty ,$$

et que minimiser  $J$  sur l'ensemble  $U^{\text{ad}}$  a un sens :

$$\text{dom}J \cap U^{\text{ad}} \neq \emptyset .$$

On fait alors sur la fonction  $J$  des hypothèses de *convexité*, de *continuité*, de *différentiabilité* et de *comportement à l'infini*. Plus précisément, les hypothèses que l'on pourra effectuer sont à prendre parmi les suivantes :

**condition de convexité :**

$$J \text{ convexe} \quad : J(\alpha u_1 + (1 - \alpha)u_2) \leq \alpha J(u_1) + (1 - \alpha)J(u_2) \quad \forall \alpha \in [0, 1],$$

$$J \text{ strictement convexe} : J(\alpha u_1 + (1 - \alpha)u_2) < \alpha J(u_1) + (1 - \alpha)J(u_2) \quad \forall \alpha \in ]0, 1[,$$

$$J \text{ fortement convexe} : J(\alpha u_1 + (1 - \alpha)u_2) \leq \alpha J(u_1) + (1 - \alpha)J(u_2) - \frac{\alpha}{2}(1 - \alpha)\|u_1 - u_2\|^2,$$

**condition de continuité :**

$$J \text{ semi-continue inférieurement (s.c.i.)} : \liminf_{u \rightarrow u_0} J(u) \geq J(u_0),$$

$$J \text{ continue} : \lim_{u \rightarrow u_0} J(u) = J(u_0),$$

$$J \text{ lipschitzienne} : |J(u_1) - J(u_2)| \leq L\|u_1 - u_2\|,$$

**condition de différentiabilité :<sup>1</sup>**

$$J \text{ sous-différentiable} : \partial J(u_0) = \{r \in \mathcal{U}, \forall u \in \mathcal{U}, J(u) - J(u_0) \geq \langle r, u - u_0 \rangle\} \neq \emptyset,$$

$$J \text{ Gâteaux-différentiable} : \exists \nabla J(u_0) \in \mathcal{U}, \forall d \in \mathcal{U}, \lim_{\epsilon \rightarrow 0^+} \frac{J(u_0 + \epsilon d) - J(u_0)}{\epsilon} = \langle \nabla J(u_0), d \rangle,$$

$$J \text{ Fréchet-différentiable} : \exists \nabla J(u_0) \in \mathcal{U}, \lim_{u \rightarrow u_0} \frac{J(u) - J(u_0) - \langle \nabla J(u_0), u - u_0 \rangle}{\|u - u_0\|} = 0,$$

**condition de comportement à l'infini :**

$$J \text{ coercive sur } U^{\text{ad}} : \lim_{\|u\| \rightarrow +\infty, u \in U^{\text{ad}}} J(u) = +\infty.$$

**— Contraintes**

Les contraintes sont modélisées par une application  $\Theta : \mathcal{U} \rightarrow \mathcal{C}$ , vérifiant des hypothèses de même nature que  $J$ , mais adaptées au fait que  $\Theta$  prend ses valeurs dans le cône  $C \subset \mathcal{C}$  :

**convexité :**

$$\Theta \text{ C-convexe} : \Theta(\alpha u_1 + (1 - \alpha)u_2) - \alpha \Theta(u_1) - (1 - \alpha)\Theta(u_2) \in -C,$$

**continuité :**

$$\Theta \text{ continue} : \lim_{u \rightarrow u_0} \Theta(u) = \Theta(u_0),$$

$$\Theta \text{ lipschitzienne} : \|\Theta(u_1) - \Theta(u_2)\| \leq \tau\|u_1 - u_2\|,$$

**différentiabilité :<sup>2</sup>**

$$\Theta \text{ C-sous-différentiable} : \partial \Theta(u_0) = \{\theta \in \mathcal{L}(\mathcal{U}, \mathcal{C}), \forall u, \Theta(u) - \Theta(u_0) - \theta.(u - u_0) \in C\},$$

$$\Theta \text{ Gâteaux-différentiable} : \exists \Theta'(u_0) \in \mathcal{L}(\mathcal{U}, \mathcal{C}), \forall d \in \mathcal{U}, \lim_{\epsilon \rightarrow 0^+} \frac{\Theta(u_0 + \epsilon d) - \Theta(u_0)}{\epsilon} = \Theta'(u_0).d.$$

1. On note  $\nabla J(u_0)$  le *gradient* de la fonction  $J$  au point  $u_0$ , la notation  $J'(u_0)$  correspondant à la différentielle.

2. Ici,  $\Theta'(u_0)$  correspond à la différentielle de la fonction  $\Theta$  au point  $u_0$ , représentée par sa *matrice jacobienne*.



**Remarque 2.** L'hypothèse de  $C$ -convexité (resp.  $C$ -sous-différentiabilité) est équivalente à supposer que la fonction  $u \mapsto \langle p, \Theta(u) \rangle$  est convexe (resp. sous-différentiable) pour tout  $p \in C^*$ . On pourrait de même formuler une hypothèse de  $C$ -semi-continuité inférieure sur  $\Theta$ , équivalente à supposer que la fonction  $u \mapsto \langle p, \Theta(u) \rangle$  est s.c.i pour tout  $p \in C^*$ .

### 1.3.2 Optimisation sans contrainte explicite

On cherche à donner une réponse aux problèmes de l'existence, de l'unicité, de la caractérisation et du calcul de la solution du problème (6).

- **Existence.** En dimension finie, supposer d'une part que la fonction  $J$  est s.c.i et d'autre part que l'ensemble  $U^{\text{ad}}$  est fermé et borné (ou à défaut de  $U^{\text{ad}}$  borné, que  $J$  est coercive sur  $U^{\text{ad}}$ ) est suffisant pour garantir l'existence de solution du problème.<sup>3</sup>
- **Unicité.** Si on suppose en plus des hypothèses précédentes que la fonction  $J$  est strictement convexe et que l'ensemble  $U^{\text{ad}}$  est convexe, on obtient l'unicité de la solution du problème.
- **Caractérisation.** Si l'on suppose de plus que  $J$  est Gâteaux-différentiable, une condition nécessaire et suffisante pour que  $u^\#$  soit solution du problème est que l'inégalité variationnelle suivante soit vérifiée :

$$\forall u \in U^{\text{ad}} , \quad \langle \nabla J(u^\#), u - u^\# \rangle \geq 0 .$$

*Cas particulier important.* Lorsque le problème n'est soumis à aucune contrainte ( $U^{\text{ad}} = \mathcal{U}$ ), cette condition prend la forme bien connue :

$$\nabla J(u^\#) = 0 .$$

- **Calcul.** Si l'on ajoute encore que  $J$  est fortement convexe (de module  $a$ ) et que sa différentielle est lipschitzienne (de constante  $A$ ), l'algorithme classique de gradient projeté :

$$u^{(k+1)} = \text{proj}_{U^{\text{ad}}} (u^{(k)} - \rho \nabla J(u^{(k)})) ,$$

converge vers l'unique solution  $u^\#$  du problème pourvu que le pas  $\rho$  de l'algorithme vérifie la condition :

$$\rho \in \left] 0, \frac{2a}{A^2} \right[ .$$

Il existe bien sûr d'autres algorithmes plus performants (gradient conjugué, quasi-Newton, ...).

### 1.3.3 Optimisation avec contraintes explicites et dualité

Dans le cas du problème (7), on peut se ramener au cas précédent en posant :

$$U^\Theta = \{u \in \mathcal{U}, \Theta(u) \in -C\} ,$$

et en effectuant la minimisation sur l'ensemble  $U^{\text{ad}} \cap U^\Theta$  (supposant que le cône  $C$  est un convexe fermé, l'ensemble  $U^\Theta$  est fermé (resp. convexe) dès que  $\Theta$  est continue (resp.  $C$ -convexe)).

Cependant, l'utilisation explicite de l'expression analytique des contraintes permet de donner une caractérisation plus opératoire des solutions du problème et de disposer d'algorithmes de calcul plus efficaces. En effet, effectuer une opération de projection est équivalent à résoudre un problème d'optimisation, ce qui peut s'avérer être une tâche difficile si l'ensemble sur lequel on projette est un convexe fermé quelconque (cas de la projection sur une intersection d'ensembles convexes...).

---

3. En dimension infinie, ces conditions ne sont pas suffisantes, et il faut de plus faire des hypothèses de convexité...

- **Qualification des contraintes et caractérisation.** Outre les hypothèses “classiques” de convexité, continuité et différentiabilité de  $J$  et  $\Theta$ , on introduit une condition de *qualification des contraintes* permettant de mieux caractériser une solution  $u^\sharp$  du problème (7). La condition que l’on utilisera est :

$$0 \in \text{int} (\Theta(U^{\text{ad}} \cap \text{dom}J) + C) ,$$

où  $\text{int}$  désigne l’intérieur topologique d’un ensemble.

À partir de cette condition générale, on retrouve les conditions plus classiques des cas de contraintes de type égalité et inégalité (où l’on suppose que  $\text{dom}J = \mathcal{U}$ ) :

**Cas**  $C = \{0\}$  :  $0 \in \text{int} (\Theta(U^{\text{ad}}))$ .

**Cas**  $C \neq \{0\}$  :  $\exists u_0 \in U^{\text{ad}}, \Theta(u_0) \in \text{int}(-C)$ .

Sous cette hypothèse de qualification, une solution  $u^\sharp$  de (7) est caractérisée par l’existence d’un élément  $p^\sharp$  appelé *multiplieur* et vérifiant les conditions de *Karush–Kuhn–Tucker* (**KKT**) :

$$\langle \nabla J(u^\sharp) + (\Theta'(u^\sharp))^\top \cdot p^\sharp, u - u^\sharp \rangle \geq 0 \quad \forall u \in U^{\text{ad}} , \quad (8a)$$

$$\Theta(u^\sharp) \in -C \quad , \quad p^\sharp \in C^* , \quad (8b)$$

$$\langle p^\sharp, \Theta(u^\sharp) \rangle = 0 . \quad (8c)$$

**Remarque 3.** La condition des écarts complémentaires (8c) est une condition de type *combinatoire* dans le cas de contrainte inégalité ; avec  $\mathcal{C} = \mathbb{R}^m$  et  $C = \mathbb{R}_+^m$ , elle prend la forme :

$$\sum_{j=1}^m p_j^\sharp \Theta_j(u^\sharp) = 0 ,$$

avec  $\Theta_j(u^\sharp) \leq 0$  et  $p_j^\sharp \geq 0$ . Chaque terme  $p_j^\sharp \Theta_j(u^\sharp)$  de la somme ci-dessus doit donc être nul, et la condition des écarts complémentaires est alors équivalente aux relations suivantes :

$$p_j^\sharp = 0 \quad \text{ou} \quad \Theta_j(u^\sharp) = 0 \quad \forall j \in \{1, \dots, m\} .$$

*Cas particuliers importants.* Utilisant le Lagrangien  $L(u, p)$  (défini page suivante), on remarque que les gradients partiels par rapport à  $u$  et  $p$  de ce Lagrangien ont pour expressions respectives :

$$\nabla_u L(u, p) = \nabla J(u) + (\Theta'(u))^\top \cdot p \quad , \quad \nabla_p L(u, p) = \Theta(u) .$$

Dans le cas où l’on a  $U^{\text{ad}} = \mathcal{U}$  (pas d’autres contraintes que celles définies par  $\Theta$ ), on montre que les conditions de Karush-Kuhn-Tucker prennent les formes particulières suivantes :

— pour les contraintes d’égalité  $\Theta(u) = 0$  :

$$\begin{aligned} \nabla_u L(u^\sharp, p^\sharp) &= 0 , \\ \nabla_p L(u^\sharp, p^\sharp) &= 0 ; \end{aligned}$$

— pour les contraintes d’inégalité  $\Theta(u) \leq 0$  :

$$\begin{aligned} \nabla_u L(u^\sharp, p^\sharp) &= 0 , \\ \Theta(u^\sharp) &\leq 0 \quad , \quad p^\sharp \geq 0 \quad , \quad \langle p^\sharp, \Theta(u^\sharp) \rangle = 0 . \end{aligned}$$

Le cas des contraintes d’égalité est bien sûr très favorable car la résolution des conditions de Karush-Kuhn-Tucker se ramène alors à la résolution d’un ensemble d’équations. . .

- **Interprétation marginaliste du multiplicateur.** On introduit la *fonction de perturbation*  $\Phi : \mathcal{C} \rightarrow \mathbb{R}$  du problème (7) comme la fonction correspondant à la minimisation du problème sous contraintes perturbées par  $v \in \mathcal{C}$  :

$$\Phi(v) = \min_{u \in U^{\text{ad}}} J(u) \quad \text{sous} \quad \Theta(u) - v \in -C. \quad (9)$$

Sous les hypothèses précédentes, la fonction  $\Phi$  est sous-différentiable au point  $v = 0$  et le multiplicateur  $p^\sharp$  des conditions de Karush–Kuhn–Tucker vérifie :

$$-p^\sharp \in \partial\Phi(0). \quad (10)$$

Dans le cas différentiable,  $p^\sharp$  s'interprète donc, au signe près, comme la *sensibilité du coût optimal* par rapport au niveau de contraintes requis. Le lien entre la fonction de perturbation et la solution du problème est donné par les deux relations :

$$\Phi(0) = J(u^\sharp) \quad \text{et} \quad \nabla\Phi(0) = -p^\sharp.$$

La fonction de perturbation  $\Phi$  est à la base de la théorie de la dualité en optimisation convexe. Elle permet de plus une interprétation géométrique simple de cette théorie (voir les exercices).

- **Lagrangien et point selle.** On appelle *Lagrangien* associé au problème (7) la fonction  $L$ , définie sur  $U^{\text{ad}} \times C^\star$  à valeurs dans  $\mathbb{R}$ , dont l'expression est :

$$L(u, p) = J(u) + \langle p, \Theta(u) \rangle, \quad (11)$$

On appelle *saut de dualité* la quantité  $\delta$  (toujours positive ou nulle) :

$$\delta = \min_{u \in U^{\text{ad}}} \max_{p \in C^\star} L(u, p) - \max_{p \in C^\star} \min_{u \in U^{\text{ad}}} L(u, p).$$

Le couple  $(u^\sharp, p^\sharp) \in U^{\text{ad}} \times C^\star$  est appelé *point selle* de  $L$  s'il vérifie la double inégalité :

$$\forall (u, p) \in U^{\text{ad}} \times C^\star, \quad L(u^\sharp, p) \leq L(u^\sharp, p^\sharp) \leq L(u, p^\sharp). \quad (12)$$

On montre que si  $(u_1^\sharp, p_1^\sharp)$  et  $(u_2^\sharp, p_2^\sharp)$  sont des points selle du Lagrangien,  $(u_1^\sharp, p_2^\sharp)$  et  $(u_2^\sharp, p_1^\sharp)$  le sont aussi. L'ensemble  $S^\sharp$  des points selle d'un Lagrangien se met donc sous la forme d'un *produit cartésien* d'ensembles :

$$S^\sharp = U^\sharp \times P^\sharp.$$

Lorsqu'un Lagrangien admet un point selle  $(u^\sharp, p^\sharp)$ , le saut de dualité est nul et on a l'égalité :

$$\max_{p \in C^\star} \min_{u \in U^{\text{ad}}} L(u, p) = \min_{u \in U^{\text{ad}}} \max_{p \in C^\star} L(u, p) = L(u^\sharp, p^\sharp). \quad (13)$$

On a alors les résultats suivants, caractérisant l'optimalité du point selle :

- (1) Si  $(u^\sharp, p^\sharp)$  est un point selle de  $L$  sur  $U^{\text{ad}} \times C^\star$ , alors  $u^\sharp$  est solution du problème (7).
- (2) Si  $J$  est convexe s.c.i propre coercive sur  $U^{\text{ad}}$ , si  $\Theta$  est  $C$ -convexe et continue et si l'hypothèse de qualification des contraintes est vérifiée, alors le Lagrangien  $L$  admet au moins un point selle sur  $U^{\text{ad}} \times C^\star$ .

Enfin, si l'on suppose de plus que les fonctions  $J$  et  $\Theta$  sont différentiables, tout point selle vérifie les conditions de Karush–Kuhn–Tucker.

- **Fonction duale et stabilité.** On définit la *fonction duale*  $H : C^\star \rightarrow \mathbb{R}$  par la relation :

$$H(p) = \min_{u \in U^{\text{ad}}} L(u, p), \quad (14)$$

et l'on note  $\widehat{U}(p)$  l'ensemble (indexé par  $p$ ) des valeurs de  $u$  réalisant le minimum dans l'expression (14). La maxi-minimisation du Lagrangien est alors équivalente à la maximisation de cette fonction duale :

$$\max_{p \in C^*} H(p) , \quad (15)$$

ce qui permet d'obtenir la partie  $p^\sharp$  du point selle. Ceci est en théorie raisonnable car la fonction  $H$  est concave, et un algorithme de gradient projeté peut permettre en pratique d'effectuer cette maximisation car les contraintes  $p \in C^*$  du problème (15) sont suffisamment simples pour que l'opération de projection soit facile à réaliser.

Cependant, même pour un  $p^\sharp \in P^\sharp$ , un élément  $u^\sharp$  provenant de la minimisation de  $L(u, p^\sharp)$  ne correspond pas forcément à la première composante d'un point selle du Lagrangien : en effet, si l'on a toujours :

$$U^\sharp \subset \widehat{U}(p^\sharp) , \quad (16)$$

l'inclusion ci-dessus peut être *stricte*, ce qui signifie que l'ensemble  $\widehat{U}(p^\sharp)$  contient des solutions "parasites" n'ayant rien à voir avec les solutions du problème (7). Pour pallier cette difficulté, on introduit la notion de *stabilité* du Lagrangien d'un problème d'optimisation, exprimant le fait que l'inclusion (16) est une *égalité*. Il n'y a alors pas de problème de fond pour le calcul de la première composante du point selle du Lagrangien.

À titre d'exemple, on considère le cas où l'on ajoute l'hypothèse que la fonction  $J$  est *strictement* convexe. Alors, le Lagrangien est strictement convexe en  $u$  et la minimisation de  $L(u, p^\sharp)$  conduit à une solution *unique*. L'ensemble  $\widehat{U}(p^\sharp)$  se réduit donc à un singleton, et on en déduit la stabilité du Lagrangien (car on sait que l'ensemble  $U^\sharp$  est non vide). Dans ce cas, on peut de plus montrer que la fonction duale  $H$  est *différentiable*. Notant  $\hat{u}(p)$  l'unique solution venant de la minimisation en  $u$  du Lagrangien  $L(u, p)$ , l'expression du gradient de  $H$  en  $p$  est :

$$\nabla H(p) = \nabla_p L(\hat{u}(p), p) = \Theta(\hat{u}(p)) .$$

La résolution du problème de minimisation (14) fournit donc la valeur de la fonction  $H$  en  $p$  ainsi que le gradient de  $H$  en ce point.

- **Calcul.** La différentiabilité de  $H$  ouvre la voie aux méthodes de gradient pour le calcul de la composante  $p^\sharp$  du point selle. Supposant que  $J$  est fortement convexe (de module  $a$ ) et que  $\Theta$  est lipschitzienne (de constante  $\tau$ ), l'algorithme d'*Uzawa* consiste à mettre en œuvre un algorithme de gradient à pas fixe pour maximiser la fonction duale  $H$ , le calcul du gradient de cette fonction nécessitant la résolution d'un problème de type (14). Partant d'un point  $(u^{(k)}, p^{(k)}) \in U^{\text{ad}} \times C^*$ , l'itération  $k$  de l'algorithme d'*Uzawa* calcule le point suivant  $(u^{(k+1)}, p^{(k+1)})$  par les relations :

$$\begin{aligned} u^{(k+1)} &= \arg \min_{u \in U^{\text{ad}}} (L(u, p^{(k)})) , \\ p^{(k+1)} &= \text{proj}_{C^*} (p^{(k)} + \rho \nabla_p L(u^{(k+1)}, p^{(k)})) . \end{aligned}$$

qui se mettent aussi sous la forme :

$$\begin{aligned} u^{(k+1)} &= \arg \min_{u \in U^{\text{ad}}} (J(u) + \langle p^{(k)}, \Theta(u) \rangle) , \\ p^{(k+1)} &= \text{proj}_{C^*} (p^{(k)} + \rho \Theta(u^{(k+1)})) . \end{aligned}$$

Cet algorithme converge vers l'unique solution  $u^\sharp$  du problème (7) avec le choix  $\rho \in ]0, \frac{2a}{\tau^2}[$ . Une variante de l'algorithme d'*Uzawa* consiste à faire un pas de gradient sur la variable  $u$  plutôt que la minimisation complète. On obtient alors l'algorithme d'*Arrow-Hurwicz* :

$$\begin{aligned} u^{(k+1)} &= \text{proj}_{U^{\text{ad}}} (u^{(k)} - \epsilon \nabla_u L(u^{(k)}, p^{(k)})) , \\ p^{(k+1)} &= \text{proj}_{C^*} (p^{(k)} + \rho \nabla_p L(u^{(k+1)}, p^{(k)})) . \end{aligned}$$

soit encore :

$$\begin{aligned} u^{(k+1)} &= \text{proj}_{U^{\text{ad}}} \left( u^{(k)} - \epsilon \left( \nabla J(u^{(k)}) + (\Theta'(u^{(k)}))^{\top} p^{(k)} \right) \right), \\ p^{(k+1)} &= \text{proj}_{C^*} \left( p^{(k)} + \rho \Theta(u^{(k+1)}) \right). \end{aligned}$$

### 1.3.4 Complément : sous-différentiel des fonctions marginales

On donne dans cette section les résultats concernant la (sous)-différentiabilité de fonctions résultant d'une opération de minimisation. Ces résultats ont été utilisés dans les paragraphes précédents "Interprétation marginaliste du multiplicateur" (pour la fonction de perturbation  $\Phi$ ) et "Fonction duale et stabilité" (pour la fonction duale  $H$ ).

Ces résultats seront détaillés durant le cours SOD314 dans le cadre de l'étude de la sous-différentiabilité des fonctions convexes.

**1.3.4.1 Cas d'une fonction convexe-convexe** Soit  $G$  une fonction définie sur le produit d'espace  $\mathcal{U} \times \mathcal{C}$ , à valeurs dans  $] -\infty, +\infty]$ , et soit  $U^{\text{ad}}$  un sous-ensemble de l'espace  $\mathcal{U}$ . On fait les hypothèses suivantes.

#### Hypothèse 1.

1. Le sous-ensemble  $U^{\text{ad}}$  est convexe fermé.
2. La fonction  $G$  est conjointement convexe en  $(u, v)$ , s.c.i. et sous-différentiable.
3. L'application  $u \mapsto G(u, v)$  est coercive sur  $U^{\text{ad}}$ .
4. Il existe  $u_0 \in U^{\text{ad}}$  et  $v_0 \in \mathcal{C}$ , tels que l'application  $v \mapsto G(u_0, v)$  est continue en  $v_0$ .

On définit alors la fonction marginale  $\Phi$  :<sup>4</sup>

$$\Phi(v) = \min_{u \in U^{\text{ad}}} G(u, v),$$

ainsi que l'ensemble des minimiseurs associé :

$$\widehat{U}(v) = \arg \min_{u \in U^{\text{ad}}} G(u, v).$$

L'hypothèse 1-2 permet d'assurer que la fonction  $\Phi$  est convexe, et l'hypothèse 1-4 implique que  $\Phi$  est bornée supérieurement au voisinage du point  $v_0$  et donc continue et sous-différentiable sur l'intérieur de son domaine.

**Théorème 1.** *Soit  $G$  une application vérifiant les hypothèses 1 et soit  $v$  un point de  $\mathcal{C}$  en lequel la fonction  $\Phi$  est sous-différentiable. Alors :*

$$\partial\Phi(v) \subset \bigcap_{\widehat{u} \in \widehat{U}(v)} \partial_v G(\widehat{u}, v). \quad (17)$$

**Cas différentiable.** Si la fonction  $v \mapsto G(u, v)$  est différentiable, alors la fonction  $\Phi$  est elle aussi différentiable et l'on a :

$$\nabla\Phi(v) = \nabla_v G(\widehat{u}, v) \quad \forall \widehat{u} \in \widehat{U}(v).$$

**Application à la fonction de perturbation.** Prenant  $G(u, v) = J(u) + \chi_{\{\Theta(u) - v \in -C\}}(u, v)$ , la fonction marginale  $\Phi$  qui s'en déduit est la *fonction de perturbation* du problème (7), et le théorème 1 permet alors de démontrer le résultat concernant l'interprétation marginaliste du multiplicateur.

4. Les hypothèses 1-1, 1-2 et 1-3 permettent d'affirmer que le min définissant la fonction  $\Phi$  existe.

**1.3.4.2 Cas d'une fonction convexe-concave** Soit  $L$  une fonction définie sur le produit des espaces  $\mathcal{U} \times \mathcal{C}$ , à valeurs dans  $] -\infty, +\infty]$ , et soit  $U^{\text{ad}}$  un sous-ensemble de l'espace  $\mathcal{U}$ . On fait les hypothèses suivantes.

**Hypothèse 2.**

1. Le sous-ensemble  $U^{\text{ad}}$  est convexe fermé.
2. L'application  $u \mapsto L(u, p)$  est propre, convexe, s.c.i., coercive sur  $U^{\text{ad}}$ .
3. L'application  $p \mapsto L(u, p)$  est concave, continue, sur-différentiable.

On définit alors la fonction marginale :

$$H(p) = \min_{u \in U^{\text{ad}}} L(u, p) ,$$

ainsi que l'ensemble des minimiseurs associé :

$$\widehat{U}(p) = \arg \min_{u \in U^{\text{ad}}} L(u, p) .$$

L'hypothèse 2-3 permet d'assurer que la fonction  $H$  est concave, s.c.s., sur-différentiable sur l'intérieur de son domaine.

**Théorème 2.** Soit  $L$  une application vérifiant l'hypothèse 2 et soit  $p$  un point de  $\mathcal{C}$  en lequel la fonction  $H$  est sur-différentiable. Alors :

$$\partial H(p) = \text{cl} \left( \text{conv} \left( \bigcup_{\widehat{u} \in \widehat{U}(p)} \partial_p L(\widehat{u}, p) \right) \right) , \tag{18}$$

où  $\text{cl}(A)$  (resp.  $\text{conv}(A)$ ) désigne la fermeture (resp. l'enveloppe convexe) d'un l'ensemble  $A$ .

**Cas différentiable.** Si la fonction  $p \mapsto L(u, p)$  est différentiable et si l'argmin  $\widehat{U}(p)$  est réduit à un point unique  $\widehat{u}(p)$ , alors la fonction  $H$  est différentiable et l'on a :

$$\nabla H(p) = \nabla_p L(\widehat{u}(p), p) .$$

**Application à la fonction duale.** Prenant  $L(u, v) = J(u) + \langle p, \Theta(u) \rangle$ , la fonction marginale  $H$  qui s'en déduit est la *fonction duale* du problème (7), et le théorème 2 appliqué dans le cas différentiable permet d'affirmer que, pour  $p \in C^*$ , on a :  $\nabla H(p) = \Theta(\widehat{u}(p))$ .

## 1.4 Quelques références bibliographiques

- G. COHEN : *Convexité et optimisation*. Cours ENPC.  
<https://cel.archives-ouvertes.fr/cel-00356686>
- J.-C. CULIOLI : *Introduction à l'optimisation*. Éditions Ellipses, 1994.
- I. EKELAND : *Analyse convexe et problèmes variationnels*. Éditions Dunod, 1974.
- R. TEMAM
- J.-C. GILBERT : *Optimisation différentiable : théorie et algorithmes*. Cours ENSTA.

- Le cours de J.-C. Gilbert a été normalement étudié en deuxième année. C’est une référence très complète, facilement accessible aux étudiants de l’ENSTA.
- Le cours de G. Cohen (qui a aussi rédigé le polycopié correspondant au cours “Optimisation des grands systèmes”) est plus centré sur la convexité. Il est disponible sur le site Web de l’auteur.
- Le livre de J.-C. Culioli offre un panorama complet et accessible de l’optimisation continue. Des programmes (écrits en MATHEMATICA) illustrent l’ensemble de l’ouvrage.
- On trouvera dans le livre d’I. Ekeland et R. Temam une présentation de la théorie de la dualité par les perturbations dans les espaces de Hilbert de dimension infinie.
- Il existe enfin **beaucoup** d’autres ouvrages d’excellente qualité traitant de l’optimisation, dont voici une sélection largement non exhaustive. . .
  - Convex Analysis, R. T. Rockafellar, 1970, Princeton University Press.
  - Convex Analysis and Minimization Algorithms, J.-B. Hiriart-Urruty et C. Lemarechal, 1993, Springer.
  - Nonlinear Programming, D. Bertsekas, 1999, Athena Scientific.
  - Numerical Optimization, J. Nocedal and S. J. Wright, 1999, Springer.
  - Nonlinear Optimization, A. Ruszczyński, 2006, Princeton University Press.
  - Convex Analysis and Nonlinear Optimization, J. Borwein et A. S. Lewis, 2006, Springer.

## 1.5 Exercices

### 1.5.1 Conditions de Karush–Kuhn–Tucker et grands systèmes

On s'intéresse à la résolution du problème d'optimisation du réseau de distribution d'eau  $i$  défini au §1.2.2, dont on rappelle ci-dessous la forme :

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

sous les contraintes :

$$v_{i,1} \geq \bar{v}_{i,1} \quad \text{et} \quad v_{i,1} + v_{i,2} = \bar{v}_{i,2} .$$

- 1) Mettre ce problème sous forme standard, s'assurer de l'existence et de l'unicité de la solution et calculer cette solution en résolvant directement les conditions de Kuhn et Tucker.
- 2) Formuler le problème de minimisation que l'on obtient, avant toute opération de minimisation, lorsque l'on connecte  $N$  réseaux par une usine de refoulement commune; expliquer pourquoi la méthode de résolution de la première question est impraticable sur ce problème quand  $N$  est grand.

### 1.5.2 Interprétation géométrique de la dualité

On s'intéresse à un problème d'optimisation sous contraintes :

$$\min_{u \in U^{\text{ad}}} J(u) \quad \text{sous} \quad \Theta(u) = 0 ,$$

que l'on résout par l'algorithme d'Uzawa :

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

Posant  $v = \Theta(u)$ , la partie minimisation de ce problème se réécrit :

$$\min_{v \in \mathcal{C}} \min_{u \in U^{\text{ad}}} J(u) + \langle p^{(k)}, v \rangle \quad \text{sous} \quad \Theta(u) - v = 0 .$$

À l'aide de la fonction de perturbation  $\Phi$ , qui s'écrit alors :

$$\Phi(v) = \min_{u \in U^{\text{ad}}} J(u) \quad \text{sous} \quad \Theta(u) - v = 0 ,$$

l'algorithme se met sous la forme :

$$\begin{aligned} v^{(k+1)} &= \arg \min_{v \in \mathcal{C}} (\Phi(v) + \langle p^{(k)}, v \rangle) , \\ p^{(k+1)} &= p^{(k)} + \rho v^{(k+1)} . \end{aligned}$$

- 1) Interpréter géométriquement cet algorithme.
- 2) Traiter le cas des contraintes inégalités :  $\Theta(u) \leq 0$ .
- 3) Cas où la fonction  $\Phi$  n'est pas strictement convexe.



### 1.5.3 Stabilité du Lagrangien

Soit à résoudre le problème de minimisation suivant :

$$\min_{u \in [-1,1]} -u \quad \text{sous } u \leq 0 ,$$

dont la solution évidente est  $u^\# = 0$ .

- 1) Montrer que le Lagrangien associé à ce problème admet sur  $[-1, 1] \times \mathbb{R}_+$  un point selle  $(u^\#, p^\#)$ .
- 2) Calculer le minimum en  $u$  du Lagrangien au point  $p = p^\#$  (algorithme de Uzawa).
- 3) Interpréter géométriquement à l'aide de l'exercice précédent.

### 1.5.4 Problèmes équivalents

Montrer les équivalences suivantes (on supposera que les problèmes d'optimisation apparaissant ci-dessous ont tous au moins une solution).

- 1) Décomposition naturelle.

$$\min_{(u_1, u_2) \in U_1^{\text{ad}} \times U_2^{\text{ad}}} J_1(u_1) + J_2(u_2) \iff \min_{u_1 \in U_1^{\text{ad}}} J_1(u_1) \quad \text{et} \quad \min_{u_2 \in U_2^{\text{ad}}} J_2(u_2) .$$

- 2) Ajout de variables.

$$\min_{u \in U^{\text{ad}}} J(u, \Theta(u)) \iff \min_{(u, v) \in U^{\text{ad}} \times \mathcal{C}} J(u, v) \quad \text{sous } \Theta(u) - v = 0 .$$

- 3) Minimisation emboîtée.

$$\min_{(u, v) \in U^{\text{ad}} \times \mathcal{C}} J(u, v) \iff \min_{v \in \mathcal{C}} F(v) \quad \text{avec} \quad F(v) = \min_{u \in U^{\text{ad}}} J(u, v) .$$

## 1.6 Corrigés

### 1.6.1 Conditions de Karush–Kuhn–Tucker et grands systèmes

1) Le problème d'optimisation sous contraintes à résoudre s'écrit sous la forme standard :

$$\min_{(v_{i,1}, v_{i,2})} \frac{1}{2} (a_{i,1}v_{i,1}^2 + a_{i,2}v_{i,2}^2) , \quad (19a)$$

sous les contraintes

$$\bar{v}_{i,1} - v_{i,1} \leq 0 , \quad (19b)$$

$$v_{i,1} + v_{i,2} - \bar{v}_{i,2} = 0 , \quad (19c)$$

et vérifie toutes les propriétés requises :

- l'ensemble admissible  $U^{\text{ad}}$ , étant  $\mathbb{R}^2$  tout entier, est bien convexe et fermé ;
- les prix de l'énergie  $a_{i,1}$  et  $a_{i,2}$  étant strictement positifs et la fonction coût  $J$  étant quadratique,  $J$  est fortement convexe (et donc coercive), continue, différentiable de dérivée  $J'$  linéaire et donc lipschitzienne ;
- les contraintes  $\Theta$ , à valeurs dans  $\mathbb{R}^2$  étant linéaires, sont convexes sur le cône  $C$ , continues et différentiables ;
- le cône des contraintes  $C$  est égal à  $\mathbb{R}_+ \times \{0\}$ , le cône dual  $C^*$  est égal à  $\mathbb{R}_+ \times \mathbb{R}$ , et les hypothèses de qualification des contraintes sont vérifiées.

Le Lagrangien du problème s'écrit alors :

$$L(v_{i,1}, v_{i,2}, \lambda_{i,1}, \lambda_{i,2}) = \frac{1}{2} (a_{i,1}v_{i,1}^2 + a_{i,2}v_{i,2}^2) + \lambda_{i,1} (\bar{v}_{i,1} - v_{i,1}) + \lambda_{i,2} (v_{i,1} + v_{i,2} - \bar{v}_{i,2}) ,$$

et les conditions de Karush–Kuhn–Tucker, qui caractérisent la solution dans le cas convexe, sont :

$$a_{i,1}v_{i,1} - \lambda_{i,1} + \lambda_{i,2} = 0 , \quad (20a)$$

$$a_{i,2}v_{i,2} + \lambda_{i,2} = 0 , \quad (20b)$$

$$\bar{v}_{i,1} - v_{i,1} \leq 0 \quad , \quad \lambda_{i,1} \geq 0 , \quad (20c)$$

$$\lambda_{i,1} (\bar{v}_{i,1} - v_{i,1}) = 0 , \quad (20d)$$

$$v_{i,1} + v_{i,2} - \bar{v}_{i,2} = 0 \quad , \quad \lambda_{i,2} \text{ quelconque} . \quad (20e)$$

Pour résoudre ce système d'inéquations, on joue aux devinettes . . .

- Supposons que la contrainte (19b) ne soit pas saturée :  $\bar{v}_{i,1} - v_{i,1} < 0$ . Dans ce cas, la condition des écarts complémentaires (20d) implique  $\lambda_{i,1} = 0$ . Il suffit alors de résoudre le système linéaire (20a)–(20b)–(20e) en les variables  $v_{i,1}$ ,  $v_{i,2}$  et  $\lambda_{i,2}$ , dont la solution est :

$$v_{i,1} = \frac{a_{i,2}}{a_{i,1} + a_{i,2}} \bar{v}_{i,2} \quad , \quad v_{i,2} = \frac{a_{i,1}}{a_{i,1} + a_{i,2}} \bar{v}_{i,2} \quad , \quad \lambda_{i,2} = -\frac{a_{i,1}a_{i,2}}{a_{i,1} + a_{i,2}} \bar{v}_{i,2} . \quad (21)$$

Il faut encore vérifier a posteriori que la contrainte (19b) n'est pas saturée, ce qui conduit à la condition :

$$(a_{i,1} + a_{i,2})\bar{v}_{i,1} - a_{i,2}\bar{v}_{i,2} < 0 . \quad (22)$$

- Si au contraire la contrainte (19b) est saturée, on a  $v_{i,1} = \bar{v}_{i,1}$ , et on résout le système linéaire (20a)–(20b)–(20e) en les variables  $v_{i,2}$ ,  $\lambda_{i,1}$  et  $\lambda_{i,2}$ . La solution est alors :

$$v_{i,1} = \bar{v}_{i,1} \quad , \quad v_{i,2} = \bar{v}_{i,2} - \bar{v}_{i,1} \quad , \quad \lambda_{i,1} = (a_{i,1} + a_{i,2})\bar{v}_{i,1} - a_{i,2}\bar{v}_{i,2} \quad , \quad \lambda_{i,2} = -a_{i,2}(\bar{v}_{i,2} - \bar{v}_{i,1}) , \quad (23)$$

et on doit vérifier a posteriori que le multiplicateur  $\lambda_{i,1}$  est positif ou nul, ce qui se traduit par la condition :

$$(a_{i,1} + a_{i,2})\bar{v}_{i,1} - a_{i,2}\bar{v}_{i,2} \geq 0 . \quad (24)$$

On a ainsi la solution du problème dans tous les cas.

**Remarque 4.** La condition de cohérence  $\bar{v}_{i,2} \geq \bar{v}_{i,1} \geq 0$  sur les *données* du problème implique que la solution vérifie dans tous les cas les conditions  $v_{i,1} \geq 0$  et  $v_{i,2} \geq 0$ , qui signifient que l’usine locale alimente le réseau et non le contraire. . . Il aurait été plus réaliste d’ajouter les conditions  $v_{i,1} \geq 0$  et  $v_{i,2} \geq 0$  sur les *variables* du problème. Mais on aurait alors eu 3 contraintes de type inégalité à traiter, soit 8 alternatives dans le jeu de devinettes précédent, ce qui aurait notablement alourdi la résolution par Karush–Kuhn–Tucker.

2) Lorsque l’on connecte  $N$  réseaux par une usine commune, le problème d’optimisation auquel on parvient avant toute opération de minimisation se met, avec les notations du § 1.2.2 sous la forme :

$$\min_{\{(u_{i,1}, u_{i,2})\}_{i=1}^N} J_{N+1} \left( \sum_{i=1}^N u_{i,1}, \sum_{i=1}^N u_{i,2} \right) + \sum_{i=1}^N \frac{1}{2} (a_{i,1} v_{i,1}^2 + a_{i,2} v_{i,2}^2), \quad (25a)$$

sous les contraintes

$$\bar{v}_{i,1} - u_{i,1} - v_{i,1} \leq 0 \quad \forall i = 1, \dots, N, \quad (25b)$$

$$u_{i,1} + u_{i,2} + v_{i,1} + v_{i,2} - \bar{v}_{i,2} = 0 \quad \forall i = 1, \dots, N. \quad (25c)$$

Pour calculer par Kuhn et Tucker la solution de ce problème, on doit prendre en compte  $N$  contraintes de type inégalité. Le nombre d’alternatives à envisager dans le jeu des devinettes qui découle des conditions de Karush–Kuhn–Tucker est alors égal à  $2^N$ . Dans le cas d’un réseau à l’échelle régionale,  $N$  est de l’ordre de 20, et le nombre d’alternatives de l’ordre du million. La résolution directe consiste donc à former et résoudre un million de systèmes d’équations linéaires de dimension  $3N$ , ce qui devient très lourd.

On peut ajouter qu’en réintroduisant les contraintes naturelles de positivité sur les variables  $v_{i,1}$  et  $v_{i,2}$  dans chacun des  $N$  premiers sous-réseaux, on doit alors considérer 8 alternatives dans chacun des sous-réseaux (voir Remarque 4) et donc  $8^N$  alternatives en tout, soit, pour  $N = 20$ , un nombre ( $\approx 10^{18}$ ) défiant toute puissance de calcul actuelle.

## 1.6.2 Interprétation géométrique de la dualité

1) Chaque itération de l’algorithme d’Uzawa comporte 2 étapes.

- La première étape consiste à effectuer la minimisation en  $u$  du Lagrangien à  $p = p^{(k)}$  fixé. À l’aide de la fonction de perturbation  $\Phi$ , cette étape se met sous la forme :

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

Dans un plan dont l’axe des abscisses représente l’espace  $\mathcal{C}$  et l’axe des ordonnées représente  $\mathbb{R}$ , elle s’interprète comme trouver la valeur  $v^{(k+1)}$  qui minimise l’écart entre les deux fonctions  $v \mapsto \Phi(v)$  et  $v \mapsto -\langle p^{(k)}, v \rangle$ . Dans le cas où la fonction  $\Phi$  est différentiable, ceci revient à trouver la valeur  $\xi$  telle que la droite d’équation  $-\langle p^{(k)}, \cdot \rangle + \xi$  soit tangente à  $\Phi$ , l’abscisse du point de tangence étant alors  $v^{(k+1)}$ . D’un point de vue géométrique, on déplace dans le plan  $\mathcal{C} \times \mathbb{R}$  une droite de pente  $-p^{(k)}$  jusqu’à “tangenter” la fonction  $\Phi$ .

- La seconde étape consiste à mettre à jour le multiplicateur par la relation :

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

et donc :

- à diminuer la pente  $-p^{(k)}$  de la droite si l’abscisse  $v^{(k+1)}$  est positive,

- à augmenter la pente  $-p^{(k)}$  de la droite si l’abscisse  $v^{(k+1)}$  est négative,

sachant que le point visé est celui correspondant à l’abscisse nulle (on cherche à calculer  $\Phi(0)$ ).

Une séquence de l’algorithme ( $p^{(k)} \rightsquigarrow v^{(k+1)} \rightsquigarrow p^{(k+1)} \rightsquigarrow v^{(k+2)}$ ), ainsi que la droite optimale de pente  $-p^\sharp$ , sont représentées Figure 1.

On retrouve ainsi l’interprétation marginaliste du multiplicateur optimal  $p^\sharp$  comme étant, au signe près, la sensibilité du coût optimal au niveau de contraintes :

$$-p^\sharp = \frac{d\Phi}{dv}(0).$$

**Remarque** Cette interprétation géométrique reste conceptuelle, car on n’évalue jamais directement la fonction  $\Phi$ . Elle permet simplement de mieux comprendre le fonctionnement de l’algorithme de Uzawa.

2) On se limite ici au cas  $\mathcal{C} = \mathbb{R}^m$ , le cône des contraintes étant alors  $C = \mathbb{R}_+^m$  et la contrainte :

$$\Theta(u) \leq 0,$$

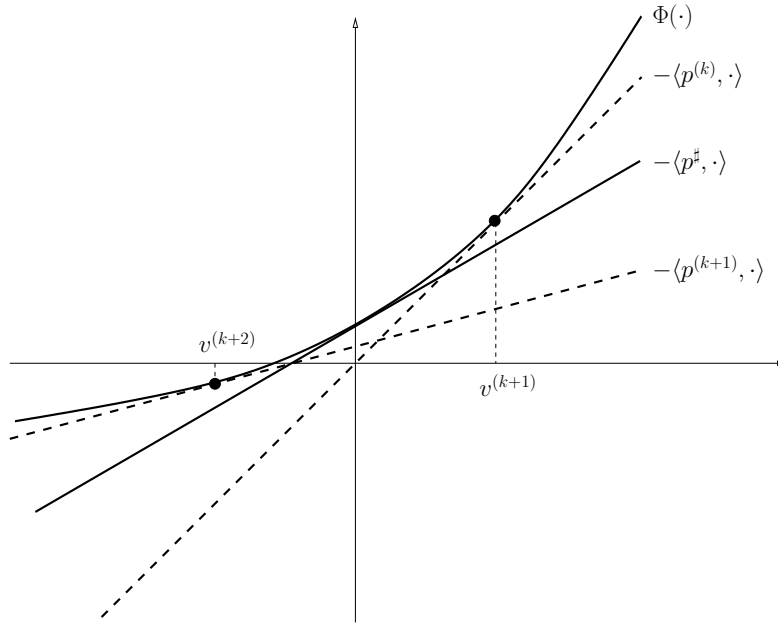


FIGURE 1 – Déroulement de l'algorithme de Uzawa

signifiant que chacune de ses composantes scalaires  $\Theta_j$  est négative ou nulle.

Une itération de l'algorithme de Uzawa consiste alors à enchaîner les deux étapes suivantes :

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

$$\forall j = 1, \dots, m, \quad p_j^{(k+1)} = \max \left\{ 0, p_j^{(k)} + \rho \Theta_j(u^{(k+1)}) \right\} .$$

Prenant alors pour fonction de perturbation :

$$\Phi(v) = \min_{u \in U_{\text{ad}}} J(u) \quad \text{sous} \quad \Theta(u) - v \leq 0 ,$$

on a les équivalences suivantes :

$$\begin{aligned} \min_{u \in U_{\text{ad}}} J(u) + \langle p^{(k)}, \Theta(u) \rangle &\iff \min_{u \in U_{\text{ad}}} \min_{w \leq 0} J(u) + \langle p^{(k)}, \Theta(u) - w \rangle \\ &\iff \min_{u \in U_{\text{ad}}} \min_{w \leq 0} \min_{v \in \mathcal{C}} \left\{ J(u) + \langle p^{(k)}, v \rangle \text{ sous } \Theta(u) - w = v \right\} \\ &\iff \min_{u \in U_{\text{ad}}} \min_{v \in \mathcal{C}} \left\{ J(u) + \langle p^{(k)}, v \rangle \text{ sous } \Theta(u) - v \leq 0 \right\} \\ &\iff \min_{v \in \mathcal{C}} \Phi(v) + \langle p^{(k)}, v \rangle . \end{aligned}$$

- La première équivalence vient du fait que,  $p^{(k)}$  étant positif ou nul, le minimum en  $w$  est atteint pour  $w = 0$ .
- Dans la seconde équivalence, on ajoute simplement des variables intermédiaires  $v$ .
- La troisième équivalence est vraie car avoir  $\Theta(u) - w = v$  avec  $w \leq 0$  est équivalent à  $\Theta(u) - v \leq 0$ .
- Enfin, la quatrième équivalence vient de la définition même de la fonction de perturbation  $\Phi$ .

On retrouve la même forme que dans le cas des contraintes de type égalité, de telle sorte que la première étape de l'algorithme s'interprète de nouveau comme trouver le point de tangence entre la fonction de perturbation  $\Phi$  et une droite de pente  $-p^{(k)}$ . La formule de remise à jour des multiplicateurs fait que l'on ne peut utiliser que des droites de pente *négative*, et l'on doit donc considérer deux cas :

- si la dérivée en zéro de la fonction de perturbation est positive (Figure 2, partie gauche), on ne peut pas atteindre la valeur  $v = 0$ ; on est alors dans le cas où les contraintes ne sont pas saturées à l'optimum :  $v^\# \leq 0$  et  $p^\# = 0$ , ce qui revient à dire que la minimisation s'effectue sans tenir compte des contraintes;
- si la dérivée en zéro de la fonction de perturbation est négative (Figure 2, partie droite), on peut atteindre la valeur  $v^\# = 0$  par des droites de pente négative; on est dans le cas des contraintes saturées, actives à l'optimum avec un multiplicateur  $p^\# \geq 0$ .

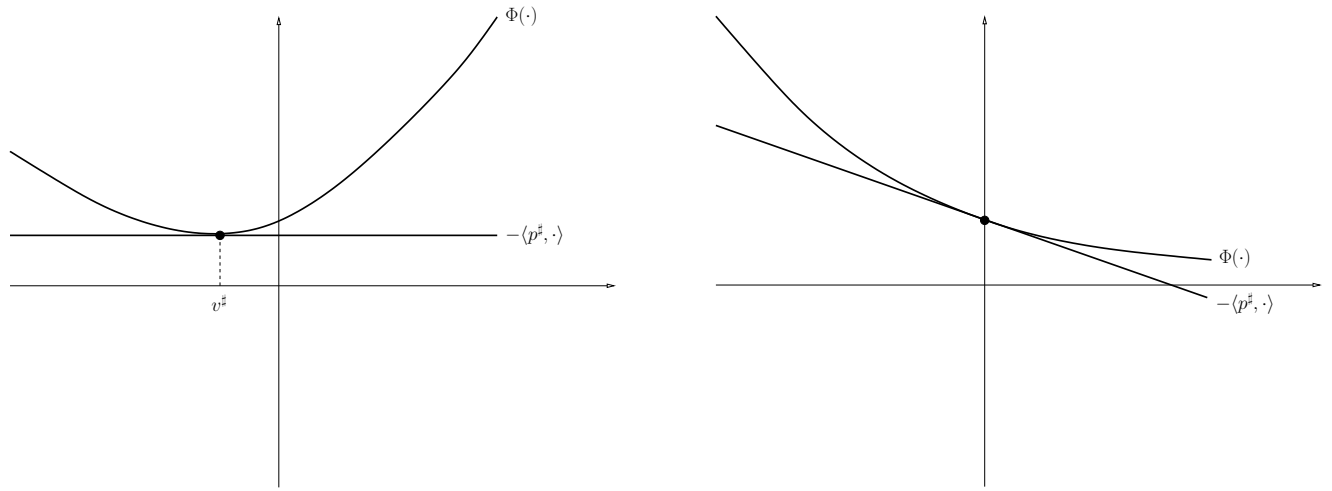


FIGURE 2 – Interprétation de l’algorithme de Uzawa dans le cas des contraintes inégalité

3) Dans l’interprétation géométrique développée ici, on ne s’intéresse qu’aux deux suites  $\{v^{(k)}\}_{k \in \mathbb{N}}$  et  $\{p^{(k)}\}_{k \in \mathbb{N}}$ , le but étant que la suite  $\{p^{(k)}\}_{k \in \mathbb{N}}$  converge vers une valeur  $p^\#$  telle que le  $v^\#$  correspondant soit égal à 0. En particulier, on n’utilise jamais la suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$ . Dans le cas où  $p^{(k)} \rightarrow p^\#$  (rappelons cependant que cette convergence n’est pas *garantie* dans l’algorithme de Uzawa), on peut calculer une valeur  $u^\#$  en résolvant le problème :

$$\min_{u \in U_{\text{ad}}} J(u) + \langle p^\#, \Theta(u) \rangle . \quad (26)$$

On sait que l’ensemble  $\hat{U}(p^\#)$  des solutions de ce problème *contient* l’ensemble  $U^\#$  des solutions du problème initial :

$$U^\# \subset \hat{U}(p^\#) . \quad (27)$$

Dans le cas où la fonction  $J$  est strictement convexe, le problème (26) a une unique solution  $u^\#$ , de telle sorte que l’inclusion (27) est en fait une égalité :  $U^\# = \hat{U}(p^\#) = \{u^\#\}$ .

Dans le cas non strictement convexe, l’inclusion (27) peut être stricte, l’ensemble  $\hat{U}(p^\#)$  contenant des solutions de (26) ne respectant pas les contraintes du problème initial. On illustre ceci sur le problème (élémentaire) suivant :

$$\min_{u \in \mathbb{R}} J(u) \quad \text{sous } u = 0 ,$$

dont la fonction de perturbation est  $\Phi = J$  et dont l’unique solution est  $u^\# = 0$ .

- On considère d’abord le cas d’une fonction  $J$  convexe, mais non strictement convexe (Figure 3, partie gauche). Alors,  $\hat{U}(p^\#) = [\alpha, \beta]$  ne contient pas uniquement la solution  $u^\# = 0$ .
- Dans le cas d’une fonction  $J$  non convexe (Figure 3, partie droite),  $\hat{U}(p^\#) = \{\alpha, \beta\}$  ne contient même pas la solution recherchée.

On notera que, du point de vue numérique, les deux cas sont proches. En effet, dans la mise en œuvre sur ordinateur, les valeurs  $p^{(k)}$  obtenues dans la zone de convergence de l’algorithme seront proches à  $\epsilon$  près de  $p^\#$  sans jamais lui être égales. Les valeurs de  $u^{(k)}$  correspondantes seront égales à  $\alpha$  ou  $\beta$  suivant le signe du  $\epsilon$ , et ne prendront donc jamais la valeur de la solution recherchée.

Une réponse intuitive aux difficultés posées par les cas non convexes est la suivante :

*il ne faut pas chercher des **droites** tangentes à la fonction de perturbation,  
mais plutôt des **paraboles** osculatrices à cette fonction de perturbation.*

Cette idée revient à minimiser l’écart entre la fonction  $\Phi$  et la parabole d’équation  $-\langle p^{(k)}, \cdot \rangle - \frac{c}{2} \|\cdot\|^2$ ,  $c$  étant un coefficient strictement positif choisi à l’avance. Elle est illustrée Figure 4 pour les cas d’une fonction de perturbation non strictement convexe d’une part, et non convexe d’autre part. On notera que, s’il suffit de choisir un coefficient  $c$  strictement positif dans le premier cas, la valeur de  $c$  doit être prise suffisamment grande dans le second cas pour pouvoir “s’ajuster” à la non-convexité de  $\Phi$ . La partie minimisation de l’algorithme d’Uzawa se met sous la forme :

$$\min_{v \in \mathcal{C}} \Phi(v) + \langle p^{(k)}, v \rangle + \frac{c}{2} \|v\|^2 .$$

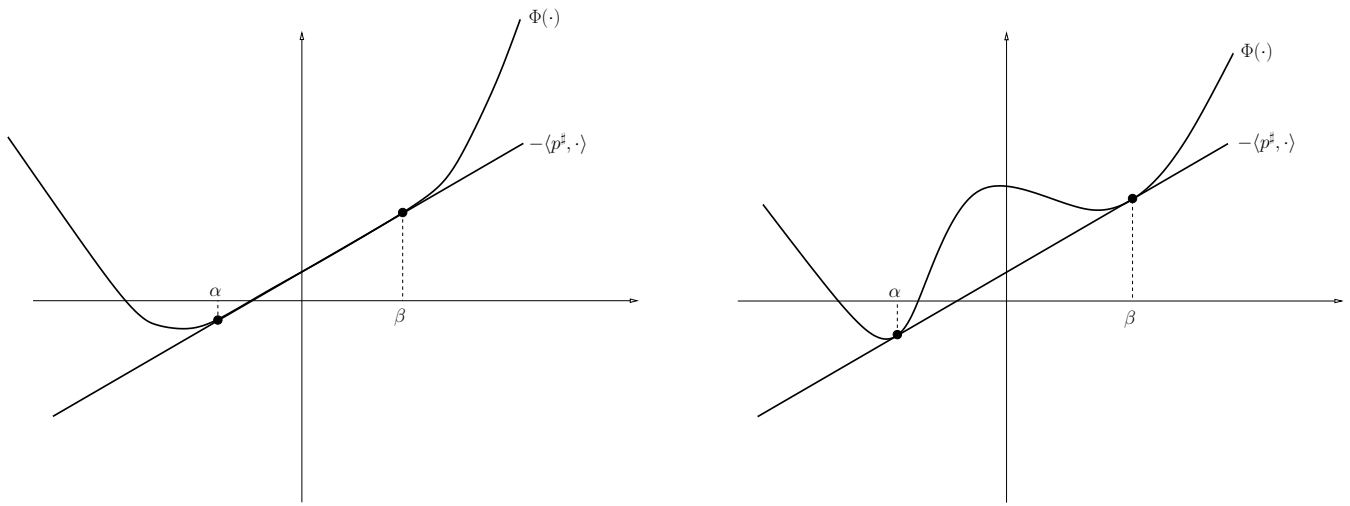


FIGURE 3 – Algorithme d'Uzawa dans le cas non convexe

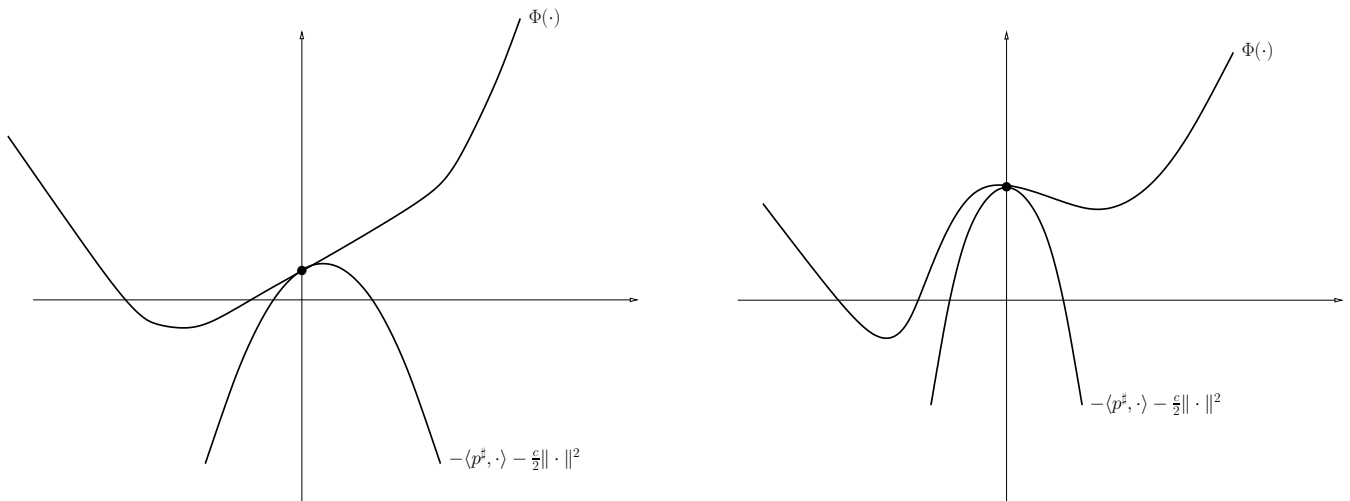


FIGURE 4 – Interprétation géométrique du Lagrangien augmenté

Dans le cas des contraintes égalité, on peut remplacer la variable  $v$  par le niveau de contraintes, et on en déduit la forme d'une itération de l'algorithme de type Uzawa correspondant :

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

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

On retrouve ainsi l'algorithme du *Lagrangien augmenté*.<sup>5</sup>

### 1.6.3 Stabilité du Lagrangien

Cet exercice consiste à traiter de manière détaillée la mise en œuvre de la dualité dans le cas d'une fonction de coût non strictement convexe (voir Figure 3). On en trouvera la solution en consultant l'exercice 2.36 du polycopié.

### 1.6.4 Problèmes équivalents

- 1) On note  $(\mathcal{P})$  le problème de minimisation en  $(u_1, u_2)$ , et  $(\mathcal{P}_1)$  (resp.  $(\mathcal{P}_2)$ ) le problème en  $u_1$  (resp.  $u_2$ ) seul.  
 $\Rightarrow$  Soit  $(u_1^\#, u_2^\#)$  une solution de  $(\mathcal{P})$ . Alors,

$$J_1(u_1^\#) + J_2(u_2^\#) \leq J_1(u_1) + J_2(u_2) \quad \forall (u_1, u_2) \in U_1^{\text{ad}} \times U_2^{\text{ad}}.$$

Cette inéquation, prise en  $u_2 = u_2^\#$ , caractérise le fait que  $u_1^\#$  est une solution de  $(\mathcal{P}_1)$ . De même pour  $(\mathcal{P}_2)$ .

$\Leftarrow$  Soit  $u_1^\#$  une solution de  $(\mathcal{P}_1)$  et  $u_2^\#$  une solution de  $(\mathcal{P}_2)$ . Alors,

$$J_1(u_1^\#) \leq J_1(u_1) \quad \forall u_1 \in U_1^{\text{ad}} \quad \text{et} \quad J_2(u_2^\#) \leq J_2(u_2) \quad \forall u_2 \in U_2^{\text{ad}}.$$

La somme de ces deux inéquations caractérise le fait que  $(u_1^\#, u_2^\#)$  une solution de  $(\mathcal{P})$ .

- 2) On note  $(\mathcal{P})$  le problème de minimisation en  $u$ , et  $(\mathcal{Q})$  le problème de minimisation sous contraintes en  $(u, v)$ .  
 $\Rightarrow$  Soit  $u^\#$  une solution de  $(\mathcal{P})$ . On pose :  $v^\# = \Theta(u^\#)$ . Alors,

$$\begin{aligned} J(u^\#, v^\#) &= J(u^\#, \Theta(u^\#)) \leq J(u, \Theta(u)) \quad \forall u \in U^{\text{ad}} \\ &\leq J(u, v) \quad \forall (u, v) \in U^{\text{ad}} \times \mathcal{C} \quad \text{tel que} \quad v = \Theta(u), \end{aligned}$$

ce qui caractérise le fait que  $(u^\#, v^\#)$  est une solution de  $(\mathcal{Q})$ .

$\Leftarrow$  Soit  $(u^\#, v^\#)$  une solution de  $(\mathcal{Q})$ . On a donc :  $v^\# = \Theta(u^\#)$ . Alors,

$$\begin{aligned} J(u^\#, \Theta(u^\#)) &= J(u^\#, v^\#) \leq J(u, v) \quad \forall (u, v) \in U^{\text{ad}} \times \mathcal{C} \quad \text{tel que} \quad v = \Theta(u) \\ &\leq J(u, \Theta(u)) \quad \forall u \in U^{\text{ad}}, \end{aligned}$$

ce qui caractérise le fait que  $u^\#$  est une solution de  $(\mathcal{P})$ .

- 3) On note  $(\mathcal{P})$  le problème de minimisation en  $(u, v)$ ,  $(\mathcal{P}_u)$  le problème de minimisation en  $u$  à  $v$  fixé et  $(\mathcal{P}_v)$  le problème de la minimisation de  $F$  par rapport à  $v$ .

$\Rightarrow$  Soit  $(u^\#, v^\#)$  une solution de  $(\mathcal{P})$ . Par définition de  $F$ , on a :

$$\begin{aligned} F(v^\#) &= \min_{u \in U^{\text{ad}}} J(u, v^\#) \\ &= J(u^\#, v^\#), \end{aligned}$$

la dernière égalité provenant du fait que le minimum de  $J$  est atteint en  $(u^\#, v^\#)$ .

Par définition du minimum,

$$\begin{aligned} J(u^\#, v^\#) &\leq J(u, v) \quad \forall (u, v) \in U^{\text{ad}} \times \mathcal{C} \\ &\leq F(v) \quad \forall v \in \mathcal{C}, \end{aligned}$$

la dernière inégalité étant obtenue en prenant le minimum en  $u$  dans les deux termes de l'inégalité précédente. Ceci caractérise le fait que  $v^\#$  est une solution de  $(\mathcal{P}_v)$ .

---

5. Le Lagrangien augmenté sera brièvement étudié au §4 de ce cours. On se reportera à l'un des cours mentionnés au §1.4 pour une étude plus approfondie des propriétés du Lagrangien augmenté, ainsi que pour son utilisation dans le cas des contraintes de type inégalité.

$\Leftarrow$  Soit  $v^\sharp$  une solution de  $(\mathcal{P}_v)$ , et soit  $u^\sharp$  une solution du problème  $(\mathcal{P}_u)$  en fixant  $v$  à la valeur  $v^\sharp$ . Par définition de  $u^\sharp$ , on a :

$$J(u^\sharp, v^\sharp) = F(v^\sharp) .$$

De plus, par définition de  $v^\sharp$  et de  $F$ , on a :

$$\begin{aligned} F(v^\sharp) &\leq F(v) \quad \forall v \in \mathcal{C} \\ &\leq J(u, v) \quad \forall (u, v) \in U^{\text{ad}} \times \mathcal{C} . \end{aligned}$$

Ceci caractérise le fait que  $(u^\sharp, v^\sharp)$  est une solution de  $(\mathcal{P})$ .





## 2 Point de vue intuitif en décomposition/coordination

### 2.1 Problème étudié

On reprend l'exemple canonique (1) donné lors de la séance précédente, où l'on veut minimiser la somme des coûts de fonctionnement d'unités sous une contrainte globale de production :

$$\min_{u_1 \in U_1^{\text{ad}}, \dots, u_N \in U_N^{\text{ad}}} \sum_{i=1}^N J_i(u_i), \quad (28a)$$

sous la contrainte

$$\sum_{i=1}^N \Theta_i(u_i) - \theta = 0. \quad (28b)$$

Chacune des variables  $u_i$  par rapport auxquelles on minimise appartient à une partie convexe fermée  $U_i^{\text{ad}}$  de  $\mathbb{R}^{n_i}$ , de telle sorte que le vecteur  $u = (u_1, \dots, u_N)$  appartient lui-même à l'ensemble convexe fermé  $U^{\text{ad}} = U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}$  contenu dans l'espace  $\mathbb{R}^n$  (avec  $n = n_1 + \dots + n_N$ ). Dans cet exemple, les applications  $\Theta_i$  sont toutes à valeurs dans le même espace  $\mathbb{R}^m$  (espace des productions).

La structure de ce problème est très particulière car :

- le fait que l'ensemble  $U^{\text{ad}}$  des contraintes non explicites se mette sous la forme d'un produit cartésien définit une *décomposition* naturelle de la variable  $u$  sous la forme  $(u_1, \dots, u_N)$ ,
- le critère  $J(u) = \sum_{i=1}^N J_i(u_i)$  et la contrainte  $\Theta(u) = \sum_{i=1}^N \Theta_i(u_i) - \theta$  sont *additifs* par rapport à cette décomposition.

C'est en exploitant cette structure que l'on va illustrer les principales méthodes de décomposition et coordination.

### 2.2 Décomposition par les prix

L'idée de la décomposition par les prix est d'*inciter* les unités à produire en leur achetant leur production à un prix donné, *identique* pour toutes les unités, et à faire évoluer ce prix jusqu'à ce que la somme des productions des unités soit égale à la valeur  $\theta$  requise. Le sous-problème d'optimisation associé à la  $i$ -ème unité que l'on résout dans cette méthode est donc de la forme :

$$\min_{u_i \in U_i^{\text{ad}}} J_i(u_i) + \langle p^{(k)}, \Theta_i(u_i) \rangle, \quad (29)$$

où  $p^{(k)}$  est le prix d'achat de la production. Cette minimisation fournit une valeur optimale  $u_i^{(k+1)}$  pour les variables de la  $i$ -ème unité, la production associée étant  $\Theta_i(u_i^{(k+1)})$ . La production fournie par l'ensemble des unités en réponse au prix  $p^{(k)}$  vaut donc  $\sum_{i=1}^N \Theta_i(u_i^{(k+1)})$ , et on ajuste le prix à la baisse ou à la hausse suivant que la production totale est inférieure ou supérieure à la demande  $\theta$ .<sup>6</sup>

On montre que, sous l'hypothèse de l'existence d'un point selle du Lagrangien du problème (28), la concaténation des solutions des sous-problèmes fournit la solution du problème global *pourvu que* la contrainte globale de production soit satisfaite.

---

6. Dans un contexte économique, la production  $\Theta_i(u_i)$  d'une unité est une quantité positive, et l'acheter correspond donc à utiliser un prix  $p^{(k)}$  négatif. Augmenter ce prix consiste à le diminuer en valeur absolue, et donc à rendre l'offre d'achat moins attractive. On notera que le fait que les prix soient négatifs provient d'une interprétation *économique* et ne correspond en rien à une propriété *mathématique*.

Avec cette hypothèse d'existence de point selle, il est assez naturel de former le Lagrangien associé au problème initial (28) :

$$L(u_1, \dots, u_N, p) = \sum_{i=1}^N J_i(u_i) + \left\langle p, \sum_{i=1}^N \Theta_i(u_i) - \theta \right\rangle ,$$

Lagrangien défini sur l'ensemble  $U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}} \times \mathbb{R}^m$ , et d'en considérer la maxi-minimisation :

$$\max_{p \in \mathbb{R}^m} \min_{u_1 \in U_1^{\text{ad}}, \dots, u_N \in U_N^{\text{ad}}} L(u_1, \dots, u_N, p) ,$$

pour résoudre le problème (28). L'algorithme que l'on utilise de manière standard pour effectuer cette maxi-minimisation est celui d'*Uzawa*, avec dans ce cas les particularités suivantes.

- La minimisation en  $u$  du Lagrangien à  $p = p^{(k)}$  fixé se décompose unité par unité grâce aux hypothèses portant sur la forme cartésienne de l'ensemble admissible et sur l'additivité des fonctions  $J$  et  $\Theta$ . En effet, le Lagrangien se met sous la forme :

$$L(u_1, \dots, u_N, p^{(k)}) = \sum_{i=1}^N \left( J_i(u_i) + \langle p^{(k)}, \Theta_i(u_i) \rangle \right) - \langle p^{(k)}, \theta \rangle ;$$

le terme  $\langle p^{(k)}, \theta \rangle$  est indépendant des  $u_i$  et peut être oublié lors de la phase de minimisation, qui consiste donc à minimiser la somme des fonctions  $J_i(u_i) + \langle p^{(k)}, \Theta_i(u_i) \rangle$  dépendant chacune uniquement de  $u_i$ , sous des contraintes  $u_i \in U_i^{\text{ad}}$  découplées. Il est par conséquent équivalent de minimiser chaque fonction individuellement, à savoir :

$$\min_{u_i \in U_i^{\text{ad}}} J_i(u_i) + \langle p^{(k)}, \Theta_i(u_i) \rangle ,$$

et la solution de ce  $i$ -ème sous-problème de minimisation est notée  $u_i^{(k+1)}$ . On retrouve bien l'idée intuitive de la décomposition par les prix qui conduisait aux sous-problèmes (29).

- Pour la maximisation au point  $p^{(k)}$  de la fonction duale  $H$  (celle qui résulte de la minimisation en  $u$  du Lagrangien), on dispose de l'expression du gradient de cette fonction duale en  $p^{(k)}$  :

$$\nabla H(p^{(k)}) = \sum_{i=1}^N \Theta_i(u_i^{(k+1)}) - \theta ,$$

expression qui utilise les solutions  $u_i^{(k+1)}$  des sous-problèmes de minimisation précédents.

On rappelle qu'il faut supposer, en plus des hypothèses classiques assurant l'existence du point selle, que les fonctions  $J_i$  soient strictement convexes pour que les solutions  $u_i^{(k+1)}$  soient uniques et donc que la fonction  $H$  soit alors différentiable.

L'algorithme dit de décomposition par les prix consiste donc tout simplement à mettre en œuvre l'algorithme d'*Uzawa* sur le Lagrangien du problème initial. La  $k$ -ème itération de cet algorithme est constituée des 2 étapes suivantes :

- (1) **Décomposition** : à prix  $p^{(k)}$  fixé, résolution des  $N$  sous-problèmes indépendants :

$$\min_{u_i \in U_i^{\text{ad}}} J_i(u_i) + \langle p^{(k)}, \Theta_i(u_i) \rangle \quad i = 1, \dots, N , \quad (30)$$

la solution du  $i$ -ème sous-problème étant notée  $u_i^{(k+1)}$ .

(2) **Coordination** : mise à jour du prix pour l'itération suivante de l'algorithme :

$$p^{(k+1)} = p^{(k)} + \rho \left( \sum_{i=1}^N \Theta_i(u_i^{(k+1)}) - \theta \right). \quad (31)$$

Durant cette itération, les unités utilisent toutes le *même* prix  $p^{(k)}$  et calculent en fonction de ce prix leur niveau optimal de production  $\Theta_i(u_i^{(k+1)})$ . Le prix est ensuite remis à jour pour tenir compte de l'écart entre la production totale de l'itération et la production requise  $\theta$ . La contrainte portant sur la production totale des unités n'est satisfaite qu'à la *convergence* du processus itératif. Le mécanisme d'échange d'informations entre les unités et le niveau de coordination est schématisé sur la figure 5.

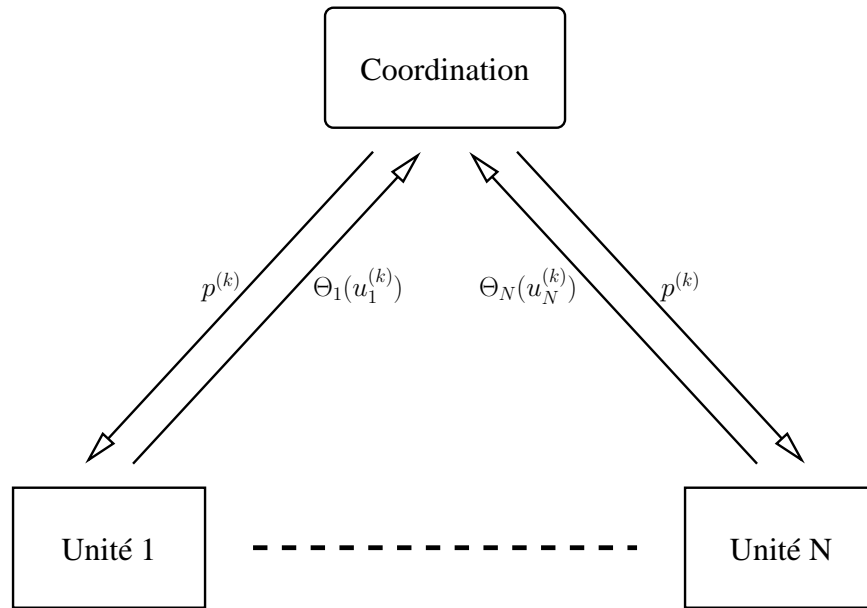


FIGURE 5 – Schéma de principe de la décomposition par les prix

En résumé, la mise en œuvre de la méthode de décomposition par les prix sur un problème d'optimisation passe par les étapes suivantes :

1. on se donne une décomposition du problème, c'est-à-dire un partitionnement  $u = (u_1, \dots, u_N)$  des variables sur lesquelles porte l'optimisation ;
2. on s'assure que les fonctions de coût et de contrainte du problème sont additives par rapport à cette décomposition ;
3. on forme le Lagrangien du problème, qui est donc lui aussi additif par rapport au partitionnement  $(u_1, \dots, u_N)$  des variables d'optimisation ;
4. on utilise l'algorithme d'Uzawa, dont la seule particularité par rapport au cas général est que l'étape de minimisation se décompose en  $N$  sous-problèmes indépendants ne faisant chacun intervenir qu'un seul  $u_i$ .

### 2.3 Décomposition par allocation de ressources

L'idée de la décomposition par allocation est duale de la précédente : elle consiste à *obliger* les unités à produire une quantité donnée, et à faire évoluer ces quantités de telle sorte que l'on obtienne

la solution du problème global. Pour cela, on se donne des valeurs  $v_i^{(k)} \in \mathbb{R}^m$  de production pour chaque unité  $i$  respectant la contrainte globale de production, à savoir :

$$\sum_{i=1}^N v_i^{(k)} - \theta = 0, \quad (32)$$

et on résout pour chaque unité  $i$  le sous-problème d'optimisation suivant :

$$\min_{u_i \in U_i^{\text{ad}}} J_i(u_i), \quad (33a)$$

sous les contraintes :

$$\Theta_i(u_i) - v_i^{(k)} = 0. \quad (33b)$$

On appelle *allocation* un vecteur  $(v_1^{(k)}, \dots, v_N^{(k)})$  vérifiant la contrainte globale (32) de production, et on note  $p_i^{(k+1)}$  le multiplicateur optimal associé à la contrainte locale de production du sous-problème (33). Utilisant l'*interprétation marginaliste des multiplicateurs* (voir §1.3.3), on montre que l'allocation  $v_i^{(k)}$  ne peut correspondre à la solution du problème initial (28) que si les multiplicateurs  $p_i^{(k+1)}$  sont tous *égaux* entre eux.<sup>7</sup>

Plus précisément, on montre que, sous l'hypothèse de l'existence d'un point selle de chacun des sous-problèmes (33), la concaténation des solutions des sous-problèmes fournit la solution du problème global pourvu que tous les multiplicateurs associés soient égaux.

Notant  $\Phi_i$  la fonction de perturbation associée au sous-problème (33) :

$$\Phi_i(v_i^{(k)}) = \min_{u_i \in U_i^{\text{ad}}} \left\{ J_i(u_i) \text{ sous } \Theta_i(u_i) - v_i^{(k)} = 0 \right\},$$

il est facile de voir que le problème initial (28) est équivalent au nouveau problème suivant :<sup>8</sup>

$$\min_{(v_1, \dots, v_N)} \sum_{i=1}^N \Phi_i(v_i), \quad (34a)$$

sous les contraintes :

$$\sum_{i=1}^N v_i - \theta = 0. \quad (34b)$$

La résolution du problème (34) peut être effectuée à l'aide d'un algorithme de *gradient projeté*. Notant  $\Sigma_\theta$  le sous-espace affine définissant les allocations :

$$\Sigma_\theta = \left\{ v = (v_1, \dots, v_N) \in (\mathbb{R}^m)^N, \text{ tels que } \sum_{i=1}^N v_i - \theta = 0 \right\},$$

la mise à jour des variables  $v$  à la  $k$ -ème itération de l'algorithme de gradient projeté s'écrit :

$$\begin{pmatrix} v_1^{(k+1)} \\ \vdots \\ v_N^{(k+1)} \end{pmatrix} = \text{proj}_{\Sigma_\theta} \begin{pmatrix} v_1^{(k)} - \rho \nabla \Phi_1(v_1^{(k)}) \\ \vdots \\ v_N^{(k)} - \rho \nabla \Phi_N(v_N^{(k)}) \end{pmatrix}.$$

7. Si les multiplicateurs associés aux unités  $i$  et  $j$  sont tels que  $p_i^{(k+1)} < p_j^{(k+1)}$ , il suffit de diminuer la valeur de l'allocation de l'unité  $i$  et d'augmenter celle de l'unité  $j$  de la même quantité pour améliorer le coût global.

8. La fonction  $\Phi_i$  étant définie sur  $\mathbb{R}^m$ , le problème d'optimisation (34) est posé sur l'espace  $(\mathbb{R}^m)^N$ .

La projection sur  $\Sigma_\theta$  du vecteur dont la  $i$ -ème composante (dans  $\mathbb{R}^m$ ) a pour valeur  $v_i^{(k)} - \rho \nabla \Phi_i(v_i^{(k)})$  se calcule analytiquement (voir le premier exercice du §2.5). Elle conduit pour chaque composante de la projection à la *formule explicite* suivante :

$$v_i^{(k+1)} = v_i^{(k)} - \rho \left( \nabla \Phi_i(v_i^{(k)}) - \frac{1}{N} \sum_{j=1}^N \nabla \Phi_j(v_j^{(k)}) \right).$$

Par l'interprétation marginaliste du multiplicateur, on sait que le gradient  $\nabla \Phi_i(v_i^{(k)})$  de la fonction de perturbation est égale, à un changement de signe près, au multiplicateur optimal  $p_i^{(k+1)}$  calculé lors de la résolution du sous-problème (33), et donc :

$$v_i^{(k+1)} = v_i^{(k)} + \rho \left( p_i^{(k+1)} - \frac{1}{N} \sum_{j=1}^N p_j^{(k+1)} \right).$$

L'algorithme de décomposition et coordination par allocation consiste donc à mettre en œuvre l'algorithme du gradient projeté sur le problème (34). Il faut pour cela, d'une part obtenir les multiplicateurs  $p_i^{(k+1)}$ , et d'autre part mettre à jour les composantes  $v_i^{(k+1)}$  de l'allocation. La  $k$ -ème itération de l'algorithme comporte donc les 2 étapes suivantes :

**(1) Décomposition** : à allocation  $(v_1^{(k)}, \dots, v_N^{(k)})$  fixée, résolution des  $N$  sous-problèmes :

$$\min_{u_i \in U_i^{\text{ad}}} J_i(u_i) \quad \text{sous} \quad \Theta_i(u_i) - v_i^{(k)} = 0 \quad i = 1, \dots, N, \quad (35)$$

le multiplicateur optimal associé à la contrainte du  $i$ -ème sous-problème étant noté  $p_i^{(k+1)}$ .<sup>9</sup>

**(2) Coordination** : mise à jour de l'allocation pour l'itération suivante de l'algorithme :

$$v_i^{(k+1)} = v_i^{(k)} + \rho \left( p_i^{(k+1)} - \frac{1}{N} \sum_{j=1}^N p_j^{(k+1)} \right) \quad i = 1, \dots, N. \quad (36)$$

Les quantités  $v_i^{(k+1)}$  produites par les unités à l'issue de l'étape de coordination satisfont par construction la contrainte globale de production : le vecteur  $(v_1^{(k+1)}, \dots, v_N^{(k+1)})$  est donc une allocation. Chaque unité calcule le prix marginal associé à la quantité qui lui est imposée, et les quantités sont remises à jour afin de réduire les écarts entre les différents prix marginaux. Dans la méthode de décomposition par allocation, toutes les contraintes sont satisfaites à chaque itération, mais les prix ne sont égaux qu'à la *convergence* du processus itératif.<sup>10</sup> Le mécanisme d'échange des informations entre les unités et le niveau de coordination est schématisé sur la figure 6. On notera que le sens de circulation de ces informations est l'*inverse* du sens correspondant à la méthode de décomposition par les prix (comparer les figures 5 et 6).

**Remarque 5.** Il est important de se rendre compte que l'ensemble des valeurs  $u_i$  vérifiant la totalité des contraintes du sous-problème (33) à une itération donnée peut correspondre à l'*ensemble vide*, dans la mesure où l'on a ajouté au sous-problème une contrainte  $\Theta_i(u_i) - v_i^{(k)} = 0$  pouvant être *incompatible* avec la contrainte  $u_i \in U_i^{\text{ad}}$ . Lorsque cette situation se produit, l'algorithme se bloque : la méthode de décomposition par les quantités ne peut pas être utilisée, du moins de manière directe.

9. On remarquera que les solutions  $u_i^{(k+1)}$  des sous-problèmes ne sont pas utilisées durant les itérations de l'algorithme ; elles fournissent par contre une solution du problème à l'issue des itérations.

10. L'examen de la relation (36) indique que l'allocation  $v^{(k)}$  est stationnaire dès que le prix de chaque unité est égal à la moyenne des prix de toutes les unités, c'est-à-dire dès que les prix des unités sont tous égaux entre eux.

# Décomposition par allocation de ressources admissibilité versus différentiabilité

Pierre Carpentier

◇

pierre.carpentier@ensta-paris.fr

2 mars 2021

## Résumé

En reprenant l'article [1], on montre comment s'affranchir du problème d'admissibilité fréquemment rencontré dans la méthode de décomposition par allocation de ressources, par le biais d'un argument de pénalisation non différentiable. Ce document pourrait servir pour faire un sujet d'examen du cours sur la décomposition.

## 1 Position du problème

On part d'un problème d'optimisation additif classique :

$$\min_{(u_i \in U_i^{\text{ad}} \subset \mathbb{R}^{n_i})_{i=1, \dots, N}} \sum_{i=1}^N J_i(u_i) \quad (1a)$$

sous les contraintes

$$\sum_{i=1}^N \Theta_i(u_i) - \theta \leq 0 \in \mathbb{R}^m, \quad (1b)$$

dont la valeur optimale est notée  $J^*$ . On fait les hypothèses suivantes.

**Hypothèse 1.** Pour tout  $i \in \{1, \dots, N\}$ , les ensembles  $U_i^{\text{ad}}$  sont convexes fermés, les fonctions  $J_i$  sont convexes propres s.c.i. et les fonctions  $\Theta_i$  sont  $\mathbb{R}^m$ -convexes continues.

**Hypothèse 2.** Le problème (1) admet un point-selle  $(u_1^*, \dots, u_N^*, \lambda^*) \in \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_N} \times \mathbb{R}_+^m$ .

## 2 Allocation de ressources et problème restreint

### 2.1 Principe de l'allocation

On introduit l'ensemble des allocations :

$$V^{\text{ad}} = \left\{ (v_1, \dots, v_N) \in \mathbb{R}^{mN}, \sum_{i=1}^N v_i = \theta \right\}.$$

FIGURE 6 – Schéma de principe de la décomposition par allocation

En résumé, la mise en œuvre de la méthode de décomposition par allocation de ressources sur un problème d'optimisation passe par les étapes suivantes :

1. on se donne une décomposition du problème, c'est-à-dire un partitionnement  $u = (u_1, \dots, u_N)$  des variables sur lesquelles porte l'optimisation ;
2. on s'assure que les fonctions de coût et de contrainte du problème sont additives par rapport à cette décomposition ;
3. on introduit de nouvelles variables  $v$  (et donc de nouvelles contraintes) de manière à former un nouveau problème équivalent au problème initial, de telle sorte que ce nouveau problème se décompose lorsque les variables  $v$  sont fixées en  $N$  sous-problèmes indépendants, chaque sous-problème ne faisant intervenir qu'un seul  $u_i$  ;
4. on utilise un algorithme de type gradient sur le problème de minimisation en  $v$ , le calcul du gradient nécessitant la résolution des  $N$  sous-problèmes en  $u_i$ .

## 2.4 Décomposition par prédiction

### 2.4.1 Présentation sur un exemple simplifié

L'idée de la décomposition par prédiction est de *combinaison* les deux méthodes de décomposition précédentes. On la présente pour commencer dans le cas de  $N$  unités liées par une *unique* contrainte couplante ( $\theta \in \mathbb{R}$ ) :

$$\min_{u_i \in U_i^{\text{ad}}} \sum_{i=1}^N J_i(u_i), \quad (37a)$$

sous la contrainte

$$\sum_{i=1}^N \Theta_i(u_i) - \theta = 0. \quad (37b)$$

Dans ce cadre simplifié, on voudrait que l'une des unités respecte un niveau de production imposé (comme dans la méthode par allocation), et que les autres unités soient incitées à produire par l'intermédiaire d'un multiplicateur (comme dans la méthode par les prix). Le sous-problème d'optimisation associé à l'unité "traitée par allocation" ( $i_0$ -ème unité) s'écrit :

$$\min_{u_{i_0} \in U_{i_0}^{\text{ad}}} J_{i_0}(u_{i_0}) \quad \text{sous} \quad \Theta_{i_0}(u_{i_0}) - v^{(k)} = 0, \quad (38)$$

tandis que le sous-problème d'optimisation associée à une unité "traitée par les prix" ( $i$ -ème unité, avec  $i \neq i_0$ ) est de la forme :

$$\min_{u_i \in U_i^{\text{ad}}} J_i(u_i) + \langle p^{(k)}, \Theta_i(u_i) \rangle. \quad (39)$$

On constate, d'une part que la somme des productions réalisées par les unités  $i \neq i_0$  peut servir pour déterminer la production à imposer à l'unité  $i_0$ , et d'autre part que le prix marginal associé à la contrainte de l'unité  $i_0$  peut servir de prix pour la production des unités  $i \neq i_0$ . Plus précisément,

- si la résolution des unités  $i \neq i_0$  avec un prix  $p^{(k-1)}$  fixé conduit à des productions  $\Theta_i(u_i^{(k)})$ , un niveau de production qu'il est raisonnable d'imposer à l'unité  $i_0$  est :

$$v^{(k)} = \theta - \sum_{i \neq i_0} \Theta_i(u_i^{(k)}), \quad (40)$$

car alors, la contrainte dans (38) correspond à la contrainte du problème initial ;



- si la résolution de l'unité  $i_0$  avec une production imposée  $v^{(k-1)}$  fournit la valeur  $p_{i_0}^{(k)}$  du multiplicateur, le prix incitatif à la production pour les unités  $i \neq i_0$  peut être :

$$p^{(k)} = p_{i_0}^{(k)}, \quad (41)$$

car ainsi, ces unités comparent dans (39) leur coût de production au coût marginal de l'unité  $i_0$ . Dans ce schéma, il n'y a donc pas de niveau de coordination comme il en existait dans les deux méthodes précédentes, puisque chaque unité joue un rôle dans la coordination des autres unités. Par contre, il y a un *choix* à effectuer pour savoir quelle unité sera traitée comme dans la méthode par allocation, les autres unités étant alors traitées comme dans la méthode par les prix. On a représenté sur la figure 7 les deux choix possibles dans le cas  $N = 2$  (problème à deux unités).

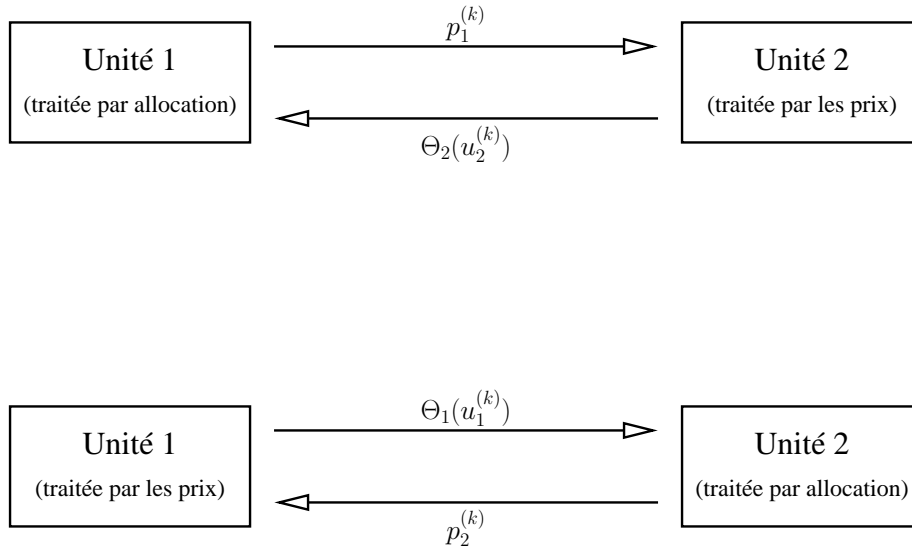


FIGURE 7 – Schéma de principe de la décomposition par prédiction

**Remarque 6.** De ce choix va dépendre l'efficacité de la méthode de décomposition par prédiction. En effet, les méthodes par les prix et par allocation présentent chacune en pratique des avantages et des inconvénients ; un choix bien raisonné peut permettre de cumuler les avantages des deux méthodes (alors qu'un mauvais choix risque de cumuler les inconvénients...).

On montre alors que, sous l'hypothèse d'existence d'un point selle pour le sous-problème (38) et d'une solution pour les sous-problèmes (39), la concaténation des solutions des  $N$  sous-problèmes fournit la solution du problème global pourvu que les relations (40) et (41) soient vérifiées par les solutions locales des sous-problèmes.

Un premier algorithme de décomposition et coordination par prédiction est alors directement suggéré par les considérations précédentes : la  $k$ -ème itération de cet algorithme consiste, à partir de valeurs  $v^{(k)}$  et  $p^{(k)}$  données, à résoudre les sous-problèmes (38) et (39), puis à obtenir les nouvelles valeurs  $v^{(k+1)}$  et  $p^{(k+1)}$  à l'aide des relations (40) et (41). Cette itération a donc la forme suivante.

**Unité  $i_0$  :** à production  $v^{(k)}$  fixée,  
— résolution du sous-problème :

$$\min_{u_{i_0} \in U_{i_0}^{\text{ad}}} J_{i_0}(u_{i_0}) \quad \text{sous} \quad \Theta_{i_0}(u_{i_0}) - v^{(k)} = 0,$$

la solution et le multiplicateur optimal étant notés  $u_{i_0}^{(k+1)}$  et  $p_{i_0}^{(k+1)}$ ,

— calcul du prix  $p^{(k+1)}$  qui sera utilisé par les unités  $i \neq i_0$  à l'itération suivante :

$$p^{(k+1)} = p_{i_0}^{(k+1)} .$$

**Unités  $i \neq i_0$  :** à prix  $p^{(k)}$  fixé,

— résolution des sous-problèmes :

$$\min_{u_i \in U_i^{\text{ad}}} J_i(u_i) + \langle p^{(k)}, \Theta_i(u_i) \rangle ,$$

les solutions optimales étant notées  $u_i^{(k+1)}$ ,

— calcul de la production  $v^{(k+1)}$  qui sera utilisée par l'unité  $i_0$  à l'itération suivante :

$$v^{(k+1)} = \theta - \sum_{i \neq i_0} \Theta_i(u_i^{(k+1)}) .$$

On constate que le traitement complet de l'unité  $i_0$  dans l'itération précédente consiste, du point de vue de la coordination, à calculer un prix  $p^{(k+1)}$  à partir de la donnée d'une production  $v^{(k)}$ , ce que l'on représente symboliquement par un opérateur  $\Psi$  :

$$p^{(k+1)} = \Psi(v^{(k)}) .$$

De même, l'ensemble des calculs effectués sur l'ensemble des unités  $i \neq i_0$  revient à obtenir une production  $v^{(k+1)}$  à partir d'un prix  $p^{(k)}$ , ce que l'on modélise par un opérateur  $\Xi$  :

$$v^{(k+1)} = \Xi(p^{(k)}) .$$

Avec ces nouvelles notations, une itération de la méthode par prédiction s'écrit sous la forme :

$$\begin{pmatrix} v^{(k+1)} \\ p^{(k+1)} \end{pmatrix} = \begin{pmatrix} 0 & \Xi \\ \Psi & 0 \end{pmatrix} \begin{pmatrix} v^{(k)} \\ p^{(k)} \end{pmatrix} ,$$

ce qui correspond à l'itération d'une méthode de *point fixe* sur le couple  $(v^{(k)}, p^{(k)})$ . Dans cette mise en œuvre *parallèle*<sup>11</sup> de la méthode, la solution obtenue à chaque itération ne satisfait pas la contrainte globale de production :  $\Theta_{i_0}(u_{i_0}^{(k)}) + \sum_{i \neq i_0} \Theta_i(u_i^{(k)}) \neq \theta$ . Par contre, la méthode est telle que l'on a toujours :  $\Theta_{i_0}(u_{i_0}^{(k+1)}) + \sum_{i \neq i_0} \Theta_i(u_i^{(k)}) = \theta$ . Si l'on veut disposer d'une solution admissible à chaque itération, il suffit dans ce cas de mettre en œuvre une version *séquentielle* de l'algorithme, dont la  $k$ -ème itération s'écrit à l'aide des opérateurs  $\Psi$  et  $\Xi$  sous la forme suivante :

— résolution des unités  $i \neq i_0$  à prix  $p^{(k)}$  fixé :

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

— résolution de l'unité  $i_0$  à production  $v^{(k+1)}$  fixé :

$$p^{(k+1)} = \Psi(v^{(k+1)}) .$$

Dans cette mise en œuvre, la résolution à l'itération  $k$  du sous-problème  $i_0$  dépend de la solution du sous-problème  $i \neq i_0$  à la même itération. La méthode de point fixe associée s'écrit :

$$p^{(k+1)} = \Psi \circ \Xi(p^{(k)}) ,$$

et on vérifie trivialement que la solution  $(u_1^{(k+1)}, \dots, u_N^{(k+1)})$  de chaque itération de cette méthode de point fixe satisfait la contrainte globale de production du problème initial. On notera enfin qu'il existe une seconde version séquentielle de l'algorithme, à savoir :  $v^{(k+1)} = \Xi \circ \Psi(v^{(k)})$ , ne fournissant quant à elle pas à chaque itération une solution admissible.

11. car les  $N$  sous-problèmes peuvent être résolus simultanément, par exemple sur un calculateur parallèle

**Remarque 7.** Il est possible d'accélérer la convergence d'un algorithme de point fixe en *relaxant* les variables sur lesquelles porte l'algorithme.

— Dans la mise en œuvre parallèle, les valeurs  $(v^{(k+1)}, p^{(k+1)})$  se calculent alors par les formules :

$$v^{(k+1)} = (1 - \epsilon) v^{(k)} + \epsilon \left( \theta - \sum_{i \neq i_0} \Theta_i(u_i^{(k+1)}) \right),$$

$$p^{(k+1)} = (1 - \rho) p^{(k)} + \rho p_{i_0}^{(k+1)},$$

où  $\epsilon$  et  $\rho$  sont des nombres strictement positifs. On parle de *sur-relaxation* si ces nombres sont plus grands que 1, et de *sous-relaxation* s'ils sont plus petits que 1. Le cas  $\epsilon = \rho = 1$  correspond à l'algorithme sans relaxation.

— Dans la version séquentielle, on ne relaxe que la valeur  $p^{(k+1)}$  si l'on veut préserver l'admissibilité de la solution :

$$p^{(k+1)} = (1 - \rho) p^{(k)} + \rho p_{i_0}^{(k+1)}.$$

Il existe une abondante littérature sur les méthodes de point fixe, que l'on pourra consulter pour plus de détails.

L'interprétation mathématique que l'on donne de la méthode de décomposition et coordination par prédiction repose sur l'idée d'écrire un problème équivalent au problème initial en introduisant une *nouvelle variable*  $v$  et en se servant de cette variable pour *scinder la contrainte initiale* en deux morceaux. Si l'on choisit d'affecter la contrainte initiale à l'unité  $i_0$ , le premier morceau de la contrainte ne doit contenir que le terme de production correspondant à l'unité  $i_0$ , le second morceau incorporant les termes de production correspondant à toutes les autres unités.

Le problème équivalent à (37) qui résulte de ces transformations est alors de la forme :

$$\min_{u_1 \in U_1^{\text{ad}}, \dots, u_N \in U_N^{\text{ad}}, v \in \mathbb{R}} \sum_{i=1}^N J_i(u_i), \quad (42a)$$

$$\text{sous les contraintes :} \\ \Theta_{i_0}(u_{i_0}) - v = 0, \quad (42b)$$

$$\sum_{i \neq i_0} \Theta_i(u_i) - \theta + v = 0. \quad (42c)$$

Notant  $p_{i_0}$  le multiplicateur associé à la première contrainte (42b) et  $p$  celui associé à la seconde contrainte (42c), le Lagrangien de ce nouveau problème est :

$$\mathcal{L}(u_1, \dots, u_N, v, p_{i_0}, p) = \sum_{i=1}^N J_i(u_i) + p_{i_0}(\Theta_{i_0}(u_{i_0}) - v) + p\left(\sum_{i \neq i_0} \Theta_i(u_i) - \theta + v\right).$$

On suppose alors qu'il est licite de réorganiser les opérations de minimisation et de maximisation de ce Lagrangien de la manière suivante :

$$\max_{p \in \mathbb{R}} \min_{v \in \mathbb{R}} \left( \max_{p_{i_0} \in \mathbb{R}} \min_{u_1 \in U_1^{\text{ad}}, \dots, u_N \in U_N^{\text{ad}}} \mathcal{L}(u_1, \dots, u_N, v, p_{i_0}, p) \right).$$

La maxi-minimisation la plus interne dans cette expression s'effectue à  $p$  et  $v$  fixés, ces variables étant ensuite optimisées lors de la maxi-minimisation "externe". Fixant les valeurs  $p = p^{(k)}$  et  $v = v^{(k)}$ , la maxi-minimisation interne s'écrit :

$$\max_{p_{i_0} \in \mathbb{R}} \min_{u_1 \in U_1^{\text{ad}}, \dots, u_N \in U_N^{\text{ad}}} J_{i_0}(u_{i_0}) + p_{i_0}(\Theta_{i_0}(u_{i_0}) - v^{(k)}) + \sum_{i \neq i_0} J_i(u_i) + p^{(k)}\left(\sum_{i \neq i_0} \Theta_i(u_i) - \theta + v^{(k)}\right).$$

Comme le terme  $p^{(k)}(-\theta + v^{(k)})$  est constant à cette étape du calcul, cette maxi-minimisation se réécrit sous la forme de  $N$  problèmes de minimisation indépendants, à savoir :

$$\text{pour } i = i_0 : \quad \min_{u_{i_0} \in U_{i_0}^{\text{ad}}} J_{i_0}(u_{i_0}) \quad \text{sous } \Theta_{i_0}(u_{i_0}) - v^{(k)} = 0, \\ \text{pour } i \neq i_0 : \quad \min_{u_i \in U_i^{\text{ad}}} J_i(u_i) + p^{(k)}\Theta_i(u_i),$$

dont on note les solutions respectivement  $(u_{i_0}^{(k+1)}, p_{i_0}^{(k+1)})$  et  $\{u_i^{(k+1)}\}_{i \neq i_0}$ .

Pour traiter la maxi-minimisation externe en  $(p, v)$ , on choisit d'annuler les gradients partiels du Lagrangien  $\mathcal{L}$  par rapport à ces variables :

$$\nabla_v \mathcal{L}(u_1, \dots, u_N, v, p_{i_0}, p) = p - p_{i_0} = 0 \quad ; \quad \nabla_p \mathcal{L}(u_1, \dots, u_N, v, p_{i_0}, p) = \sum_{i \neq i_0} \Theta_i(u_i) - \theta + v = 0,$$

et on en déduit la mise à jour pour l'itération suivante des variables  $p$  et  $v$  :

$$p^{(k+1)} = p_{i_0}^{(k+1)} \quad ; \quad v^{(k+1)} = \theta - \sum_{i \neq i_0} \Theta_i(u_i^{(k+1)}).$$

On retrouve exactement les sous-problèmes formulés dans le cadre de la méthode de décomposition par prédiction, ainsi que les formules de remise à jour de la production et du prix de cette méthode.

**Remarque 8.** Il est a priori surprenant de constater que, dans cette interprétation, c’est le gradient partiel du Lagrangien  $\mathcal{L}$  par rapport à  $v$  (resp.  $p$ ) qui est utilisé pour remettre à jour la variable  $p$  (resp.  $v$ ). L’interprétation en terme de Lagrangien suggère d’utiliser plutôt le gradient partiel de  $\mathcal{L}$  par rapport à  $v$  (resp.  $p$ ) pour remettre à jour la variable  $v$  (resp.  $p$ ) par un algorithme de type Arrow-Hurwicz, qui s’écrit dans ce cas :

$$\begin{aligned} v^{(k+1)} &= v^{(k)} - \epsilon \nabla_v \mathcal{L}(u_1^{(k+1)}, \dots, u_N^{(k+1)}, v^{(k)}, p_{i_0}^{(k+1)}, p^{(k)}) \\ p^{(k+1)} &= p^{(k)} + \rho \nabla_p \mathcal{L}(u_1^{(k+1)}, \dots, u_N^{(k+1)}, v^{(k)}, p_{i_0}^{(k+1)}, p^{(k)}) , \end{aligned}$$

soit les formules de remise à jour suivantes :

$$\begin{aligned} v^{(k+1)} &= v^{(k)} - \epsilon (p^{(k)} - p_{i_0}^{(k+1)}) \\ p^{(k+1)} &= p^{(k)} + \rho \left( \sum_{i \neq i_0} \Theta_i(u_i^{(k+1)}) - \theta + v^{(k)} \right) , \end{aligned}$$

On dispose ainsi, en marge de l’algorithme de type “point fixe” exposé précédemment, d’un algorithme de type “Arrow-Hurwicz” pour la remise à jour des variables  $(v^{(k)}, p^{(k)})$ . Ces deux algorithmes sont en pratique assez complémentaires, le premier convergeant plus rapidement que le second dans le cas d’interaction faible<sup>12</sup> entre les sous-problèmes, et le second convergeant même dans le cas d’interaction forte.

## 2.4.2 Esquisse du cas général

Dans l’exemple précédent, il a fallu choisir à quelle unité était allouée la contrainte de production pour mettre en œuvre la méthode par prédiction. Ceci se généralise sans difficulté au cas de  $N$  unités couplées par  $m$  contraintes. Il faut alors choisir, *pour chaque contrainte* du problème, à quelle unité on alloue cette contrainte. La répartition des contraintes sur les unités correspond en fait à une *décomposition de l’espace d’arrivée des contraintes*  $\mathbb{R}^m$  selon les sous-problèmes. Plus précisément, on décompose l’ensemble  $\{1, \dots, m\}$  des indices des contraintes du problème en  $N$  sous-ensembles,<sup>13</sup> le  $i$ -ème sous-ensemble contenant les indices des contraintes affectées à la  $i$ -ème unité. Ainsi, dans la méthode par prédiction, chaque unité ne “voit” que la partie des contraintes qui lui est affectée. C’est là une différence fondamentale avec la méthode par les prix (dans laquelle toutes les contraintes “disparaissent” par dualité) et avec la méthode par allocation (dans laquelle chaque sous-problème “voit” toutes les contraintes). Une fois la répartition des contraintes effectuée, le sous-problème associé à l’unité  $i$  se formule :

- en fixant dans les contraintes affectées au sous-problème  $i$  les productions des unités  $j \neq i$  aux valeurs obtenues lors de l’itération précédente ;
- en ajoutant dans le critère du sous-problème  $i$  les termes de coût correspondant à la production de l’unité  $i$  qui apparaissent dans les contraintes affectées aux unités  $j \neq i$ .

Pour illustrer cette procédure, on détaille le cas où la  $\ell$ -ème contrainte du problème, de la forme :

$$\sum_{j=1}^N \Theta_{\ell,j}(u_j) - \theta_\ell = 0 ,$$

est affectée à l’unité  $i$ . Lors de la  $k$ -ème itération de décomposition et coordination par prédiction, le traitement de cette  $\ell$ -ème contrainte affecte l’ensemble des sous-problèmes de la manière suivante :

12. quand la contrainte est affectée à l’unité  $i_0$ , interaction faible signifie que le terme  $\Theta_{i_0}$  est prépondérant dans la contrainte initiale

13. certains de ces sous-ensembles pouvant être vides

1. on incorpore dans le sous-problème associé à l'unité  $i$  une contrainte de la forme :

$$\Theta_{\ell,i}(u_i) + \sum_{j \neq i} \Theta_{\ell,j}(u_j^{(k)}) - \theta_\ell = 0 ,$$

$u_j^{(k)}$  provenant de la résolution des sous-problèmes  $j \neq i$  à l'itération précédente ;

2. on ajoute dans le critère de chaque sous-problème associé à l'unité  $j \neq i$  un terme de la forme :

$$\langle p_\ell^{(k)}, \Theta_{\ell,j}(u_j) \rangle ,$$

$p_\ell^{(k)}$  étant le multiplicateur obtenu à l'itération précédente pour la contrainte du sous-problème  $i$ .

La prise en compte sur chacune des unités des modifications engendrées par le traitement de toutes les contraintes conduit à la formulation des sous-problèmes à résoudre dans la méthode de décomposition et coordination par prédiction. Il n'y a de plus aucune difficulté à étendre l'interprétation de nature mathématique à base de Lagrangiens et de maxi-minimisations emboîtées développée sur l'exemple ne comportant qu'une seule contrainte au cas général, et donc à obtenir encore des algorithmes de type point fixe et de type Arrow-Hurwicz.

**Remarque 9.** On consultera le polycopié (§2.1.4) pour une description complète du cas général. On notera dès à présent que les conditions de convergence de l'algorithme par prédiction de type point fixe font intervenir une condition *géométrique* portant sur la décomposition de l'espace d'arrivée des contraintes par rapport à la décomposition de l'espace de départ.

### 2.4.3 Application à un exemple

Pour fixer les idées, on illustre les étapes de la formulation des sous-problèmes lors de l'affectation des contraintes sur un exemple comportant trois unités et soumis à deux contraintes : <sup>14</sup>

$$\min_{(u_1, u_2, u_3)} J_1(u_1) + J_2(u_2) + J_3(u_3) ,$$

sous les contraintes :

$$\begin{aligned} \Theta_1(u_1) - \Theta_2(u_2) + \Theta_3(u_3) &\leq 0 , \\ \Omega_1(u_1) - \Omega_3(u_3) &= 0 . \end{aligned}$$

On décompose le problème en 3 sous-problèmes dépendant chacun d'une seule variable  $u_i$ , on note  $p_1$  (resp.  $p_2$ ) le multiplicateur associé à la première (resp. deuxième) contrainte, et *on choisit* :

- d'affecter la première contrainte au sous-problème 1 (portant sur la variable  $u_1$ ),
- d'affecter la deuxième contrainte au sous-problème 3 (portant sur la variable  $u_3$ ).

Le traitement des contraintes dans la méthode par prédiction conduit aux modifications suivantes.

1. La prise en compte de la première contrainte induit
  - une contrainte  $\Theta_1(u_1) - v^{(k)} \leq 0$  dans le sous-problème 1, avec  $v^{(k)} = \Theta_2(u_2^{(k)}) - \Theta_3(u_3^{(k)})$ ,
  - un terme additionnel  $-\langle p_1^{(k)}, \Theta_2(u_2) \rangle$  dans le critère du sous-problème 2,
  - un terme additionnel  $\langle p_1^{(k)}, \Theta_3(u_3) \rangle$  dans le critère du sous-problème 3.
2. La prise en compte de la deuxième contrainte induit
  - une contrainte  $-\Omega_3(u_3) - w^{(k)} = 0$  dans le sous-problème 3, avec  $w^{(k)} = -\Omega_1(u_1^{(k)})$ ,
  - un terme additionnel  $\langle p_2^{(k)}, \Omega_1(u_1) \rangle$  dans le critère du sous-problème 1.

---

14. La première contrainte est une contrainte d'inégalité, mais cela ne modifie qu'à la marge la mise en œuvre de l'algorithme par prédiction.

Le tableau 2.4.3 montre les transformations des sous-problèmes à chaque étape du processus d'affectation des contraintes.

Pour conclure, on peut faire sur cet exemple les observations suivantes.

- Les signes des termes additionnels dans les critères des sous-problèmes dépendent de la manière dont on écrit les contraintes. Il faut en particulier veiller à ce que la même écriture de la contrainte soit utilisée lors de la modification des différents sous-problèmes.
- Il n'existe pas dans cet exemple d'ordre de résolution des sous-problèmes conduisant à des solutions admissibles à chaque itération dans une version séquentielle de l'algorithme, car le sous-problème 1 fait intervenir la solution du sous-problème 3 *et réciproquement*.

On vérifiera que, si la première contrainte était affectée au sous-problème 2 portant sur la variable  $u_2$ , un ordre de résolution admissible en version séquentielle serait : 1 – 3 – 2.

<b>Avant affectation des contraintes</b>		
$\min_{u_1} J_1(u_1)$	$\min_{u_2} J_2(u_2)$	$\min_{u_3} J_3(u_3)$
<b>Après affectation de la première contrainte au sous-problème 1</b>		
$\min_{u_1} J_1(u_1)$ sous la contrainte $\Theta_1(u_1) - \Theta_2(u_2^{(k)}) + \Theta_3(u_3^{(k)}) \leq 0$	$\min_{u_2} J_2(u_2) - \langle p_1^{(k)}, \Theta_2(u_2) \rangle$	$\min_{u_3} J_3(u_3) + \langle p_1^{(k)}, \Theta_3(u_3) \rangle$
<b>Après affectation de la deuxième contrainte au sous-problème 3</b>		
$\min_{u_1} J_1(u_1) + \langle p_2^{(k)}, \Omega_1(u_1) \rangle$ sous la contrainte $\Theta_1(u_1) - \Theta_2(u_2^{(k)}) + \Theta_3(u_3^{(k)}) \leq 0$	$\min_{u_2} J_2(u_2) - \langle p_1^{(k)}, \Theta_2(u_2) \rangle$	$\min_{u_3} J_3(u_3) + \langle p_1^{(k)}, \Theta_3(u_3) \rangle$ sous la contrainte $\Omega_1(u_1^{(k)}) - \Omega_3(u_3) = 0$

TABLE 1 – Constitution des sous-problèmes dans l'exemple à 3 unités et 2 contraintes





## 2.5 Exercices

### Projection sur un hyperplan

On considère l'hyperplan  $\Sigma_\theta$  de  $\mathbb{R}^N$  défini par :

$$\Sigma_\theta = \left\{ u = (u_1, \dots, u_N) \in \mathbb{R}^N, \sum_{i=1}^N u_i = \theta \right\}.$$

- 1) Calculer la projection d'un point  $u^0 \in \mathbb{R}^N$  sur  $\Sigma_\theta$ .
- 2) Retrouver la formule (36) de mise à jour des  $v_i$  dans la méthode de décomposition par allocation.

### Décomposition appliquée à un réseau d'eau

On a vu lors de la première séance que le problème de l'optimisation de  $N$  sous-réseaux de distribution d'eau connectés par une usine de refoulement se mettait, après minimisation par rapport aux variables locales des sous-réseaux, sous la forme :

$$\min_{\{(u_{i,1}, u_{i,2})\}_{i=1}^{N+1}} \sum_{i=1}^{N+1} J_i(u_{i,1}, u_{i,2}), \quad (43a)$$

sous les contraintes :

$$\sum_{i=1}^N u_{i,1} - u_{N+1,1} = 0, \quad \sum_{i=1}^N u_{i,2} - u_{N+1,2} = 0, \quad (43b)$$

où l'on rappelle que, pour  $i$  variant de 1 à  $N$ , chaque fonction  $J_i$  est définie par une minimisation :

$$J_i(u_{i,1}, u_{i,2}) = \min_{(v_{i,1}, v_{i,2})} \left( \frac{1}{2} (a_{i,1} v_{i,1}^2 + a_{i,2} v_{i,2}^2) \text{ sous } \begin{cases} \bar{v}_{i,1} - u_{i,1} - v_{i,1} \leq 0 \\ u_{i,1} + u_{i,2} + v_{i,1} + v_{i,2} - \bar{v}_{i,2} = 0 \end{cases} \right), \quad (44)$$

alors que la fonction  $J_{N+1}$  correspond simplement au coût de l'usine de refoulement :

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

Pour que le problème ait un sens physique, on ajoute sur les variables  $u_{i,t}$  les contraintes suivantes :

$$0 \leq u_{i,1} \leq \bar{v}_{i,1} \quad \text{et} \quad 0 \leq u_{i,2} \leq \bar{v}_{i,2} - \bar{v}_{i,1} \quad \forall i = 1, \dots, N.$$

Ces contraintes assurent que les volumes fournis aux sous-réseaux par l'usine de refoulement sont positifs, et que ces volumes sont compatibles avec les contraintes locales de chacun des sous-réseaux. Il n'y a par contre pas de contraintes spécifiques sur les variables  $u_{N+1,1}$  et  $u_{N+1,2}$ .

### Décomposition par les prix

- 1) Écrire les sous-problèmes apparaissant dans une itération de l'algorithme de décomposition par les prix appliqué au problème (43), ainsi que l'étape de coordination associée..
- 2) Résoudre complètement le sous-problème associé aux variables  $u_{N+1,1}$  et  $u_{N+1,2}$ . Donner une interprétation économique de ce sous-problème, ainsi que des  $N$  autres sous problèmes.
- 3) Écrire le sous-problème d'optimisation associé au  $i$ -ème sous-réseau ( $1 \leq i \leq N$ ) en réintroduisant les variables  $(v_{i,1}, v_{i,2})$  et les contraintes locales de ce sous-réseau.
- 4) Discuter des particularités du sous-problème précédent.

**Décomposition par allocation de ressources**

Répondre aux mêmes questions que dans le cas de la décomposition par les prix.

**Décomposition par prédiction**

Affecter les contraintes couplantes au  $(N + 1)$ -ème sous-problème et répondre aux mêmes questions que précédemment.

## 2.6 Corrigés

### Projection sur un hyperplan particulier

Soit  $u^\sharp$  la projection du point  $u^0$  sur  $\Sigma_\theta$  :

$$u^\sharp = \text{proj}_{\Sigma_\theta}(u^0) .$$

Par définition de la projection, le point  $u^\sharp$  réalise le minimum de la distance entre le point  $u^0$  et l'ensemble  $\Sigma_\theta$ , et s'interprète donc comme la solution du problème d'optimisation :

$$\min_{u \in \mathbb{R}^N} \frac{1}{2} \sum_{i=1}^N (u_i - u_i^0)^2 ,$$

sous la contrainte

$$\sum_{i=1}^N u_i = \theta .$$

Ce problème a toutes les bonnes propriétés (critère quadratique défini positif, contrainte linéaire, ...). Notant  $p$  le multiplicateur (scalaire) associé à l'unique contrainte et  $p^\sharp$  sa valeur optimale, les conditions de Karush–Kuhn–Tucker associées s'écrivent :

- (i)  $u_i^\sharp - u_i^0 + p^\sharp = 0$ ,  $\forall i = 1, \dots, N$ ,
- (ii)  $\sum_{i=1}^N u_i^\sharp = \theta$ .

Additionnant les  $N$  premières conditions (i) et remplaçant dans le résultat la somme des  $u_i^\sharp$  par  $\theta$  (condition (ii)), on obtient :

$$p^\sharp = \frac{1}{N} \left( \sum_{i=1}^N u_i^0 - \theta \right) .$$

Reportant dans chacune des  $N$  premières conditions la valeur de  $p^\sharp$  ainsi obtenue, on obtient :

$$u_i^\sharp = u_i^0 - \frac{1}{N} \left( \sum_{j=1}^N u_j^0 - \theta \right) \quad \forall i = 1, \dots, N ,$$

ce qui fournit le résultat recherché.

**Application à la décomposition par allocation.** Dans la méthode de décomposition par allocation de ressources, l'étape de remise à jour de l'allocation consiste à effectuer un pas de gradient, dont le résultat est projeté sur la contrainte  $\Sigma_\theta$ . Le pas de gradient *avant* projection de la  $i$ -ème composante  $v_i^{(k)}$  de l'allocation fournit la valeur :

$$v_i^{(k)} + \rho p_i^{(k+1)} ,$$

car  $p_i^{(k+1)} = -\nabla \Phi_i(v_i^{(k)})$  (interprétation marginaliste des multiplicateurs). Utilisant le fait que le point  $(v_1^{(k)}, \dots, v_N^{(k)})$  appartient à l'hyperplan  $\Sigma_\theta$ , la formule de projection :

$$\begin{pmatrix} v_1^{(k+1)} \\ \vdots \\ v_N^{(k+1)} \end{pmatrix} = \text{proj}_{\Sigma_\theta} \begin{pmatrix} v_1^{(k)} + \rho p_1^{(k+1)} \\ \vdots \\ v_N^{(k)} + \rho p_N^{(k+1)} \end{pmatrix} ,$$

fournit exactement la relation (36) :

$$v_i^{(k+1)} = v_i^{(k)} + \rho \left( p_i^{(k+1)} - \frac{1}{N} \sum_{j=1}^N p_j^{(k+1)} \right) \quad \forall i = 1, \dots, N .$$

## Décomposition appliquée à un réseau d'eau

On note  $U_i^{\text{ad}}$  l'ensemble admissible auquel doit appartenir le vecteur  $(u_{i,1}, u_{i,2})$  pour  $i$  variant de 1 à  $N$  :

$$U_i^{\text{ad}} = \{(u_{i,1}, u_{i,2}) \in \mathbb{R}^2, \quad 0 \leq u_{i,1} \leq \bar{v}_{i,1} \text{ et } 0 \leq u_{i,2} \leq \bar{v}_{i,2} - \bar{v}_{i,1}\} .$$

Dans le cas où l'on définit les variables  $u_{N+1,1}$  et  $u_{N+1,2}$  par les équations (43b), il n'y a pas de contraintes spécifiques sur ces variables, de telle sorte que l'ensemble admissible  $U_{N+1}^{\text{ad}}$  est égal à tout l'espace  $\mathbb{R}^2$ .

### Forme des fonctions $J_i$ , pour $1 \leq i \leq N$

Supposant les débits  $u_{i,1}$  et  $u_{i,2}$  fixés, le problème de l'optimisation du  $i$ -ème sous-réseau s'écrit :

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

sous les contraintes

$$\bar{v}_{i,1} - v_{i,1} - u_{i,1} \leq 0 ,$$

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

Le Lagrangien obtenu en dualisant les contraintes est :

$$L(v_{i,1}, v_{i,2}, p_{i,1}, p_{i,2}) = \frac{1}{2} (a_{i,1}v_{i,1}^2 + a_{i,2}v_{i,2}^2) + p_{i,1}(\bar{v}_{i,1} - v_{i,1} - u_{i,1}) + p_{i,2}(v_{i,1} + v_{i,2} + u_{i,1} + u_{i,2} - \bar{v}_{i,2}) ,$$

et les conditions de Karush–Kuhn–Tucker prennent la forme :

$$a_{i,1}v_{i,1} - p_{i,1} + p_{i,2} = 0 ,$$

$$a_{i,2}v_{i,2} + p_{i,2} = 0 ,$$

$$\bar{v}_{i,1} - v_{i,1} - u_{i,1} \leq 0 \quad , \quad p_{i,1} \geq 0 ,$$

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

$$p_{i,1}(\bar{v}_{i,1} - v_{i,1} - u_{i,1}) = 0 .$$

On se livre alors au jeu habituel des devinettes sur la dernière condition (dite des écarts complémentaires).

- Si la contrainte inégalité n'est pas saturée ( $\bar{v}_{i,1} - v_{i,1} - u_{i,1} < 0$ ), la condition des écarts complémentaires implique  $p_{i,1} = 0$ . La résolution du système linéaire des conditions d'optimalité résultantes fournit la solution primale :

$$v_{i,1} = \frac{a_{i,2}}{a_{i,1} + a_{i,2}}(\bar{v}_{i,2} - u_{i,1} - u_{i,2}) \quad , \quad v_{i,2} = \frac{a_{i,1}}{a_{i,1} + a_{i,2}}(\bar{v}_{i,2} - u_{i,1} - u_{i,2}) ,$$

et le fait que la contrainte ne soit effectivement pas saturée conduit à la condition :

$$(a_{i,1} + a_{i,2})\bar{v}_{i,1} < a_{i,2}\bar{v}_{i,2} + a_{i,1}u_{i,1} - a_{i,2}u_{i,2} .$$

- Si la contrainte est saturée, on a  $v_{i,1} = \bar{v}_{i,1} - u_{i,1}$ , la solution primale est :

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

et dire que le multiplicateur  $p_{i,1}$  est positif ou nul équivaut à la condition :

$$(a_{i,1} + a_{i,2})\bar{v}_{i,1} \geq a_{i,2}\bar{v}_{i,2} + a_{i,1}u_{i,1} - a_{i,2}u_{i,2} .$$

Remplaçant alors dans la fonction coût du  $i$ -ème sous-problème les variables  $v_{i,1}$  et  $v_{i,2}$  par ces valeurs optimales, on obtient l'expression du coût  $J_i$  du  $i$ -ème sous-réseau :

$$J_i(u_{i,1}, u_{i,2}) = \begin{cases} \frac{a_{i,1}a_{i,2}}{(a_{i,1} + a_{i,2})} (u_{i,1} + u_{i,2} - \bar{v}_{i,2})^2 & \text{si } (a_{i,1} + a_{i,2})\bar{v}_{i,1} < a_{i,2}\bar{v}_{i,2} + a_{i,1}u_{i,1} - a_{i,2}u_{i,2} , \\ \frac{1}{2} (a_{i,1}(u_{i,1} - \bar{v}_{i,1})^2 + a_{i,2}(u_{i,2} - \bar{v}_{i,2} + \bar{v}_{i,1})^2) & \text{si } (a_{i,1} + a_{i,2})\bar{v}_{i,1} \geq a_{i,2}\bar{v}_{i,2} + a_{i,1}u_{i,1} - a_{i,2}u_{i,2} . \end{cases}$$

On notera que la fonction  $J_i$  n'est pas strictement convexe (terme  $(u_{i,1} + u_{i,2} - \bar{v}_{i,2})^2$  dans l'expression de  $J_i$ ). De manière intuitive, on voit que dans le cas où la contrainte inégalité du problème d'optimisation du  $i$ -ème sous-réseau n'est pas saturée, on peut diminuer  $u_{i,1}$  (jusqu'à saturer la contrainte) tout en augmentant  $u_{i,2}$  (pour continuer à respecter la contrainte égalité) sans aucune modification de coût : la solution du problème n'est donc pas unique.

### Forme de la fonction $J_{N+1}$

La fonction  $J_{N+1}$  est directement la fonction coût de l'usine permettant d'alimenter les  $N$  premiers sous-réseaux. Elle est donc de la forme :

$$J_{N+1}(u_{N+1,1}, u_{N+1,2}) = \frac{1}{2} (a_{N+1,1}u_{N+1,1}^2 + a_{N+1,2}u_{N+1,2}^2) ,$$

$a_{N+1,1}$  et  $a_{N+1,2}$  représentant les coûts unitaires de l'énergie pour cette usine,  $u_{N+1,1}$  et  $u_{N+1,2}$  étant les volumes globaux distribués par l'usine à l'ensemble des  $N$  premiers sous-réseaux pour chacun des pas de temps.

### Décomposition par les prix

1) Notant  $p_1$  et  $p_2$  les prix associés aux contraintes explicites du problème (43), le Lagrangien s'écrit :

$$L(u_{1,1}, u_{1,2}, \dots, u_{N+1,1}, u_{N+1,2}, p_1, p_2) = \sum_{i=1}^{N+1} (J_i(u_{i,1}, u_{i,2})) + p_1 \left( \sum_{i=1}^N u_{i,1} - u_{N+1,1} \right) + p_2 \left( \sum_{i=1}^N u_{i,2} - u_{N+1,2} \right) .$$

L'algorithme de décomposition par les prix consiste à minimiser ce Lagrangien par rapport à toutes les variables  $u_{i,t}$  en fixant les valeurs  $p_1$  et  $p_2$ , puis à remettre à jour les prix par un pas de gradient de longueur  $\rho$ . Initialisant l'algorithme avec des prix  $p_1^{(0)}$  et  $p_2^{(0)}$  quelconques (nuls, par exemple), l'étape de minimisation de la  $k$ -ème itération de l'algorithme se décompose en la minimisation de  $N + 1$  sous-problèmes indépendants :

$$\min_{(u_{i,1}, u_{i,2}) \in U_i^{\text{ad}}} J_i(u_{i,1}, u_{i,2}) + p_1^{(k)} u_{i,1} + p_2^{(k)} u_{i,2} \quad \forall i = 1 \dots N ,$$

$$\min_{(u_{N+1,1}, u_{N+1,2}) \in \mathbb{R}^2} J_{N+1}(u_{N+1,1}, u_{N+1,2}) - p_1^{(k)} u_{N+1,1} - p_2^{(k)} u_{N+1,2} .$$

Notant  $(u_{1,1}^{(k+1)}, u_{1,2}^{(k+1)}), \dots, (u_{N+1,1}^{(k+1)}, u_{N+1,2}^{(k+1)})$  les solutions de ces problèmes, l'étape de remise à jour des prix est :

$$p_1^{(k+1)} = p_1^{(k)} + \rho \left( \sum_{i=1}^N u_{i,1}^{(k+1)} - u_{N+1,1}^{(k+1)} \right) ,$$

$$p_2^{(k+1)} = p_2^{(k)} + \rho \left( \sum_{i=1}^N u_{i,2}^{(k+1)} - u_{N+1,2}^{(k+1)} \right) .$$

2) Dans le dernier sous-problème de minimisation, la forme de la fonction de coût  $J_{N+1}$  est explicite :

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

La solution de ce sous-problème est alors :

$$u_{N+1,1}^{(k+1)} = \frac{p_1^{(k)}}{a_{N+1,1}} , \quad u_{N+1,2}^{(k+1)} = \frac{p_2^{(k)}}{a_{N+1,2}} ,$$

et correspond à faire produire l'usine jusqu'à ce que son prix marginal au pas de temps  $t$  (soit  $a_{N+1,t}u_{N+1,t}^{(k+1)}$ ) soit égal au prix proposé par la coordination (soit  $p_t^{(k)}$ ).

Pour les  $N$  premiers sous-problèmes, la minimisation consiste à établir le meilleur compromis possible entre le coût de production de l'eau par l'usine local (terme  $J_i(u_{i,1}, u_{i,2})$ , résultant d'une autre minimisation par rapport aux variables locales  $v_{i,1}$  et  $v_{i,2}$ ) et le coût d'achat de l'eau à l'extérieur du sous-réseau (terme  $p_1^{(k)}u_{i,1} + p_2^{(k)}u_{i,2}$ ).

3) Pour  $i = 1, \dots, N$ , la forme détaillée du  $i$ -ème sous-problème de minimisation est obtenue en remplaçant  $J_i$  par sa définition (3), qui fait elle-même intervenir une minimisation. Fusionnant les deux opérations de minimisation, et écrivant explicitement les contraintes de l'ensemble  $U_i^{\text{ad}}$ , le sous-problème associé au  $i$ -ème sous-réseau s'écrit :

$$\min_{(v_{i,1}, v_{i,2}, u_{i,1}, u_{i,2}) \in \mathbb{R}^4} \frac{1}{2} (a_{i,1}v_{i,1}^2 + a_{i,2}v_{i,2}^2) + p_1^{(k)}u_{i,1} + p_2^{(k)}u_{i,2} , \quad (45a)$$

sous les contraintes

$$\bar{v}_{i,1} - v_{i,1} - u_{i,1} \leq 0 , \quad (45b)$$

$$v_{i,1} + v_{i,2} + u_{i,1} + u_{i,2} - \bar{v}_{i,2} = 0 , \quad (45c)$$

$$0 \leq u_{i,1} \leq \bar{v}_{i,1} , \quad (45d)$$

$$0 \leq u_{i,2} \leq \bar{v}_{i,2} - \bar{v}_{i,1} . \quad (45e)$$

4) La résolution du sous-problème (45) est sensiblement plus complexe que celle correspondant au cas du sous-réseau isolé (premier exercice de la première séance). Ainsi, si l'on résout directement les conditions de Karush–Kuhn–Tucker, on doit envisager 32 alternatives (5 contraintes de type inégalité).<sup>15</sup> On peut d'ailleurs envisager bien d'autres méthodes de résolution, allant de l'énumération exhaustive (en discrétisant les valeurs que peuvent prendre les variables  $v_{i,1}$ ,  $v_{i,2}$ ,  $u_{i,1}$  et  $u_{i,2}$ ) à l'algorithme de Uzawa (en formant le Lagrangien associé au sous-problème (45)) en passant par les méthodes de gradient avec projection sur les contraintes.

On notera que le problème (45) n'est a priori pas *strictement convexe*, car les variables  $u_{i,1}$  et  $u_{i,2}$  apparaissent *linéairement* dans la fonction coût à minimiser. Plus précisément, on a montré au début de l'exercice que le critère du problème (45) après minimisation en  $(v_{i,1}, v_{i,2})$  s'écrit :

$$\frac{a_{i,1}a_{i,2}}{(a_{i,1} + a_{i,2})} (u_{i,1} + u_{i,2} - \bar{v}_{i,2})^2 + p_1^{(k)} u_{i,1} + p_2^{(k)} u_{i,2} ,$$

pourvu que la solution  $(u_{i,1}, u_{i,2})$  vérifie l'inégalité :

$$a_{i,1}u_{i,1} - a_{i,2}u_{i,2} \geq (a_{i,1} + a_{i,2})\bar{v}_{i,1} - a_{i,2}\bar{v}_{i,2} .$$

Dans le cas  $p_1^{(k)} = p_2^{(k)}$ , le critère ne dépend que de la somme  $u_{i,1} + u_{i,2}$ , et le problème n'a pas forcément une *solution unique* : il peut alors y avoir un problème de *stabilité* du Lagrangien, et donc une impossibilité d'obtenir par cet algorithme une solution du problème initial.

Pour conclure, on remarquera que, dans le cas d'un sous-réseau particulier  $i_0$  complexe en lui-même (présence de plusieurs usines locales et de plusieurs réservoirs), la minimisation par rapport aux seules variables locales  $v_{i_0,t}$  peut déjà être une tâche difficile. Y ajouter la minimisation par rapport aux variables  $u_{i_0,t}$  (comme le fait la méthode de décomposition par les prix) conduit à un sous-problème (45) dont la résolution peut être trop consommatrice en temps-calcul pour pouvoir être insérée dans l'algorithme itératif de coordination. Pour illustrer ce propos, supposons que la minimisation en  $v_{i_0,t}$  uniquement conduit à un temps calcul de l'ordre de la minute ; optimiser en  $v_{i_0,t}$  et en  $u_{i_0,t}$  nécessite un temps de calcul plus important, de l'ordre de l'heure pour des méthodes de résolution basées sur l'énumération des solutions (programmation dynamique, par exemple) : le processus de coordination étant basé sur l'algorithme du gradient sur les prix, il demandera plusieurs dizaines d'itérations pour converger, soit quelques journées de temps de calcul pour obtenir la solution du problème global, ce qui est lourd, même pour un grand système.<sup>16</sup>

### Décomposition par allocation de ressources

1) On se donne des apports d'eau  $(w_{i,1}^{(0)}, w_{i,2}^{(0)})$ ,  $1 \leq i \leq N$ , pour les  $N$  premiers sous-réseaux, et on calcule pour le dernier sous-réseau les productions  $(w_{N+1,1}^{(0)}, w_{N+1,2}^{(0)})$  de telle sorte que l'on ait une allocation :

$$\sum_{i=1}^{N+1} w_{i,1}^{(0)} = 0 \quad , \quad \sum_{i=1}^{N+1} w_{i,2}^{(0)} = 0 .$$

La  $k$ -ème itération de l'algorithme de décomposition par allocation de ressources consiste à minimiser le critère  $J_i$  de chaque sous-réseau sous une contrainte d'allocation locale. Pour chacun des  $N$  premiers sous-réseaux, on doit ainsi résoudre :

$$\min_{(u_{i,1}, u_{i,2}) \in U_i^{\text{ad}}} J_i(u_{i,1}, u_{i,2}) \quad \text{sous} \quad u_{i,1} - w_{i,1}^{(k)} = 0 \quad \text{et} \quad u_{i,2} - w_{i,2}^{(k)} = 0 , \quad (46)$$

ce qui, outre la solution évidente :<sup>17</sup>

$$u_{i,1}^{(k+1)} = w_{i,1}^{(k)} \quad , \quad u_{i,2}^{(k+1)} = w_{i,2}^{(k)} ,$$

fournit les multiplicateurs optimaux associés aux contraintes, que l'on obtient en écrivant les conditions d'optimalité en  $u_{i,1}$  et en  $u_{i,2}$  du Lagrangien associé au problème (46) :

$$J_i(u_{i,1}, u_{i,2}) + p_{i,1} (u_{i,1} - w_{i,1}^{(k)}) + p_{i,2} (u_{i,2} - w_{i,2}^{(k)}) ,$$

15. En fait, il n'y a que 18 alternatives à traiter car les contraintes de bornes portant sur les variables  $u_{i,t}$  individuellement s'excluent mutuellement deux à deux.

16. On rappelle que la méthode décomposition par les prix ne fournit une solution admissible — c'est à dire vérifiant *toutes* les contraintes — qu'après avoir convergé, ce qui interdit d'interrompre l'algorithme en cours de calcul.

17. évidente dans la mesure où l'on ne s'intéresse qu'aux variables  $u_{i,1}$  et  $u_{i,2}$ , car la simple évaluation de la fonction  $J_i$  en un point est déjà un problème de minimisation par rapport aux variables locales  $v_{i,1}$  et  $v_{i,2}$

ce qui donne :

$$p_{i,1}^{(k+1)} = -\frac{\partial J_i}{\partial u_{i,1}}(u_{i,1}^{(k+1)}, u_{i,2}^{(k+1)}) \quad , \quad p_{i,2}^{(k+1)} = -\frac{\partial J_i}{\partial u_{i,2}}(u_{i,1}^{(k+1)}, u_{i,2}^{(k+1)}) .$$

Le sous-problème associé au dernier sous-réseau est :

$$\min_{(u_{N+1,1}, u_{N+1,2}) \in \mathbb{R}^2} J_{N+1}(u_{N+1,1}, u_{N+1,2}) \quad \text{sous} \quad -u_{N+1,1} - w_{N+1,1}^{(k)} = 0 \quad \text{et} \quad -u_{N+1,2} - w_{N+1,2}^{(k)} = 0 .$$

Le Lagrangien associé s'écrit :

$$J_{N+1}(u_{N+1,1}, u_{N+1,2}) - p_{N+1,1} \left( u_{N+1,1} + w_{N+1,1}^{(k)} \right) - p_{N+1,2} \left( u_{N+1,2} + w_{N+1,2}^{(k)} \right) ,$$

et sa solution complète est :

$$\begin{aligned} u_{N+1,1}^{(k+1)} &= -w_{N+1,1}^{(k)} \quad , \quad u_{N+1,2}^{(k+1)} = -w_{N+1,2}^{(k)} , \\ p_{N+1,1}^{(k+1)} &= \frac{\partial J_{N+1}}{\partial u_{N+1,1}}(u_{N+1,1}^{(k+1)}, u_{N+1,2}^{(k+1)}) \quad , \quad p_{N+1,2}^{(k+1)} = \frac{\partial J_{N+1}}{\partial u_{N+1,2}}(u_{N+1,1}^{(k+1)}, u_{N+1,2}^{(k+1)}) . \end{aligned}$$

On notera le changement de signe pour les multiplicateurs du dernier sous-problème par rapport aux multiplicateurs des  $N$  premiers, changement dû à la forme des contraintes du dernier sous-problème.

L'étape de remise à jour consiste à projeter un pas de gradient sur l'hyperplan  $\sum_{i=1}^{N+1} w_{i,t} = 0$  des allocations admissibles à chaque pas de temps  $t$ , ce qui, pour  $1 \leq i \leq N+1$ , fournit la nouvelle allocation :

$$w_{i,1}^{(k+1)} = w_{i,1}^{(k)} + \rho \left( p_{i,1}^{(k+1)} - \frac{1}{N+1} \sum_{j=1}^{N+1} p_{j,1}^{(k+1)} \right) \quad , \quad w_{i,2}^{(k+1)} = w_{i,2}^{(k)} + \rho \left( p_{i,2}^{(k+1)} - \frac{1}{N+1} \sum_{j=1}^{N+1} p_{j,2}^{(k+1)} \right) .$$

2) Comme on dispose dans le dernier sous-problème de la forme explicite de la fonction de coût  $J_{N+1}$ , on a directement l'expression de ce dernier :

$$\begin{aligned} \min_{(u_{N+1,1}, u_{N+1,2}) \in \mathbb{R}^2} \frac{1}{2} (a_{N+1,1} u_{N+1,1}^2 + a_{N+1,2} u_{N+1,2}^2) , \\ \text{sous les contraintes :} \end{aligned}$$

$$-u_{N+1,1} - w_{N+1,1}^{(k)} = 0 \quad \text{et} \quad -u_{N+1,2} - w_{N+1,2}^{(k)} = 0 ,$$

d'où l'on déduit les valeurs des multiplicateurs optimaux associés aux contraintes :

$$p_{N+1,1}^{(k+1)} = a_{N+1,1} u_{N+1,1}^{(k+1)} \quad , \quad p_{N+1,2}^{(k+1)} = a_{N+1,2} u_{N+1,2}^{(k+1)} .$$

3) Pour  $i = 1, \dots, N$ , la forme détaillée du  $i$ -ème sous-problème de minimisation est obtenue en remplaçant  $J_i$  par sa définition (44), et s'écrit :

$$\min_{(v_{i,1}, v_{i,2}, u_{i,1}, u_{i,2}) \in \mathbb{R}^4} \frac{1}{2} (a_{i,1} v_{i,1}^2 + a_{i,2} v_{i,2}^2) , \quad (47a)$$

sous les contraintes

$$\bar{v}_{i,1} - v_{i,1} - u_{i,1} \leq 0 , \quad (47b)$$

$$v_{i,1} + v_{i,2} + u_{i,1} + u_{i,2} - \bar{v}_{i,2} = 0 , \quad (47c)$$

$$u_{i,1} - w_{i,1}^{(k)} = 0 , \quad (47d)$$

$$u_{i,2} - w_{i,2}^{(k)} = 0 , \quad (47e)$$

$$0 \leq u_{i,1} \leq \bar{v}_{i,1} , \quad (47f)$$

$$0 \leq u_{i,2} \leq \bar{v}_{i,2} - \bar{v}_{i,1} . \quad (47g)$$

Pour ces  $N$  premiers sous-problèmes, la minimisation consiste donc à faire produire l'usine locale (variables  $v_{i,1}$  et  $v_{i,2}$ ) au moindre coût, les quantités d'eau  $u_{i,1}$  et  $u_{i,2}$  fournies par l'usine du dernier sous-réseau étant fixées à des valeurs devant respecter certaines contraintes.

4) La résolution du sous-problème (47) est de complexité équivalente à celle correspondant au cas du sous-réseau isolé (premier exercice de la première séance) *pourvu que l'ensemble des points*  $(v_{i,1}, v_{i,2}, u_{i,1}, u_{i,2})$  *satisfaisant les contraintes ne soit pas vide*. On doit envisager deux cas :

- $w_{i,1}^{(k)} \in [0, \bar{v}_{i,1}]$  **et**  $w_{i,2}^{(k)} \in [0, \bar{v}_{i,2} - \bar{v}_{i,1}]$  : l'ensemble des solutions n'est pas vide, les contraintes de bornes (47f) et (47g) sur les variables  $u_{i,1}$  et  $u_{i,2}$  sont toujours vérifiées et peuvent être ignorées ; notant  $\lambda_{i,1}$ ,  $\lambda_{i,2}$ ,  $p_{i,1}$  et  $p_{i,2}$  les multiplicateurs respectifs des contraintes (47b), (47c), (47d) et (47e), on obtient, en écrivant les conditions d'optimalité par rapport aux variables  $u_{i,1}$  et  $u_{i,2}$  les relations suivantes :

$$p_{i,1}^{(k+1)} = \lambda_{i,1}^{(k+1)} - \lambda_{i,2}^{(k+1)} \quad , \quad p_{i,2}^{(k+1)} = -\lambda_{i,2}^{(k+1)} .$$

Il suffit alors de minimiser le critère (47a) sous les contraintes (47b) et (47c) dans lesquelles on remplace les  $u_{i,t}$  par  $w_{i,t}^{(k)}$ , pour obtenir les multiplicateurs  $\lambda_{i,t}^{(k+1)}$ , et donc les multiplicateurs  $p_{i,t}^{(k+1)}$  utilisés par la coordination.

- $w_{i,1}^{(k)} \notin [0, \bar{v}_{i,1}]$  **ou**  $w_{i,2}^{(k)} \notin [0, \bar{v}_{i,2} - \bar{v}_{i,1}]$  : l'ensemble des solutions est vide, et il n'est alors pas possible de calculer les multiplicateurs  $p_{i,1}^{(k+1)}$  et  $p_{i,2}^{(k+1)}$ . On tombe sur l'inconvénient majeur de la méthode par allocation de ressources, qui peut conduire à formuler des sous-problèmes "trop contraints", ce qui bloque l'algorithme de décomposition et coordination.

Dans le cas de ce problème particulier, il existe cependant un moyen ad hoc de faire fonctionner l'algorithme de décomposition et coordination par allocation. Il suffit en effet de "déplacer" les contraintes (47f) et (47g), en les écrivant sous la forme équivalente  $0 \leq w_{i,1} \leq \bar{v}_{i,1}$  et  $0 \leq w_{i,2} \leq \bar{v}_{i,2} - \bar{v}_{i,1}$  (car  $u_{i,1} = w_{i,1}$  et  $u_{i,2} = w_{i,2}$ ), afin qu'elles soient traitées par le niveau de coordination et non lors de la résolution des sous-problèmes. Alors, ces résolutions ne posent plus de difficultés car les ensembles de points satisfaisant les contraintes ne sont plus jamais vides. Quant au niveau de coordination, il consiste à projeter un pas de gradient sur l'intersection des hyperplans d'équation  $\sum_{i=1}^{N+1} w_{i,t} = 0$  d'une part, et de l'intersection de tous les ensembles  $U_i^{\text{ad}}$  d'autre part. Ce type de projection complexe peut être réalisé en projetant successivement sur chacun des ensembles composant l'intersection. On doit alors mettre en œuvre un algorithme itératif de type point fixe, illustré Figure 8 pour le cas  $N = 1$  : partant d'un point  $w^{(0)}$  représentant le pas de gradient avant projection, on projette alternativement sur l'hyperplan  $\Sigma$  (points  $w^{(1)}, w^{(3)}, w^{(5)}, \dots$ ) et sur l'ensemble  $U_1^{\text{ad}}$  (points  $w^{(2)}, w^{(4)}, w^{(6)}, \dots$ ) jusqu'à converger vers le point  $w^\#$  appartenant à l'intersection  $\Sigma \cap U_1^{\text{ad}}$ .

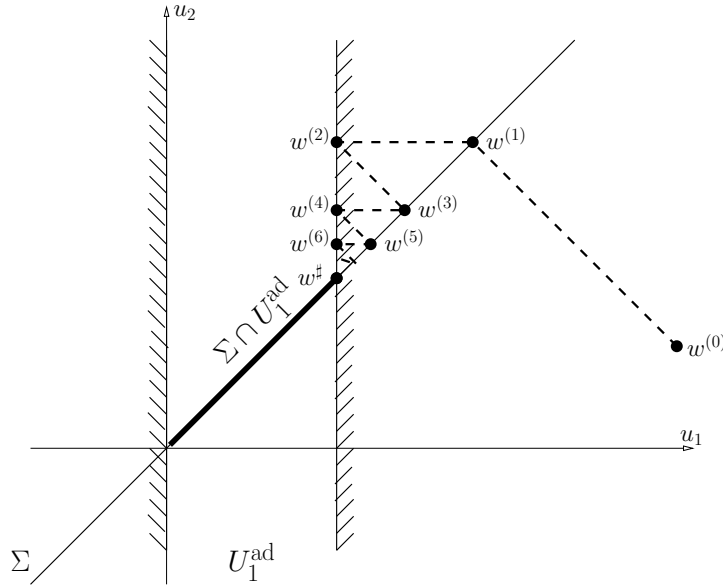


FIGURE 8 – Projection sur l'intersection de deux ensembles convexes

Cette solution est séduisante car, d'une part les résolutions des sous-réseaux restent simples (du même ordre de complexité que celle des sous-réseaux isolés), et d'autre part l'algorithme de projection est facile à mettre en œuvre et converge très rapidement. Elle est cependant adaptée au problème traité et n'est pas générale.

**Remarque 10.** Une autre solution respectant l'esprit de la décomposition par allocation de ressources (sinon sa lettre...) consiste à se donner une allocation  $(w_1, \dots, w_N)$  de taille  $N$  (et non de taille  $N + 1$  comme dans le cas exposé précédemment). Dans cette solution, pour chacun des  $N$  premiers sous-réseaux, on doit résoudre :

$$\min_{(u_{i,1}, u_{i,2}) \in U_i^{\text{ad}}} J_i(u_{i,1}, u_{i,2}) \quad \text{sous} \quad u_{i,1} - w_{i,1}^{(k)} = 0 \quad \text{et} \quad u_{i,2} - w_{i,2}^{(k)} = 0 , \quad (48)$$



le sous-problème associé au dernier sous-réseau étant alors :

$$\min_{(u_{N+1,1}, u_{N+1,2}) \in \mathbb{R}^2} J_{N+1}(u_{N+1,1}, u_{N+1,2}) \quad \text{sous} \quad u_{N+1,1} - \sum_{i=1}^N w_{i,1}^{(k)} = 0 \quad \text{et} \quad u_{N+1,2} - \sum_{i=1}^N w_{i,2}^{(k)} = 0 .$$

Il n'y a plus aucune contrainte portant sur les variables  $w_i$  et la méthode de gradient assurant leur remise à jour ne comporte donc plus d'opération de projection sur un hyperplan. Le calcul de gradients pour la remise à jour des variables  $w_i$  doit bien sûr être adapté car alors, les dérivées par rapport à chaque variable  $w_i$  sont la somme d'un terme provenant de la fonction  $J_i$  et d'un terme provenant de la fonction  $J_{N+1}$ .

Dans cette nouvelle mise en œuvre de la méthode par allocation, lorsqu'on déplace les contraintes portant sur les variables  $u_i$  sur les variables  $w_i$  afin de rendre les ensembles admissibles des sous-problèmes non vides, on n'a plus à considérer au niveau de la remise à jour des variables  $w_i$  que des contraintes de borne, pour lesquelles la projection se fait de manière explicite.

### Décomposition par prédiction

1) Dans la mesure où l'on a *choisi* de faire gérer les deux contraintes couplantes, à savoir :

$$\sum_{i=1}^N u_{i,1} - u_{N+1,1} = 0 \quad , \quad \sum_{i=1}^N u_{i,2} - u_{N+1,2} = 0 \quad ,$$

par le dernier sous-réseau, la minimisation associée à ce sous-réseau doit être effectuée sous contraintes (comme dans la méthode par allocation). Pour les  $N$  premiers sous-réseaux, on doit modifier le critère à minimiser afin de tenir compte de manière duale des contraintes (comme dans la méthode par les prix). On écrit ici l'algorithme dans la version classique (parallèle), ainsi que dans la version *séquentielle*, la résolution d'un sous-problème à la  $k$ -ème itération dans ce dernier cas utilisant toutes les informations déjà obtenues à cette itération.

- **Version parallèle** Initialisant l'algorithme avec des prix  $(p_{N+1,1}^{(0)}, p_{N+1,2}^{(0)})$  et avec des volumes  $(u_{i,1}^{(0)}, u_{i,2}^{(0)})$ ,  $1 \leq i \leq N$ , on résout à la  $k$ -ème itération de l'algorithme par prédiction le sous-problème associé à chacun des  $N$  premiers sous-réseaux, à savoir :

$$\min_{(u_{i,1}, u_{i,2}) \in U_i^{\text{ad}}} J_i(u_{i,1}, u_{i,2}) + p_{N+1,1}^{(k)} u_{i,1} + p_{N+1,2}^{(k)} u_{i,2} \quad \forall i = 1 \dots N \quad ,$$

problème dont la solution est notée  $(u_{i,1}^{(k+1)}, u_{i,2}^{(k+1)})$ .

On résout aussi le problème sous-contrainte associé au dernier sous-réseau :

$$\min_{(u_{N+1,1}, u_{N+1,2}) \in \mathbb{R}^2} J_{N+1}(u_{N+1,1}, u_{N+1,2}) \quad \text{sous} \quad \sum_{i=1}^N u_{i,1}^{(k)} - u_{N+1,1} = 0 \quad \text{et} \quad \sum_{i=1}^N u_{i,2}^{(k)} - u_{N+1,2} = 0 \quad ,$$

et on note  $(p_{N+1,1}^{(k+1)}, p_{N+1,2}^{(k+1)})$  les multiplicateurs optimaux associés aux contraintes.

Dans cette version de l'algorithme, tous les sous-problèmes peuvent être résolus simultanément.

- **Version séquentielle** Initialisant l'algorithme avec des prix  $(p_{N+1,1}^{(0)}, p_{N+1,2}^{(0)})$ , on résout à la  $k$ -ème itération de l'algorithme par prédiction le sous-problème associé à chacun des  $N$  premiers sous-réseaux, à savoir :

$$\min_{(u_{i,1}, u_{i,2}) \in U_i^{\text{ad}}} J_i(u_{i,1}, u_{i,2}) + p_{N+1,1}^{(k)} u_{i,1} + p_{N+1,2}^{(k)} u_{i,2} \quad \forall i = 1 \dots N \quad ,$$

problème dont la solution est notée  $(u_{i,1}^{(k+1)}, u_{i,2}^{(k+1)})$  (on notera qu'il n'existe aucune influence entre ces  $N$  sous-problèmes, qui peuvent donc être résolus en *parallèle* bien que l'on décrive ici une version *séquentielle* de l'algorithme).

On résout ensuite le problème sous-contrainte associé au dernier sous-réseau en utilisant les dernières valeurs lors des résolutions précédentes :

$$\min_{(u_{N+1,1}, u_{N+1,2}) \in \mathbb{R}^2} J_{N+1}(u_{N+1,1}, u_{N+1,2}) \quad \text{sous} \quad \sum_{i=1}^N u_{i,1}^{(k+1)} - u_{N+1,1} = 0 \quad \text{et} \quad \sum_{i=1}^N u_{i,2}^{(k+1)} - u_{N+1,2} = 0 \quad ,$$

et on note  $(p_{N+1,1}^{(k+1)}, p_{N+1,2}^{(k+1)})$  les multiplicateurs optimaux associés aux contraintes.

Dans cette version de l'algorithme, le dernier sous-problème doit être résolu après les  $N$  premiers sous-problèmes.

La version séquentielle, conduisant à des solutions admissibles (c'est à dire vérifiant *toutes* les contraintes du problème initial) à chaque itération, et étant "presque autant parallélisable" que la version parallèle, sera seule considérée dans la suite.

2) Utilisant la forme explicite de la fonction de coût  $J_{N+1}$ , on obtient directement l'expression des multiplicateurs issus de la résolution du dernier sous-problème :

$$p_{N+1,1}^{(k+1)} = a_{N+1,1} \left( \sum_{i=1}^N u_{i,1}^{(k+1)} \right) , \quad p_{N+1,2}^{(k+1)} = a_{N+1,2} \left( \sum_{i=1}^N u_{i,2}^{(k+1)} \right) .$$

Ce sous-réseau est résolu comme dans la méthode par allocation de ressources, ce qui ne pose pas de difficultés particulières car il n'est soumis à aucune autre contrainte que celles issues de l'allocation, de telle sorte que l'ensemble satisfaisant les contraintes n'est jamais vide.

3) Pour chaque  $i = 1, \dots, N$ , la forme détaillée du  $i$ -ème sous-problème de minimisation est la même que celle obtenue dans la méthode par les prix, et est donc décrite par les équations (45). Pour ces  $N$  premiers sous-problèmes. la minimisation consiste donc à établir le meilleur compromis possible entre le coût de production de l'eau par l'usine local (terme  $J_i(u_{i,1}, u_{i,2})$ , résultant d'une autre minimisation par rapport aux variables locales  $v_{i,1}$  et  $v_{i,2}$ ) et le coût d'achat de l'eau à l'extérieur du sous-réseau (terme  $p_{N+1,1}^{(k)} u_{i,1} + p_{N+1,2}^{(k)} u_{i,2}$ ).

4) La restriction faite lors de la discussion à propos de la méthode par les prix concernant la lourdeur éventuelle des résolutions locales reste valable, mais on peut maintenant interrompre l'algorithme en cours de calcul car, dans ce cas, la solution obtenue à la fin de chaque itération de la méthode par prédiction est admissible. Dans cette application spécifique, la méthode par prédiction s'interprète comme une "méthode par les prix admissible" ; ceci n'est pas général et est dû à la forme très particulière du dernier sous-problème.

Supposons pour conclure que le problème d'optimisation associé au sous-réseau numéro  $i_0$  soit déjà complexe (sous-réseau comportant de nombreuses usines et réservoirs locaux), de telle sorte qu'il ne soit pas possible de le résoudre comme dans la méthode par les prix. Comment faire pour modifier la méthode par prédiction afin que ce sous-réseau y soit traité comme dans la méthode par allocation ?

Pour répondre à cette question, on va transformer le problème initial en y introduisant de nouvelles variables  $\nu_{i,t}$  pour  $i$  variant de 1 à  $N$ , qui vont représenter les variables  $u_{i,t}$  dans le dernier sous-problème (technique connue en optimisation sous le nom de "dédoublment de variables"). Le problème global ainsi reformulé s'écrit :

$$\min_{(u_i, \nu_i)} \sum_{i=1}^N J_i(u_i) + J_{N+1} \left( \sum_{i=1}^N \nu_i \right) . \quad (49a)$$

sous les contraintes :

$$\forall i = 1, \dots, N , \quad u_i - \nu_i = 0 \text{ et } u_i \in U_i^{\text{ad}} , \quad (49b)$$

où, pour simplifier l'écriture, les variables  $u_i$  et  $\nu_i$  représentent respectivement les vecteurs  $(u_{i,1}, u_{i,2})$  et  $(\nu_{i,1}, \nu_{i,2})$ . C'est sur cette nouvelle formule que l'on met en œuvre la méthode de prédiction, en affectant toutes les contraintes au dernier sous-problème, sauf les deux contraintes  $u_{i_0} - \nu_{i_0} = 0$  et  $u_{i_0} \in U_{i_0}^{\text{ad}}$  que l'on affecte au  $i_0$ -ème sous-problème. La  $k$ -ème itération de l'algorithme (en version séquentielle) est alors :

— *Sous-problème  $i$  ( $i \neq i_0$  et  $i \neq N + 1$ ) :*

$$\min_{u_i \in U_i^{\text{ad}}} J_i(u_i) + p_i^{(k)} u_i ,$$

La résolution de ces sous-problèmes fournit les valeurs  $u_i^{(k+1)}$ .

— *Sous-problème  $N + 1$  :*

$$\min_{\nu_i \in \mathbb{R}^2} J_{N+1} \left( \sum_{i=1}^N \nu_i \right) - p_{i_0}^{(k)} \nu_{i_0} \quad \text{sous} \quad u_i^{(k+1)} - \nu_i = 0 \quad \forall i \neq i_0 .$$

La résolution de ce sous-problème fournit la valeur  $\nu_{i_0}^{(k+1)}$  ainsi que les valeurs  $p_i^{(k+1)}$  des multiplicateurs.

— *Sous-problème  $i_0$  :*

$$\min_{u_{i_0} \in U_{i_0}^{\text{ad}}} J_{i_0}(u_{i_0}) \quad \text{sous} \quad u_{i_0} - \nu_{i_0}^{(k+1)} = 0 .$$

La résolution de ce sous-problème fournit la valeur  $p_{i_0}^{(k+1)}$  du multiplicateur.

On remarque tout d'abord que l'ordonnement des sous-problèmes est tel que la solution obtenue à chaque itération est admissible. On constate ensuite, en écrivant les conditions d'optimalité du sous-problème  $N + 1$ , que toutes les valeurs  $p_i^{(k+1)}$  ( $i \neq i_0$ ) sont égales entre elles (ceci est dû au fait que  $J_{N+1}$  ne dépend que de la somme des  $\nu_i$ ). La résolution est donc très semblable à la précédente, car les sous-problèmes pour  $i \neq i_0$  optimisent leur variable  $u_i$  en fonction d'un prix unique calculé par le sous-problème  $N + 1$ . L'optimisation du sous-problème  $i_0$  se fait à  $u_{i_0}$  fixé, ce qui était le but de l'opération. Quant à la résolution du sous-problème  $N + 1$ , elle se fait en ayant fixé toutes les variables  $\nu_i$  sauf  $\nu_{i_0}$  et correspond donc à la minimisation en  $\nu_{i_0}$ . On peut remarquer pour finir qu'il est possible de ne pas prendre en compte les contraintes  $u_{i_0} \in U_{i_0}^{\text{ad}}$  dans le sous-problème  $i_0$  pourvu que l'on ajoute dans le sous-problème  $N + 1$  les contraintes  $\nu_{i_0} \in U_{i_0}^{\text{ad}}$  (puisque  $u_{i_0} = \nu_{i_0}$ ), ce qui supprime tout problème d'admissibilité dans le sous-problème  $i_0$ .

D'un point de vue pratique, cette formulation obtenue par dédoublement de variables et méthode de prédiction est pertinente<sup>18</sup> car un sous-réseau  $i_0$  complexe est en général un sous-réseau pour le contrôle duquel on dispose de beaucoup de flexibilité. La plage de variation des variables  $u_{i_0,t}$  que peut supporter un tel sous-réseau est donc en général grande, de telle sorte qu'il y a peu de risques d'imposer à ces variables des valeurs incompatibles avec l'ensemble des contraintes.

---

18. même sans tenir compte de la toute dernière remarque concernant le "déplacement" des contraintes  $U_{i_0}^{\text{ad}}$



# 3 Principe du problème auxiliaire et applications

## 3.1 Introduction

On a étudié jusqu'ici un problème d'optimisation de la forme :

$$\min_{u \in U^{\text{ad}}} J(u), \quad (50a)$$

sous les contraintes :

$$\Theta(u) \in -C. \quad (50b)$$

On a supposé que ce problème était de grande taille, et on a présenté trois familles de méthodes de décomposition et coordination (prix, allocation et prédiction) permettant de le résoudre. La mise en œuvre de ces méthodes reposait sur les hypothèses, apparemment essentielles, suivantes.

1. Il existe une décomposition de l'espace  $\mathcal{U}$  telle que l'ensemble admissible  $U^{\text{ad}}$  se mette sous la forme d'un produit cartésien  $U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}$  ;

$$u = (u_1, \dots, u_N) \in U^{\text{ad}} \iff u_i \in U_i^{\text{ad}} \quad \forall i = 1, \dots, N.$$

2. La fonction de coût  $J$  est additive par rapport à cette décomposition :

$$J(u) = \sum_{i=1}^N J_i(u_i).$$

3. La fonctionnelle des contraintes  $\Theta$  est additive par rapport à cette décomposition :

$$\Theta(u) = \sum_{i=1}^N \Theta_i(u_i).$$

**Remarque 11.** Dans la méthode de décomposition par prédiction, on a vu qu'il fallait de plus introduire une décomposition de l'espace  $\mathcal{C}$  d'arrivée des contraintes  $\Theta$ , compatible avec celle de l'espace  $\mathcal{U}$ .

Notre but est maintenant de relâcher ces hypothèses. En fait, on gardera l'hypothèse concernant la forme "produit cartésien" de l'ensemble admissible  $U^{\text{ad}}$  qui n'est pas contraignante car il suffit de mettre toutes les contraintes couplantes dans la fonction  $\Theta$  plutôt que dans l'ensemble  $U^{\text{ad}}$ . Par contre, on veut faire disparaître les hypothèses d'additivité sur  $J$  et  $\Theta$  car :

- beaucoup de problèmes ne sont pas additifs (même s'ils le sont souvent "presque") ;
- certaines méthodes de résolution (comme le Lagrangien augmenté) introduisent des couplages.

Pour cela, on va introduire un principe général en optimisation appelé *principe du problème auxiliaire* (**PPA**), qui fournit un cadre unifié permettant de *construire* des algorithmes d'optimisation, avec les avantages suivants :

- le **PPA** permet en décomposition/coordination de s'affranchir des conditions d'additivité ;
- le **PPA** permet d'améliorer le conditionnement des problèmes ;
- le **PPA** permet de simplifier l'étude de la convergence des algorithmes.

On va commencer par exposer ce principe dans le cas sans contrainte explicite. On traitera dans une deuxième partie le cas général.

## 3.2 Optimisation sur un ensemble admissible

On s'intéresse à un problème d'optimisation sans contrainte explicite, c'est à dire dans lequel un ensemble  $U^{\text{ad}}$  rassemble toutes les contraintes d'admissibilité :

$$\min_{u \in U^{\text{ad}}} J(u) + J^\Sigma(u) . \quad (51)$$

On suppose que l'espace  $\mathcal{U}$  se décompose en un produit de  $N$  sous-espaces  $\mathcal{U}_i$  tels que l'ensemble  $U^{\text{ad}}$  se mette sous la forme  $U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}$ . L'écriture de la fonction coût comme la somme des deux termes  $J$  et  $J^\Sigma$  correspond au fait que l'on considère que c'est la partie  $J$  qui rend le problème complexe, alors que la partie  $J^\Sigma$  ne pose pas de difficultés. Ainsi, chaque fois qu'il sera question de décomposer ce problème, on supposera que la fonction  $J^\Sigma$  est additive par rapport à la décomposition de l'espace  $\mathcal{U}$  :<sup>19</sup>

$$J^\Sigma(u_1, \dots, u_N) = \sum_{i=1}^N J_i^\Sigma(u_i) .$$

On rappelle que, pour résoudre le problème (51), on a besoin des hypothèses "minimales" suivantes :

1.  $U^{\text{ad}}$  est un convexe fermé de  $\mathcal{U}$  ;
2.  $J$  et  $J^\Sigma$  sont des fonctions convexes, s.c.i. propres ;
3.  $J + J^\Sigma$  est coercive ;
4.  $J$  est Gâteaux-différentiable.

Les trois premières hypothèses sont suffisantes pour assurer l'existence de solutions au problème (51). La dernière hypothèse permet de caractériser une solution  $u^\sharp$  de (51) par l'inéquation variationnelle :

$$\forall u \in U^{\text{ad}} , \quad \langle \nabla J(u^\sharp) , u - u^\sharp \rangle + J^\Sigma(u) - J^\Sigma(u^\sharp) \geq 0 . \quad (52)$$

### 3.2.1 Présentation de la méthode

Quand on veut décomposer le problème (51), seul le terme  $J(u)$  est gênant car il n'est a priori pas additif par rapport à la décomposition de l'espace  $\mathcal{U}$  dont on dispose.

**La première idée** pour surmonter cette difficulté est de remplacer ce terme par son approximation au premier ordre autour d'un point  $u^{(k)}$  :

$$J(u) \approx J(u^{(k)}) + \langle \nabla J(u^{(k)}) , u - u^{(k)} \rangle .$$

En fait, le terme  $J(u^{(k)}) - \langle \nabla J(u^{(k)}) , u^{(k)} \rangle$  étant constant, cette approximation revient à remplacer dans (51) le terme  $J(u)$  par le terme  $\langle \nabla J(u^{(k)}) , u \rangle$  qui est *linéaire* en  $u$ . On constate alors que :

- ce nouveau terme, comme toute expression linéaire, est additif par rapport à n'importe quelle décomposition de l'espace  $\mathcal{U}$ ,
- la condition d'optimalité du problème *après approximation* :  $\min_{u \in U^{\text{ad}}} \langle \nabla J(u^{(k)}) , u \rangle + J^\Sigma(u)$ , est exactement la condition (52) dans le cas  $u^{(k)} = u^\sharp$ .

On utilise ainsi la *différentiabilité* de  $J$  pour remplacer l'hypothèse d'*additivité* que l'on faisait jusqu'alors sur cette fonction. Cependant, cette linéarisation risque de faire disparaître le caractère *coercif* de la fonction coût : il faut donc introduire un nouvel ingrédient dans le problème...

19. On notera en particulier que, en l'absence de terme  $J$ , le problème (51) se décompose en  $N$  sous-problèmes totalement indépendants dont la  $i$ -ème instance est :  $\min_{u_i \in U_i^{\text{ad}}} J_i^\Sigma(u_i)$ .

**La deuxième idée** consiste alors à ajouter dans le critère un terme *fortement convexe*, et donc coercif. Afin de ne pas perturber les conditions d'optimalité, on veut que le développement au premier ordre de ce nouveau terme soit nul au point  $u^{(k)}$ . Pour cela, on choisit une fonction  $K^{(k)}$ , appelée *noyau de décomposition*, que l'on suppose fortement convexe et différentiable, et on ajoute au critère (dans lequel le terme  $J$  a déjà été linéarisé) le terme :

$$\frac{1}{\epsilon^{(k)}} \left( K^{(k)}(u) - K^{(k)}(u^{(k)}) - \langle \nabla K^{(k)}(u^{(k)}), u - u^{(k)} \rangle \right),$$

où la constante multiplicative  $1/\epsilon^{(k)}$ , avec  $\epsilon^{(k)} > 0$ , permet de jouer sur le module de forte convexité de la fonction  $K^{(k)}$ . In fine, après suppression des termes constants et normalisation par  $\epsilon^{(k)}$ , le problème auquel on parvient est le suivant :

$$\min_{u \in U^{\text{ad}}} K^{(k)}(u) + \langle \epsilon^{(k)} \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}), u \rangle + \epsilon^{(k)} J^\Sigma(u). \quad (53)$$

Ce problème est appelé *problème auxiliaire au point*  $u^{(k)}$  associé au problème (51). Sa solution<sup>20</sup> est notée  $u^{(k+1)}$ , et les conditions d'optimalité associées s'écrivent :

$$\forall u \in U^{\text{ad}}, \langle \nabla K^{(k)}(u^{(k+1)}) + \epsilon^{(k)} \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}), u - u^{(k+1)} \rangle + \epsilon^{(k)} (J^\Sigma(u) - J^\Sigma(u^{(k+1)})) \geq 0. \quad (54)$$

Partant d'un point quelconque  $u^{(0)} \in U^{\text{ad}}$ , on construit ainsi la suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$  engendrée par l'enchaînement des résolutions des problèmes auxiliaires (53). Supposons que cette suite converge vers un point  $\bar{u}$ . Alors, la solution du problème auxiliaire au point  $\bar{u}$  est le point  $\bar{u}$  lui-même, et les conditions d'optimalité (54) se simplifient en :

$$\forall u \in U^{\text{ad}}, \epsilon^{(k)} (\langle \nabla J(\bar{u}), u - \bar{u} \rangle + J^\Sigma(u) - J^\Sigma(\bar{u})) \geq 0.$$

On retrouve au facteur  $\epsilon^{(k)}$  près les conditions d'optimalité (52), d'où l'on déduit que  $\bar{u}$  est une solution du problème initial (51). Ceci suggère, pour résoudre (51), la mise en œuvre de l'algorithme suivant.

### Algorithme 1.

0. Choisir une suite de fonctions  $\{K^{(k)}\}_{k \in \mathbb{N}}$  et une suite de coefficients  $\{\epsilon^{(k)}\}_{k \in \mathbb{N}}$ .
1. Choisir un point  $u^{(0)} \in U^{\text{ad}}$  et un seuil de convergence  $\sigma$ ; poser  $k = 0$ .
2. Calculer  $u^{(k+1)} = \arg \min_{u \in U^{\text{ad}}} \{K^{(k)}(u) + \langle \epsilon^{(k)} \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}), u \rangle + \epsilon^{(k)} J^\Sigma(u)\}$ .
3. Retourner à l'étape 2 en incrémentant  $k$  de 1 tant que  $\|u^{(k+1)} - u^{(k)}\| > \sigma$ .

Il peut paraître paradoxal de remplacer la résolution du problème (51) par la résolution d'une suite de problèmes apparemment de même nature. Ceci n'a de sens que si la résolution du problème auxiliaire (53) s'avère plus facile que celle du problème initial, pour des raisons liées :

- au nombre d'opérations mises en jeu par la résolution,
- au conditionnement numérique du problème auxiliaire,
- à la possibilité de le décomposer.

Le fait que les problèmes auxiliaires soient plus ou moins faciles à traiter provient directement du fait que l'on choisit les noyaux de décomposition  $K^{(k)}$  : c'est dans la pertinence de ce choix que se révélera au cas par cas l'intérêt du principe du problème auxiliaire.

20. unique car on a supposé que la fonction  $K^{(k)}$  est fortement convexe

### 3.2.2 Théorème de convergence

#### Théorème 3.

On suppose que les hypothèses suivantes sont vérifiées :

- H1** l'ensemble  $U^{\text{ad}}$  est un convexe fermé non vide de  $\mathcal{U}$ , espace de Hilbert de dimension finie ;
- H2** la fonction  $J$  est convexe, s.c.i., propre, Gâteaux-différentiable, et sa différentielle  $J'$  est lipschitzienne de constante  $A$  ;
- H3** la fonction  $J^\Sigma$  est convexe, s.c.i., propre ;
- H4** la fonction  $(J + J^\Sigma)$  est coercive sur  $U^{\text{ad}}$  ;
- H5** les noyaux de décomposition  $K^{(k)}$  sont fortement convexes de module  $b^{(k)}$ , s.c.i., propres, Gâteaux-différentiables de différentielles  $K^{(k)'}$  lipschitziennes de constante  $B^{(k)}$ , et l'on a :

$$\exists b > 0, \forall k \in \mathbb{N}, b^{(k)} \geq b \quad \text{et} \quad \exists B > 0, \forall k \in \mathbb{N}, B^{(k)} \leq B ;$$

- H6** les coefficients  $\epsilon^{(k)}$  vérifient les relations :

$$\exists \alpha > 0, \exists \beta > 0, \forall k \in \mathbb{N}, \alpha \leq \epsilon^{(k)} \leq \frac{2b^{(k)}}{A + \beta} .$$

Alors, on a les résultats suivants :

- R1** le problème initial (51) admet au moins une solution  $u^\sharp$ , et chaque problème auxiliaire (53) admet une unique solution  $u^{(k+1)}$  ;
- R2** la suite  $\{J(u^{(k)}) + J^\Sigma(u^{(k)})\}_{k \in \mathbb{N}}$  converge vers la valeur  $J(u^\sharp) + J^\Sigma(u^\sharp)$ , et cette suite est strictement décroissante jusqu'à la convergence ;
- R3** la suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$  est bornée, et tout point d'accumulation de cette suite est solution de (51).

Si on fait l'hypothèse supplémentaire :

- H7** la fonction  $J$  est fortement convexe de module  $a$ ,

alors on obtient de plus que :

- R4** la suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$  converge vers l'unique solution  $u^\sharp$  du problème (51) ;
- R5** on dispose de la majoration a posteriori d'erreur :

$$\|u^{(k+1)} - u^\sharp\| \leq \frac{A + B^{(k)}/\epsilon^{(k)}}{a} \|u^{(k+1)} - u^{(k)}\| .$$

La preuve de ce théorème va suivre le schéma suivant, fréquemment rencontré dans l'étude de la stabilité d'un algorithme itératif.

1. On choisit une fonction  $\mathcal{L}$ , appelée *fonction de Lyapounov*, opérant sur les variables  $u^{(k)}$  et telle que  $\mathcal{L}$  soit bornée inférieurement et coercive.
2. On montre que la suite  $\{\mathcal{L}(u^{(k)})\}_{k \in \mathbb{N}}$  décroît au cours des itérations de l'algorithme, ce qui implique que cette suite converge et que la suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$  est bornée.
3. On cherche à calculer la limite de la suite  $\{\mathcal{L}(u^{(k)})\}_{k \in \mathbb{N}}$  en fonction de la valeur du minimum du problème initial.
4. On extrait une sous-suite convergente de la suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$  et on cherche à caractériser sa limite.



*Démonstration.* L'ensemble  $U^{\text{ad}}$  étant un fermé d'un espace de dimension finie et la fonction  $(J + J^\Sigma)$  étant s.c.i. coercive sur cet ensemble, le problème (51) admet au moins une solution  $u^\sharp$ . La fonction coût du problème auxiliaire (53) est la somme de fonctions s.c.i. convexes et est donc elle-même s.c.i. convexe. Comme sa composante  $K^{(k)}$  est fortement convexe, elle est elle aussi fortement convexe et donc coercive. Enfin, l'ensemble  $U^{\text{ad}}$  est convexe. Le problème auxiliaire (53) admet donc une solution  $u^{(k+1)}$  unique, ce qui achève la démonstration du résultat **R1**.

On suit alors le schéma général de preuve en quatre étapes présenté à la page précédente.

1. On effectue le choix de fonction de Lyapounov  $\mathcal{L}$  suivant :

$$\mathcal{L} = J + J^\Sigma,$$

2. La suite  $\{\mathcal{L}(u^{(k)})\}_{k \in \mathbb{N}}$  est minorée par la valeur  $J(u^\sharp) + J^\Sigma(u^\sharp)$  qui correspond au minimum sur  $U^{\text{ad}}$ . Montrons que cette suite est décroissante pour prouver qu'elle converge.
  - Évaluant les conditions d'optimalité (54) du  $k$ -ème problème auxiliaire au point  $u = u^{(k)}$ , on obtient :

$$J^\Sigma(u^{(k)}) - J^\Sigma(u^{(k+1)}) \geq \frac{1}{\epsilon^{(k)}} \left\langle \nabla K^{(k)}(u^{(k)}) - \nabla K^{(k)}(u^{(k+1)}) - \epsilon^{(k)} \nabla J(u^{(k)}), u^{(k)} - u^{(k+1)} \right\rangle. \quad (55)$$

- Le fait que  $J'$  soit lipschitzienne de constante  $A$  s'écrit (voir Annexe 3.5 du polycopié) :

$$J(v) - J(u) \leq \langle \nabla J(u), v - u \rangle + \frac{A}{2} \|v - u\|^2.$$

Évaluer cette expression en  $u = u^{(k)}$  et  $v = u^{(k+1)}$  conduit à :

$$J(u^{(k)}) - J(u^{(k+1)}) \geq \left\langle \nabla J(u^{(k)}), u^{(k)} - u^{(k+1)} \right\rangle - \frac{A}{2} \|u^{(k)} - u^{(k+1)}\|^2. \quad (56)$$

Le noyau  $K^{(k)}$  étant fortement convexe de module  $b^{(k)}$  et différentiable, sa différentielle  $K^{(k) \prime}$  est fortement monotone (voir Annexe 3.4 du polycopié) et donc :

$$\left\langle \nabla K^{(k)}(u^{(k)}) - \nabla K^{(k)}(u^{(k+1)}), u^{(k)} - u^{(k+1)} \right\rangle \geq b^{(k)} \|u^{(k)} - u^{(k+1)}\|^2.$$

Additionnant les expressions (55) et (56) et utilisant la forte monotonie de  $K^{(k) \prime}$ , on obtient :

$$\begin{aligned} (J + J^\Sigma)(u^{(k)}) - (J + J^\Sigma)(u^{(k+1)}) &\geq \left( \frac{b^{(k)}}{\epsilon^{(k)}} - \frac{A}{2} \right) \|u^{(k)} - u^{(k+1)}\|^2 \\ &\geq \frac{\beta}{2} \|u^{(k)} - u^{(k+1)}\|^2, \end{aligned}$$

d'après l'hypothèse **H6**. On en déduit que la suite  $\{\mathcal{L}(u^{(k)})\}_{k \in \mathbb{N}}$  est strictement décroissante tant que l'on a  $u^{(k)} \neq u^{(k+1)}$ , et qu'elle converge vers une valeur  $\mu$  vérifiant :

$$\lim_{k \rightarrow +\infty} (J + J^\Sigma)(u^{(k)}) = \mu \geq J(u^\sharp) + J^\Sigma(u^\sharp), \quad (57)$$

et que la suite  $\{\|u^{(k)} - u^{(k+1)}\|\}_{k \in \mathbb{N}}$  converge vers zéro.

Enfin la coercivité de  $(J + J^\Sigma)$  implique que la suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$  est bornée.

3. Montrons que l'on a :  $\mu = J(u^\sharp) + J^\Sigma(u^\sharp)$ .

- De la convexité de  $J$ , on tire :

$$(J + J^\Sigma)(u^\sharp) - (J + J^\Sigma)(u^{(k+1)}) \geq \left\langle \nabla J(u^{(k+1)}), u^\sharp - u^{(k+1)} \right\rangle + J^\Sigma(u^\sharp) - J^\Sigma(u^{(k+1)}). \quad (58)$$

- L'évaluation des conditions d'optimalité du  $k$ -ème problème auxiliaire en  $u = u^\sharp$  fournit :

$$J^\Sigma(u^\sharp) - J^\Sigma(u^{(k+1)}) \geq \frac{1}{\epsilon^{(k)}} \left\langle \nabla K^{(k)}(u^{(k)}) - \nabla K^{(k)}(u^{(k+1)}) - \epsilon^{(k)} \nabla J(u^{(k)}), u^\sharp - u^{(k+1)} \right\rangle. \quad (59)$$

Remplacer l'expression (59) dans (58) conduit à la relation :

$$(J + J^\Sigma)(u^\sharp) - (J + J^\Sigma)(u^{(k+1)}) \geq \xi^{(k+1)}, \quad (60)$$

avec :

$$\xi^{(k+1)} = \left\langle \nabla J(u^{(k+1)}) - \nabla J(u^{(k)}), u^\sharp - u^{(k+1)} \right\rangle + \frac{1}{\epsilon^{(k)}} \left\langle \nabla K^{(k)}(u^{(k)}) - \nabla K^{(k)}(u^{(k+1)}), u^\sharp - u^{(k+1)} \right\rangle .$$

La valeur absolue de chacun des deux termes dont la somme vaut  $\xi^{(k+1)}$  se majore par application de l'inégalité de Schwartz et en utilisant le fait que  $J'$  et  $K^{(k)'}$  sont lipschitziennes de constantes respectives  $A$  et  $B^{(k)}$  :

$$\begin{aligned} |\xi^{(k+1)}| &\leq \left( A + \frac{B^{(k)}}{\epsilon^{(k)}} \right) \|u^{(k)} - u^{(k+1)}\| \|u^\sharp - u^{(k+1)}\| \\ &\leq \left( A + \frac{B}{\alpha} \right) \|u^{(k)} - u^{(k+1)}\| \|u^\sharp - u^{(k+1)}\| , \end{aligned}$$

d'après les hypothèses **H5** et **H6**. La suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$  étant bornée, le terme  $\|u^\sharp - u^{(k+1)}\|$  peut être majoré indépendamment de  $k$ . Comme la suite  $\{\|u^{(k)} - u^{(k+1)}\|\}_{k \in \mathbb{N}}$  converge vers zéro, on en déduit que :

$$\lim_{k \rightarrow +\infty} \xi^{(k)} = 0 ,$$

On a déjà montré que la suite  $\{\mathcal{L}(u^{(k)})\}_{k \in \mathbb{N}}$  est décroissante. De (57) et (60), on déduit :

$$(J + J^\Sigma)(u^\sharp) \leq (J + J^\Sigma)(u^{(k)}) \leq (J + J^\Sigma)(u^\sharp) - \xi^{(k)} ,$$

et donc que la suite  $\{\mathcal{L}(u^{(k)})\}_{k \in \mathbb{N}}$  converge vers  $J(u^\sharp) + J^\Sigma(u^\sharp)$ , d'où le résultat **R2**.

4. Comme la suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$  est bornée et que l'ensemble  $U^{\text{ad}}$  est fermé, on peut en extraire une sous-suite  $\{u^{(k_i)}\}$  convergeant vers un élément  $\bar{u} \in U^{\text{ad}}$ . Le caractère s.c.i. de la fonction  $(J + J^\Sigma)$  implique alors :

$$\lim_{k_i \rightarrow +\infty} (J + J^\Sigma)(u^{(k_i)}) \geq (J + J^\Sigma)(\bar{u}) .$$

La convergence de toute la suite  $\{\mathcal{L}(u^{(k)})\}_{k \in \mathbb{N}}$  implique :  $(J + J^\Sigma)(\bar{u}) = (J + J^\Sigma)(u^\sharp)$ . Cette égalité prouve que  $\bar{u}$  est une solution du problème initial (51), d'où le résultat **R3**.

On suppose de plus que la fonction  $J$  est fortement convexe de module  $a$ .

- L'évaluation des conditions d'optimalité du  $k$ -ème problème auxiliaire en  $u = u^\sharp$  fournit :

$$\left\langle \nabla K^{(k)}(u^{(k+1)}) + \epsilon^{(k)} \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}), u^\sharp - u^{(k+1)} \right\rangle + \epsilon^{(k)} \left( J^\Sigma(u^\sharp) - J^\Sigma(u^{(k+1)}) \right) \geq 0 .$$

- L'évaluation des conditions d'optimalité du problème initial en  $u = u^{(k+1)}$  fournit :

$$\epsilon^{(k)} \left\langle \nabla J(u^\sharp), u^{(k+1)} - u^\sharp \right\rangle + \epsilon^{(k)} \left( J^\Sigma(u^{(k+1)}) - J^\Sigma(u^\sharp) \right) \geq 0 .$$

Sommant ces deux dernières inégalités, et ajoutant de part et d'autre de l'inégalité résultante le terme additionnel  $\epsilon^{(k)} \langle \nabla J(u^{(k+1)}), u^{(k+1)} - u^\sharp \rangle$ , on obtient :

$$\begin{aligned} \epsilon^{(k)} \left\langle \nabla J(u^{(k+1)}) - \nabla J(u^\sharp), u^{(k+1)} - u^\sharp \right\rangle &\leq \epsilon^{(k)} \left\langle \nabla J(u^{(k)}) - \nabla J(u^{(k+1)}), u^\sharp - u^{(k+1)} \right\rangle + \\ &\quad \left\langle \nabla K^{(k)}(u^{(k+1)}) - \nabla K^{(k)}(u^{(k)}), u^\sharp - u^{(k+1)} \right\rangle . \end{aligned}$$

Utilisant la forte monotonie de  $J'$  sur le terme à gauche de cette nouvelle inégalité, et l'inégalité de Schwartz ainsi que le caractère lipschitzien sur les deux termes de droite, il vient :

$$\epsilon^{(k)} a \|u^{(k+1)} - u^\sharp\|^2 \leq \left( \epsilon^{(k)} A + B^{(k)} \right) \|u^{(k)} - u^{(k+1)}\| \|u^\sharp - u^{(k+1)}\| ,$$

d'où, après simplification :

$$\|u^{(k+1)} - u^\sharp\| \leq \frac{A + B^{(k)}/\epsilon^{(k)}}{a} \|u^{(k)} - u^{(k+1)}\| .$$

On en déduit immédiatement la majoration d'erreur **R4**. Avec les hypothèses **H5** et **H6**, on obtient :

$$\|u^{(k+1)} - u^\sharp\| \leq \frac{A + B/\alpha}{a} \|u^{(k)} - u^{(k+1)}\| ,$$

et la convergence vers zéro de la suite  $\{\|u^{(k)} - u^{(k+1)}\|\}_{k \in \mathbb{N}}$  assure la convergence globale de la suite  $\{u^{(k)}\}_{k \in \mathbb{N}}$  vers  $u^\sharp$ , d'où **R5**.  $\square$

### 3.2.3 Variantes de l'algorithme

Dans de nombreux cas, il est intéressant de pouvoir choisir les coefficients  $\epsilon^{(k)}$  tous égaux à la valeur 1. On donne ici deux variantes de l'algorithme permettant ce choix.

1. Une première méthode provient directement de l'examen de l'hypothèse **H6** du théorème de convergence. Pour faire le choix  $\epsilon^{(k)} = 1$ , il suffit en effet que la condition :

$$\frac{2b^{(k)}}{A + \beta} \geq 1 ,$$

soit vérifiée. Cette condition peut être obtenue en augmentant artificiellement le module de forte convexité  $b^{(k)}$  du noyau de décomposition par l'ajout d'un terme quadratique. En particulier, utiliser le noyau :

$$K^{(k)}(u) + \frac{\gamma^{(k)}}{2} \|u\|^2 ,$$

fait passer ce module à la valeur  $(b^{(k)} + \gamma^{(k)})$ . Le choix de coefficients  $\gamma^{(k)}$  suffisamment élevés permet alors d'assurer le respect des conditions de convergence de l'algorithme. Ceci revient à rajouter dans le critère du problème auxiliaire (53) le terme :

$$\frac{\gamma^{(k)}}{2} \|u\|^2 - \gamma^{(k)} \langle u^{(k)}, u \rangle .$$

Ajoutant le terme constant  $\gamma^{(k)} \|u^{(k)}\|^2/2$ , ce problème auxiliaire se met sous la forme :

$$\min_{u \in U^{\text{ad}}} \frac{\gamma^{(k)}}{2} \|u - u^{(k)}\|^2 + K^{(k)}(u) + \langle \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}), u \rangle + J^\Sigma(u) .$$

Le terme quadratique additionnel se comporte comme un *terme de freinage* autour du point  $u^{(k)}$ , qui ralentit donc la convergence de l'algorithme. C'est pourquoi il y a en fait intérêt à ne pas choisir les coefficients  $\gamma^{(k)}$  trop grands.

2. Une seconde méthode permettant d'assurer la convergence de l'algorithme tout en choisissant les coefficients  $\epsilon^{(k)}$  égaux à 1 consiste à *relaxer* les solutions des problèmes auxiliaires à l'aide de coefficients  $\rho^{(k)}$  choisis à l'avance. L'étape 2 de l'algorithme 1 est alors remplacée par les deux sous-étapes suivantes.

$$2.1 \text{ Calculer } u^{(k+\frac{1}{2})} = \arg \min_{u \in U^{\text{ad}}} \{ K^{(k)}(u) + \langle \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}), u \rangle + J^\Sigma(u) \} .$$

$$2.2 \text{ Calculer } u^{(k+1)} = \rho^{(k)} u^{(k+\frac{1}{2})} + (1 - \rho^{(k)}) u^{(k)} .$$

Le résultat  $u^{(k+1)}$  de l'itération  $k$  est obtenu en relaxant la solution  $u^{(k+\frac{1}{2})}$  du  $k$ -ème problème auxiliaire (dans lequel on a choisi  $\epsilon^{(k)} = 1$ ) avec le résultat  $u^{(k)}$  de l'itération précédente.

On peut montrer que les conditions de convergence de ce nouvel algorithme sont en tout point semblables à celles données dans le théorème 3, pourvu que l'on ait  $\rho^{(k)} \leq 1$ . Cette dernière condition assure que le point  $u^{(k+1)}$ , obtenu par combinaison convexe de deux points de  $U^{\text{ad}}$  est lui-même dans  $U^{\text{ad}}$  (car cet ensemble est un convexe).

Ces deux variantes sont en fait proches l'une de l'autre, la première consistant à empêcher *a priori* que la solution du problème auxiliaire au point  $u^{(k)}$  ne s'écarte trop de ce point (en y incorporant un terme quadratique de freinage), la seconde produisant le même effet *a posteriori* en relaxant la solution du problème auxiliaire au point  $u^{(k)}$  avec ce dernier point.

### 3.2.4 Intérêts théoriques et pratiques

**Algorithmique.** Le premier intérêt du principe du problème auxiliaire est théorique. Il permet en effet de retrouver les principaux algorithmes d'optimisation dans un cadre unifié où l'étude de convergence a été effectuée une fois pour toute. Ainsi, dans le cas  $J^\Sigma \equiv 0$ ,

— le choix d'un noyau de décomposition purement quadratique :

$$K(u) = \frac{1}{2} \|u\|^2,$$

conduit au problème auxiliaire suivant :

$$\min_{u \in U^{\text{ad}}} \frac{1}{2} \|u\|^2 + \langle \epsilon^{(k)} \nabla J(u^{(k)}) - u^{(k)}, u \rangle,$$

dont la solution se calcule de manière analytique et correspond à une itération de l'algorithme de gradient projeté sur l'ensemble  $U^{\text{ad}}$  :

$$u^{(k+1)} = \text{proj}_{U^{\text{ad}}} \left( u^{(k)} - \epsilon^{(k)} \nabla J(u^{(k)}) \right);$$

— le choix d'un noyau de décomposition approximant  $J$  au second ordre :

$$K^{(k)}(u) = \frac{1}{2} \langle u, J''(u^{(k)}) . u \rangle,$$

( $J''(u^{(k)})$  étant le hessien de  $J$  au point  $u^{(k)}$ ) conduit au problème auxiliaire suivant :

$$\min_{u \in U^{\text{ad}}} \frac{1}{2} \langle u, J''(u^{(k)}) . u \rangle + \langle \epsilon^{(k)} \nabla J(u^{(k)}) - J''(u^{(k)}) . u^{(k)}, u \rangle,$$

dont la solution se calcule de manière analytique dans le cas où  $U^{\text{ad}} = \mathcal{U}$  :

$$u^{(k+1)} = u^{(k)} - \epsilon^{(k)} [J''(u^{(k)})]^{-1} . \nabla J(u^{(k)}),$$

et correspond exactement à une itération de l'algorithme de Newton-Raphson avec  $\epsilon^{(k)} = 1$ . On notera qu'il est essentiel que le noyau de décomposition  $K^{(k)}$  puisse dépendre de l'indice d'itération  $k$  pour justifier l'utilisation de l'algorithme 1 dans ce cas.

**Remarque 12.** Dans le cas où  $U^{\text{ad}}$  est différent de  $\mathcal{U}$ , on ne peut plus calculer de manière analytique la solution de ce problème auxiliaire. <sup>21</sup>

**Décomposition.** Le second et principal intérêt (du moins pour ce cours) du principe du problème auxiliaire concerne la décomposition. Comme on l'a dit au début de ce chapitre, on se place dans le cas où l'espace  $\mathcal{U}$  admet une décomposition  $\mathcal{U}_1 \times \dots \times \mathcal{U}_N$  telle que l'ensemble  $U^{\text{ad}}$  se mette sous la forme d'un produit cartésien et que la fonction  $J^\Sigma$  soit additive par rapport à cette décomposition. Si on choisit alors une suite de noyaux  $K^{(k)}$  eux aussi additifs par rapport à cette décomposition :

$$K^{(k)}(u_1, \dots, u_N) = \sum_{i=1}^N K_i^{(k)}(u_i),$$

le terme  $\langle \epsilon^{(k)} \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}), u \rangle$  prend la forme suivante :

$$\sum_{i=1}^N \left\langle \epsilon^{(k)} \nabla_{u_i} J(u_1^{(k)}, \dots, u_N^{(k)}) - \nabla K_i^{(k)}(u_i^{(k)}), u_i \right\rangle,$$

---

21. Hormis le cas sphérique, la projection sur  $U^{\text{ad}}$  de la solution d'un problème d'optimisation quadratique sans contrainte n'est pas identique à la solution du même problème avec contrainte d'appartenir à  $U^{\text{ad}}$ .

(où  $\nabla_{u_i} J(u_1^{(k)}, \dots, u_N^{(k)})$  est le gradient partiel de  $J$  par rapport à  $u_i$  évalué au point  $(u_1^{(k)}, \dots, u_N^{(k)})$ ) de telle sorte que le  $k$ -ème problème auxiliaire s'écrit :

$$\min_{(u_1, \dots, u_N) \in U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}} \sum_{i=1}^N \left( K_i^{(k)}(u_i) + \left\langle \epsilon^{(k)} \nabla_{u_i} J(u_1^{(k)}, \dots, u_N^{(k)}) - \nabla K_i^{(k)}(u_i^{(k)}), u_i \right\rangle + \epsilon^{(k)} J_i^\Sigma(u_i) \right) .$$

Ce problème auxiliaire se scinde alors en  $N$  sous-problèmes indépendants dont la  $i$ -ème instance est :

$$\min_{u_i \in U_i^{\text{ad}}} K_i^{(k)}(u_i) + \left\langle \epsilon^{(k)} \nabla_{u_i} J(u_1^{(k)}, \dots, u_N^{(k)}) - \nabla K_i^{(k)}(u_i^{(k)}), u_i \right\rangle + \epsilon^{(k)} J_i^\Sigma(u_i) .$$

Ces  $N$  sous-problèmes peuvent être résolus *en parallèle*, la concaténation des  $N$  solutions  $u_i^{(k+1)}$  permettant de formuler les sous-problèmes auxiliaires de l'itération suivante, et on parle alors de mise en œuvre de type *Jacobi*. Dans le cas où l'on ne dispose pas d'un ordinateur parallèle de type MIMD (Multiple Instructions Multiple Data), on préférera utiliser une version *séquentielle* (dite de type *Gauss-Seidel*) de l'algorithme dans laquelle la solution  $u_{i-1}^{(k+1)}$  du  $(i-1)$ -ème sous-problème auxiliaire à l'itération  $k$  est ré-injectée dans le  $i$ -ème sous-problème à la même itération :

$$\min_{u_i \in U_i^{\text{ad}}} K_i^{(k)}(u_i) + \left\langle \epsilon^{(k)} \nabla_{u_i} J(u_1^{(k+1)}, \dots, u_{i-1}^{(k+1)}, u_i^{(k)}, \dots, u_N^{(k)}) - \nabla K_i^{(k)}(u_i^{(k)}), u_i \right\rangle + \epsilon^{(k)} J_i^\Sigma(u_i) .$$

Les sous-problèmes ne peuvent alors plus être résolus en parallèle. On constate en pratique que la version séquentielle converge en un nombre d'itérations plus faible que la version parallèle, cette dernière ne permettant un gain de temps que sur une machine disposant de plusieurs processeurs.

**Remarque 13.** Parmi l'ensemble des choix possibles pour les noyaux de décomposition, il en existe un, appelé *choix canonique*, consistant à préserver autant qu'il est possible du problème initial dans les problèmes auxiliaires tout en assurant le caractère décomposable du noyau. Pour une mise en œuvre parallèle, ce choix est le suivant :

$$K^{(k)}(u_1, \dots, u_N) = \sum_{i=1}^N K_i^{(k)}(u_i) = \sum_{i=1}^N J \left( u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i, u_{i+1}^{(k)}, \dots, u_N^{(k)} \right) .$$

On a alors  $\nabla K_i^{(k)}(u_i^{(k)}) = \nabla_{u_i} J(u_1^{(k)}, \dots, u_N^{(k)})$ , si bien que le terme  $\langle \epsilon^{(k)} \nabla J(u^{(k)}) - \nabla K^{(k)}(u^{(k)}), u \rangle$  s'annule pourvu qu'on puisse prendre  $\epsilon^{(k)} = 1$ . Ajoutant les termes quadratiques  $\gamma^{(k)} \|u\|^2$  permettant d'assurer ce dernier choix, les sous-problèmes auxiliaires prennent la forme simplifiée suivante :

$$\min_{u_i \in U_i^{\text{ad}}} \frac{\gamma^{(k)}}{2} \|u_i - u_i^{(k)}\|^2 + J \left( u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i, u_{i+1}^{(k)}, \dots, u_N^{(k)} \right) + J_i^\Sigma(u_i) .$$

**Conditionnement.** Enfin, le dernier intérêt que l'on mentionnera concerne le conditionnement numérique des problèmes d'optimisation. Le PPA permet en effet de remplacer un problème dont la résolution s'avère numériquement difficile par une suite de problèmes bien conditionnés de ce point de vue. À titre d'exemple, on considère le problème :

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

et l'on suppose qu'il se prête a priori bien à une résolution par la méthode de Newton, à ceci près que le hessien de  $J$  n'est pas défini positif à l'optimum  $u^\#$ .<sup>22</sup> Alors, le conditionnement de la matrice hessienne  $J''(u)$  se détériore au voisinage de  $u^\#$ , et l'itération de l'algorithme de Newton :

$$u^{(k+1)} = u^{(k)} - [J''(u^{(k)})]^{-1} \cdot \nabla J(u^{(k)}) ,$$

“explose” lorsque  $u^{(k)}$  tend vers  $u^\#$ .

22. C'est le cas du problème de l'équilibre d'un réseau de distribution d'eau : l'inversion de la matrice des dérivées secondes est aisée *génériquement*, mais cette matrice devient *singulière* lorsque le débit dans un tuyau est nul.

On applique maintenant le principe du problème auxiliaire avec les noyaux :

$$\frac{\gamma}{2} \|u\|^2 + \frac{1}{2} \langle u, J''(u^{(k)}) \cdot u \rangle ,$$

et avec  $\epsilon^{(k)} = 1$ . Le problème auxiliaire au point  $u^{(k)}$  est quadratique, sa matrice hessienne est égale à  $(\gamma I + J''(u^{(k)}))$  et elle est donc définie positive dès que la matrice  $J''(u^{(k)})$  est semi-définie positive. Le problème auxiliaire se résout alors de manière explicite et sa solution est :

$$u^{(k+1)} = u^{(k)} - [\gamma I + J''(u^{(k)})]^{-1} \cdot \nabla J(u^{(k)}) ,$$

que l'on peut interpréter comme l'itération d'une méthode de Newton "avec reconditionnement". Comme on l'a déjà noté, le choix du coefficient  $\gamma$  résulte d'un compromis entre, d'une part le respect des conditions de convergence de l'algorithme ( $\gamma$  doit être "grand"), et d'autre part le ralentissement induit par le terme quadratique de freinage ( $\gamma$  doit être "petit").

### 3.2.5 Complément : lien avec la "Mirror Descent Method"

Le principe du problème auxiliaire a été présenté jusqu'alors dans le cadre des *espaces de Hilbert* de dimension finie, et sa généralisation au cas des espaces de Hilbert de dimension infinie est aisée. On rappelle que dans un espace de Hilbert  $\mathcal{H}$ , le théorème de représentation de Riesz permet d'identifier de manière unique tout élément de l'espace dual  $\mathcal{H}^*$  de l'espace  $\mathcal{H}$  avec un élément de  $\mathcal{H}$  lui-même. Cette identification est la clé permettant d'écrire une formule de type gradient, puisque le point courant et le gradient vivent alors tous deux dans le même espace de Hilbert.

La situation change lorsque l'on se pose la question de mettre en œuvre des méthodes de type gradient dans un *espace de Banach* car le théorème d'identification de Riesz ne peut plus être appliqué. Pour surmonter cette difficulté, A. Nemirovski et D. B. Yudin ont proposé une méthode appelée "Mirror Descent Method" qui se révèle être identique au principe du problème auxiliaire tout en étant basée sur des idées différentes. On va donner ici les éléments essentiels de cette méthode.

On considère le problème :

$$\min_{u \in U^{\text{ad}} \subset \mathcal{U}} J(u) , \tag{61}$$

où l'ensemble  $U^{\text{ad}}$  est un convexe fermé d'intérieur non vide d'un espace de Banach  $\mathcal{U}$  et où la fonction  $J$  est convexe, s.c.i. propre, coercive et différentiable. On notera  $\mathcal{U}^*$  le dual topologique de l'espace  $\mathcal{U}$  et  $\langle \cdot, \cdot \rangle : \mathcal{U}^* \times \mathcal{U} \rightarrow \mathbb{R}$  le produit de dualité associé. On notera  $J^*$  la conjuguée de Fenchel de la fonction  $J$  :

$$J^*(u^*) = \sup_{u \in \mathcal{U}} \langle u^*, u \rangle - J(u) . \tag{62}$$

On définit la distance de Bregman  $D_K$  associée à une fonction  $K : \mathcal{U} \rightarrow \mathbb{R}$ , supposée fortement convexe et continuellement différentiable, de la manière suivante :

$$D_K(u, v) = K(u) - K(v) - \langle \nabla K(v), u - v \rangle . \tag{63}$$

La méthode du Mirror Descent Algorithm permet de contourner le fait que l'on ne peut faire directement pour le problème (61) une méthode de gradient de la forme

$$u^{(k+1)} = \text{proj}_{U^{\text{ad}}} (u^{(k)} - t^{(k)} \nabla J(u^{(k)})) , \tag{64}$$

dans la mesure où les éléments  $u^{(k)}$  et  $\nabla J(u^{(k)})$  vivent respectivement dans les espaces  $\mathcal{U}$  et  $\mathcal{U}^*$ , qui sont différents sauf dans le cas Hilbertien. La méthode du Mirror Descent Algorithm consiste à faire cette étape de gradient dans l'espace dual en se servant de la distance de Bregman associée à une fonction  $K$  que l'on choisit a priori. Plus précisément, partant d'un point  $u^{(k)}$ , une itération de cette méthode consiste en les étapes suivantes.

1. Mappage du point  $u^{(k)} \in \mathcal{U}$  dans l'espace dual  $\mathcal{U}^*$  :

$$u^{(k)} \mapsto y^{(k)} = \nabla K(u^{(k)}) . \quad (65a)$$

2. Pas de gradient dans l'espace dual  $\mathcal{U}^*$  :

$$y^{(k)} \mapsto y^{(k+1)} = y^{(k)} - t^{(k)} \nabla J(u^{(k)}) . \quad (65b)$$

3. Mappage du point  $y^{(k+1)} \in \mathcal{U}^*$  dans l'espace primal  $\mathcal{U}$  :

$$y^{(k+1)} \mapsto v^{(k+1)} = \nabla K^*(y^{(k+1)}) . \quad (65c)$$

4. Projection sur l'ensemble admissible  $U^{\text{ad}}$  pour la distance  $D_K$  :

$$v^{(k+1)} \mapsto u^{(k+1)} = \arg \min_{u \in U^{\text{ad}}} D_K(u, v^{(k+1)}) . \quad (65d)$$

Compte tenu de la propriété suivante de la conjuguée de Fenchel :

$$\nabla K(\nabla K^*(y)) = y , \quad (66)$$

on obtient que l'itération (65) de l'algorithme associé au Mirror Descent Algorithm, et plus précisément la minimisation (65d), se met sous la forme :

$$u^{(k+1)} = \arg \min_{u \in U^{\text{ad}}} K(u) + \langle t^{(k)} \nabla J(u^{(k)}) - \nabla K(u^{(k)}), u \rangle , \quad (67)$$

ce qui correspond exactement à l'algorithme associé au principe du problème auxiliaire.

### 3.3 Optimisation sous contraintes explicites

On se place maintenant dans le cadre de l'optimisation sous contraintes explicites, et on s'intéresse au problème suivant :

$$\min_{u \in U^{\text{ad}}} J(u) + J^\Sigma(u) . \quad (68a)$$

sous les contraintes :

$$\Theta(u) + \Theta^\Sigma(u) \in -C . \quad (68b)$$

Le Lagrangien de ce problème est :

$$J(u) + J^\Sigma(u) + \langle p, \Theta(u) + \Theta^\Sigma(u) \rangle , \quad (69)$$

qui se présente comme la somme de deux Lagrangiens  $L$  et  $L^\Sigma$ , à savoir :

$$L(u, p) = J(u) + \langle p, \Theta(u) \rangle \quad \text{et} \quad L^\Sigma(u, p) = J^\Sigma(u) + \langle p, \Theta^\Sigma(u) \rangle . \quad (70)$$

On s'intéresse alors à la question de trouver un point selle du Lagrangien  $L + L^\Sigma$ , ce qui, moyennant une hypothèse supplémentaire (par rapport au cas sans contrainte explicite) de qualification des contraintes, est équivalent à la question de trouver une solution du problème (68).

### 3.3.1 Extension du principe du problème auxiliaire

Comme au §3.2, on suppose que l'espace  $\mathcal{U}$  admet une décomposition  $\mathcal{U}_1 \times \dots \times \mathcal{U}_N$  telle que l'ensemble  $U^{\text{ad}}$  se mette sous la forme du produit cartésien  $U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}$  de  $N$  ensembles convexes fermés non vides. L'écriture des fonctions de coût, de contraintes et du Lagrangien comme la somme de deux termes est là pour souligner que l'on considère que les fonctions  $J^\Sigma$  et  $\Theta^\Sigma$  (et donc  $L^\Sigma$ ) ne poseront pas de difficulté lorsque l'on décomposera le problème (on y reviendra au §3.3.2). L'extension du principe du problème auxiliaire au cas des problèmes de point selle passe, comme au §3.2, par les deux points suivants :

1. on suppose que les fonctions  $J$  et  $\Theta$  (qui rendent le problème "difficile") sont *différentiables*, ce qui permet de les remplacer par leur approximation au premier ordre autour d'un point  $u^{(k)}$  ;
2. on se donne un noyau de décomposition  $\Lambda$  défini sur  $\mathcal{U} \times \mathcal{C}$  à valeurs dans  $\mathbb{R}$ , *fortement convexe* en  $u$  et *fortement concave* en  $p$ ,<sup>23</sup> ainsi qu'une constante  $\epsilon > 0$ .

Le problème auxiliaire au point  $(u^{(k)}, p^{(k)})$  associé au problème (68) consiste à trouver, dans  $U^{\text{ad}} \times C^\star$ , le point selle du Lagrangien auxiliaire  $\mathfrak{L}^{(k)}$  défini par :<sup>24</sup>

$$\mathfrak{L}^{(k)}(u, p) = \Lambda(u, p) + \langle (\epsilon \nabla_u L - \nabla_u \Lambda)(u^{(k)}, p^{(k)}), u \rangle + \langle (\epsilon \nabla_p L - \nabla_p \Lambda)(u^{(k)}, p^{(k)}), p \rangle + \epsilon L^\Sigma(u, p), \quad (71)$$

$\nabla_u L$  (resp.  $\nabla_u \Lambda$ ) et  $\nabla_p L$  (resp.  $\nabla_p \Lambda$ ) étant les gradients partiels de  $L$  (resp.  $\Lambda$ ) par rapport aux variables  $u$  et  $p$ . L'unique point selle  $(u^{(k+1)}, p^{(k+1)})$  du Lagrangien auxiliaire  $\mathfrak{L}^{(k)}$  est caractérisé par les conditions d'optimalité :

$$\begin{aligned} \forall u \in U^{\text{ad}}, \quad & \langle \nabla_u \Lambda(u^{(k+1)}, p^{(k+1)}) + \epsilon \nabla_u (L - \Lambda)(u^{(k)}, p^{(k)}), u - u^{(k+1)} \rangle + \epsilon (L^\Sigma(u, p^{(k+1)}) - L^\Sigma(u^{(k+1)}, p^{(k+1)})) \geq 0 \\ \forall p \in C^\star, \quad & \langle \nabla_p \Lambda(u^{(k+1)}, p^{(k+1)}) + \epsilon \nabla_p (L - \Lambda)(u^{(k)}, p^{(k)}), p - p^{(k+1)} \rangle + \epsilon (L^\Sigma(u^{(k+1)}, p) - L^\Sigma(u^{(k+1)}, p^{(k+1)})) \leq 0. \end{aligned}$$

Partant d'un point  $(u^{(0)}, p^{(0)})$  appartenant à  $U^{\text{ad}} \times C^\star$ , on construit la suite  $\{(u^{(k)}, p^{(k)})\}_{k \in \mathbb{N}}$  engendrée par l'enchaînement des résolutions des problèmes auxiliaires (71). Si l'on suppose que cette suite converge vers  $(\bar{u}, \bar{p})$ , les conditions d'optimalité du problème auxiliaire au point  $(\bar{u}, \bar{p})$  se simplifient et montrent que  $(\bar{u}, \bar{p})$  est un point selle du Lagrangien (69), et donc que  $\bar{u}$  est une solution du problème (68). Comme au §3.2, ceci suggère l'algorithme de résolution suivant.

#### Algorithme 2.

0. Choisir un noyau de décomposition  $\Lambda$  et un coefficient  $\epsilon > 0$ .
1. Choisir un point  $(u^{(0)}, p^{(0)}) \in U^{\text{ad}} \times C^\star$  et un seuil de convergence  $\sigma$  ; poser  $k = 0$ .
2. Calculer le point selle  $(u^{(k+1)}, p^{(k+1)})$  du Lagrangien auxiliaire (71).
3. Refaire l'étape 2 en incrémentant  $k$  de 1 tant que  $\|u^{(k+1)} - u^{(k)}\| + \|p^{(k+1)} - p^{(k)}\| > \sigma$ .

Dans le déroulement de l'algorithme précédent, l'étape 2 correspondant au calcul du point selle, c'est-à-dire à la résolution *simultanée* des conditions d'optimalité associées, peut être remplacée par la résolution *séquentielle* de ces conditions. L'étape 2 de l'algorithme 2 est alors remplacée par les deux sous-étapes suivantes.

- 2.1 Calculer  $u^{(k+1)} = \arg \min_{u \in U^{\text{ad}}} \mathfrak{L}^{(k)}(u, p^{(k)})$ .
- 2.2 Calculer  $p^{(k+1)} = \arg \max_{p \in C^\star} \mathfrak{L}^{(k)}(u^{(k+1)}, p)$ .

23. La forte convexité en  $u$  (resp. forte concavité en  $p$ ) permet d'assurer la coercivité lors de la phase de minimisation (resp. maximisation) dans la recherche du point selle.

24. ce point selle est unique grâce aux hypothèses faites sur  $\Lambda$



On dispose ainsi de deux familles d'algorithmes, l'une effectuant directement la recherche du point selle et l'autre alternant une phase de minimisation primale et une phase de maximisation duale.

### 3.3.2 Décomposition naturelle d'un problème sous contraintes

Avant d'examiner plus en détail ces deux familles d'algorithmes, il reste à préciser ce que l'on entend par " $J^\Sigma$  et  $\Theta^\Sigma$  ne posent pas de difficulté en décomposition". Pour cela, on considère le problème :

$$\begin{aligned} & \min_{u \in U^{\text{ad}}} J^\Sigma(u) \\ & \text{sous la contrainte} \\ & \Theta^\Sigma(u) \in -C, \end{aligned}$$

dont la solution consiste à trouver un point selle sur  $U^{\text{ad}} \times C^*$  du Lagrangien :

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

Deux points de vue sont envisageables.

1. On souhaite décomposer la recherche de point selle elle-même et on cherche donc à écrire le Lagrangien sous forme de  $N$  Lagrangiens indépendants :

$$L^\Sigma(u, p) = \sum_{i=1}^N L_i^\Sigma(u_i, p_i) ,$$

de telle sorte que le point selle global soit la concaténation des points selle des Lagrangiens indépendants. La décomposition désirée de la variable duale  $p$  en  $(p_1, \dots, p_N)$  correspond à une décomposition de l'espace  $\mathcal{C}$  et du cône  $C$  des contraintes en :

$$\mathcal{C} = \mathcal{C}_1 \times \dots \times \mathcal{C}_N \quad \text{et} \quad C = C_1 \times \dots \times C_N ,$$

et la recherche du point selle du Lagrangien  $L_i^\Sigma$  est effectuée dans  $U_i^{\text{ad}} \times C_i$ . Ce Lagrangien s'interprète comme un problème d'optimisation de la forme :

$$\min_{u_i \in U_i^{\text{ad}}} J_i^\Sigma(u_i) \quad \text{sous} \quad \Theta_i^\Sigma(u_i) \in -C_i .$$

Dans ce cas, chaque fonction  $\Theta_i$  est à valeurs dans un sous-espace  $\mathcal{C}_i$  de  $\mathcal{C}$  qui lui est *propre* :

$$\Theta_i : \mathcal{U}_i \longrightarrow \mathcal{C}_i ,$$

et les conditions de décomposition sont d'une part que la fonction de coût soit additive :

$$J^\Sigma(u_1, \dots, u_N) = \sum_{i=1}^N J_i^\Sigma(u_i) ,$$

et d'autre part que la fonction de contraintes soit de nature bloc-diagonale<sup>25</sup> par rapport aux décompositions respectives des espaces  $\mathcal{U}$  et  $\mathcal{C}$  :

$$\Theta^\Sigma(u_1, \dots, u_N) = \begin{pmatrix} \Theta_1^\Sigma(u_1) \\ \vdots \\ \Theta_N^\Sigma(u_N) \end{pmatrix} = \begin{pmatrix} \Theta_1^\Sigma & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \Theta_N^\Sigma \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ \vdots \\ u_N \end{pmatrix} .$$

---

25. dans le sens où il n'existe pas dans les contraintes de termes de la forme  $\Theta_i^\Sigma(u_j)$ , avec  $i \neq j$

On retrouve le cas où l'ensemble des contraintes du problème s'exprime sous la forme d'un produit cartésien d'ensembles admissibles, chacun de ces ensembles étant de la forme :

$$\{u_i \in U_i^{\text{ad}} \text{ et } \Theta_i^\Sigma(u_i) \in -C_i\} .$$

On retrouve aussi la méthode de décomposition par prédiction dans le cas très simple où les contraintes affectées à un sous-problème ne dépendent que des variables de ce sous-problème. On notera que la condition de type bloc-diagonal sur  $\Theta$  est beaucoup plus forte que la condition d'additivité "classique" que l'on avait rencontrée jusqu'alors, car chaque composante de  $\Theta$  doit maintenant dépendre d'une unique composante  $u_i$  de la variable  $u$ .

2. Dans la recherche du point selle, on se contente de décomposer la phase de minimisation en  $u$  à  $p$  fixé. Cette décomposition ne présente pas de difficulté si le Lagrangien est additif par rapport à la variable  $u$  suivant la décomposition de l'espace  $\mathcal{U}$ , et il suffit pour cela que les fonctions  $J^\Sigma$  et  $\Theta^\Sigma$  soient elles mêmes additives :

$$J^\Sigma(u_1, \dots, u_N) = \sum_{i=1}^N J_i^\Sigma(u_i) \quad \text{et} \quad \Theta^\Sigma(u_1, \dots, u_N) = \sum_{i=1}^N \Theta_i^\Sigma(u_i) .$$

La minimisation à  $p = p^{(k)}$  fixé se décompose alors en  $N$  sous-problèmes indépendants :

$$\min_{u_i \in U_i^{\text{ad}}} J_i^\Sigma(u_i) + \langle p^{(k)}, \Theta_i^\Sigma(u_i) \rangle .$$

La remise à jour des variables  $p^{(k)}$  peut être effectuée par un algorithme de type gradient, et on obtient ainsi la méthode de décomposition par les prix décrite au §2.2. On notera que, dans ce cas, toutes les fonctions  $\Theta_i$  sont à valeurs dans le même espace  $\mathcal{C}$  :

$$\Theta_i : \mathcal{U}_i \longrightarrow \mathcal{C} .$$

On constate donc qu'il y a au moins deux interprétations à la question de la "décomposabilité" naturelle d'un problème sous contraintes. On va dans la suite choisir des noyaux de décomposition  $\Lambda$  particuliers, de manière à être ramené à l'une ou l'autre de ces interprétations.

### 3.3.3 Généralisation de la méthode de décomposition par les prix

La méthode de décomposition par les prix exposée au §2.2 consistait essentiellement à faire alterner des phases de minimisation complète en  $u$  avec des étapes de gradient en  $p$  sur un Lagrangien qui se décomposait en  $u$  une fois la variable  $p$  fixée. Pour en effectuer la généralisation, on va se placer dans le cadre de la *variante séquentielle* de l'algorithme 2 (phases de minimisation et de maximisation consécutives), la partie "facile"  $L^\Sigma$  du Lagrangien étant additive par rapport à la variable  $u$ . On suppose donc qu'il existe une décomposition  $(u_1, \dots, u_N)$  de  $u$  telle que :

- $U^{\text{ad}}$  se mette sous la forme d'un produit cartésien par rapport à cette décomposition ;
- les fonctions  $J^\Sigma$  et  $\Theta^\Sigma$  soient additives par rapport à cette décomposition.

En ce qui concerne le choix du noyau de décomposition  $\Lambda$ , on remarque que :

- comme on cherche à mettre à jour la variable duale  $p$  par une méthode de type gradient, il est raisonnable de choisir une dépendance en  $p$  du noyau de la forme  $-\|p\|^2$  ;
- pour pouvoir décomposer la phase de minimisation en  $u$ , il est nécessaire que le noyau de décomposition dépende de  $u$  de manière additive, tout en ayant une forme  $K(u)$  assez générale.

Ceci conduit au choix de noyau de décomposition suivant :

$$\Lambda(u, p) = K(u) - \frac{1}{2\alpha} \|p\|^2 , \tag{72}$$

où  $K$  est une fonction fortement convexe différentiable et où  $\alpha$  est une constante positive soulignant la forte concavité en  $p$  du noyau  $\Lambda$ . Avec ce choix, l'étape 2.2 de maximisation en  $p$  dans la variante séquentielle de l'algorithme 2 prend la forme d'un problème quadratique (en fait sphérique) sous contraintes dont la solution peut être calculée explicitement. L'algorithme qui en résulte est décrit ci-dessous. Pour tout ce qui concerne la convergence de cet algorithme (théorème, démonstration et remarques), on se reportera au §3.3.2.3 dans le polycopié du cours.

### Algorithme 3.

0. Choisir un noyau de décomposition  $K$  et des coefficients  $\alpha$  et  $\epsilon$  tous deux positifs .
1. Choisir un point  $(u^{(0)}, p^{(0)}) \in U^{\text{ad}} \times C^*$  et un seuil de convergence  $\sigma$  ; poser  $k = 0$  .
- 2.1 Résoudre  $\min_{u \in U^{\text{ad}}} K(u) + \langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}), u \rangle + \epsilon J^\Sigma(u) + \epsilon \langle p^{(k)}, \Theta'(u^{(k)}) \cdot u + \Theta^\Sigma(u) \rangle$ ,  
et noter  $u^{(k+1)}$  la solution de ce problème.
- 2.2 Calculer  $p^{(k+1)} = \text{proj}_{C^*} (p^{(k)} + \rho(\Theta + \Theta^\Sigma)(u^{(k+1)}))$ , avec  $\rho = \epsilon\alpha$ .
3. Refaire l'étape 2 en incrémentant  $k$  de 1 tant que  $\|u^{(k+1)} - u^{(k)}\| + \|p^{(k+1)} - p^{(k)}\| > \sigma$  .

Cet algorithme constitue une *extension* des algorithmes de Uzawa et de Arrow-Hurwicz, que l'on retrouve d'ailleurs exactement avec les choix suivants :<sup>26</sup>

- **Uzawa** :  $J^\Sigma = 0$ ,  $\Theta = 0$ ,  $K = J$  ;
- **Arrow-Hurwicz** :  $J^\Sigma = 0$ ,  $\Theta^\Sigma = 0$ ,  $K = \|u\|^2/2$ .

On remarque en particulier que l'étape 2.1 correspond exactement au problème de minimisation que l'on obtiendrait en écrivant le problème auxiliaire (53) associé à la minimisation par rapport à  $u$  du Lagrangien  $L + L^\Sigma$  évalué au point  $p = p^{(k)}$ .

L'algorithme 3 généralise donc l'algorithme de décomposition par les prix au cas où les fonctions  $J$  et  $\Theta$  sont *différentiables* plutôt qu'*additives*. Son principal intérêt est qu'il suffit de choisir un noyau  $K$  lui-même additif pour que l'étape 2.1 de minimisation en  $u$  éclate en  $N$  sous-problèmes de minimisation indépendants, l'expression du  $i$ -ème sous-problème étant :

$$\min_{u_i \in U_i^{\text{ad}}} K_i(u_i) + \langle \epsilon \nabla_{u_i} J(u^{(k)}) - \nabla K(u_i^{(k)}), u_i \rangle + \epsilon J_i^\Sigma(u_i) + \epsilon \langle p^{(k)}, \Theta'_{u_i}(u^{(k)}) \cdot u_i + \Theta_i^\Sigma(u_i) \rangle . \quad (73)$$

Parmi les possibilités de noyaux de décomposition, on notera les deux choix canoniques suivants :

1.  $K_i(u_i) = J(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i, u_{i+1}^{(k)}, \dots, u_N^{(k)})$ ,
2.  $K_i(u_i) = J(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i, u_{i+1}^{(k)}, \dots, u_N^{(k)}) + \langle p^{(k)}, \Theta(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i, u_{i+1}^{(k)}, \dots, u_N^{(k)}) \rangle$ ,

permettant de simplifier la forme des problèmes auxiliaires pourvu que l'on puisse faire le choix  $\epsilon = 1$ . On notera que ces choix correspondent à des noyaux de décomposition dépendant de l'indice  $k$  des itérations, ce qui va au-delà du théorème de convergence dont on dispose pour cet algorithme, mais est largement utilisé en pratique.

#### 3.3.4 Généralisation de la méthode de décomposition par prédiction

L'esprit de la méthode de décomposition par prédiction telle qu'elle a été exposée au § 2.4 consiste à répartir les contraintes initiales entre les sous-problèmes,<sup>27</sup> et à chercher un point selle pour chacun des sous-problèmes. La généralisation de cette méthode consiste donc à se placer dans le cadre de l'algorithme 2 original (recherche de point selle), la partie "facile"  $L^\Sigma$  du problème se

26. pourvu que l'on puisse prendre  $\epsilon = 1$ , ce qui passe par l'ajout éventuel d'un terme quadratique en  $u$  au noyau  $K$

27. certains sous-problèmes pouvant ne se voir attribuer aucune contrainte

décomposant en  $u$  et en  $p$  en une somme de Lagrangiens indépendants. On suppose donc qu'il existe une décomposition  $(u_1, \dots, u_N)$  de la variable  $u$  et une décomposition  $(p_1, \dots, p_N)$  de la variable  $p$  telle que

- $U^{\text{ad}}$  se mette sous la forme d'un produit cartésien par rapport à la décomposition en  $u$  ;
- la fonction  $J^\Sigma$  soit additive par rapport à la décomposition en  $u$  ;
- la fonction  $\Theta^\Sigma$  soit bloc-diagonale par rapport aux décompositions en  $u$  et en  $p$ .

Afin de pouvoir réinterpréter le calcul du point selle dans l'algorithme 2 comme un problème d'optimisation sous contraintes, on choisit un noyau de décomposition  $\Lambda$  qui se mette sous la forme d'un Lagrangien, à savoir :

$$\Lambda(u, p) = K(u) + \langle p, \Omega(u) \rangle , \quad (74)$$

où  $K$  est une fonction fortement convexe différentiable et où  $\Omega$  est une fonction différentiable à valeurs dans l'espace  $\mathcal{C}$ . On notera que le noyau  $\Lambda$  ainsi choisi n'est pas toujours fortement concave en  $p$ . Le choix du noyau de décomposition consiste donc à se donner, outre la fonction  $K$ , un opérateur de contraintes  $\Omega$  de même nature que  $\Theta$ . L'algorithme qui en résulte est le suivant.

#### Algorithme 4.

0. Choisir les fonctions  $K$  et  $\Omega$  du noyau de décomposition, ainsi qu'un coefficient  $\epsilon > 0$  .
  1. Choisir un point  $(u^{(0)}, p^{(0)}) \in U^{\text{ad}} \times C^*$  et un seuil de convergence  $\sigma$  ; poser  $k = 0$  .
  2. Résoudre  $\min_{u \in U^{\text{ad}}} K(u) + \langle \epsilon \nabla J(u^{(k)}) - \nabla K(u^{(k)}), u \rangle + \epsilon J^\Sigma(u) + \langle p^{(k)}, (\epsilon \Theta'(u^{(k)}) - \Omega'(u^{(k)})) . u \rangle$   
sous la contrainte  $\Omega(u) + \epsilon \Theta^\Sigma(u) - \Omega(u^{(k)}) + \epsilon \Theta(u^{(k)}) \in -C$
- et noter  $(u^{(k+1)}, p^{(k+1)})$  un point selle de ce problème.
3. Refaire l'étape 2 en incrémentant  $k$  de 1 tant que  $\|u^{(k+1)} - u^{(k)}\| + \|p^{(k+1)} - p^{(k)}\| > \sigma$  .

On ne dispose d'une preuve de convergence pour cet algorithme que dans le cas où les fonctions de coût  $K$  et  $J$  sont quadratiques, où les fonctions de contrainte  $\Theta$  et  $\Omega$  sont affines, où les contraintes explicites sont des contraintes d'égalité et où l'ensemble admissible  $U^{\text{ad}}$  est l'espace tout entier. Le théorème de convergence, donné au § 3.3.3.3 du polycopié, fait une hypothèse géométrique concernant les formes respectives des contraintes  $\Theta$  et  $\Omega$ , à savoir :

$$\Theta J^{-1} \Omega^\top + \Omega J^{-1} \Theta^\top > 0 , \quad (75)$$

où, par abus de notation,  $J$ ,  $\Theta$  et  $\Omega$  sont les matrices représentatives des fonctions du même nom. Cette condition provient du fait que le noyau de décomposition  $\Lambda$  n'est pas fortement concave en  $p$ . Un cas dans lequel cette contrainte géométrique est vérifiée est celui où la matrice  $\Theta$  est "presque" bloc-diagonale, dans le sens où les termes hors-diagonaux sont petits par rapport aux termes diagonaux. Alors, prendre pour  $\Omega$  la partie diagonale de  $\Theta$  assure que la condition (75) soit satisfaite. C'est ce qu'on appelle la situation d'*interaction faible* entre les sous-problèmes.

L'utilisation de cet algorithme en décomposition passe par le fait que le calcul d'un point selle du Lagrangien auxiliaire  $\Lambda$  se décompose en  $N$  calculs de points selle, et donc que :

- la fonction  $K$  soit additive par rapport à la décomposition en  $u$  ;
- la fonction  $\Omega$  soit bloc-diagonale par rapport aux décompositions en  $u$  et en  $p$ .

Deux cas se présentent.

1.  $\Theta^\Sigma = 0$  : on n'a alors aucune contrainte a priori sur la manière de décomposer  $p$  ; une fois effectué ce choix de décomposition, on choisit la contrainte auxiliaire  $\Omega$  sous forme bloc-diagonale.

2.  $\Theta^\Sigma \neq 0$  : la décomposition de la variable  $p$  est alors une donnée du problème ; on se contente de choisir la contrainte auxiliaire  $\Omega$  sous forme bloc-diagonale.

À titre d'exemple, étant donnée une décomposition de la variable  $p$  (choisie ou imposée suivant les cas), on dispose directement de la décomposition associée des composantes de la contrainte  $\Theta$  :

$$\Theta(u) = \begin{pmatrix} \Theta_1(u) \\ \vdots \\ \Theta_N(u) \end{pmatrix},$$

et il existe alors un choix canonique de noyau de décomposition, à savoir :

- $K_i(u_i) = J(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i, u_{i+1}^{(k)}, \dots, u_N^{(k)})$ ,
- $\Omega_i(u_i) = \Theta_i(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i, u_{i+1}^{(k)}, \dots, u_N^{(k)})$ .

Avec le choix  $\epsilon = 1$ , le  $i$ -ème sous-problème de calcul de point selle s'écrit :

$$\min_{u_i \in U_i^{\text{ad}}} J(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i, u_{i+1}^{(k)}, \dots, u_N^{(k)}) + J_i^\Sigma(u_i) + \sum_{j \neq i} \left\langle p_j^{(k)}, (\Theta_j)_{u_i}'(u_1^{(k)}, \dots, u_N^{(k)}) \cdot u_i^{(k)} \right\rangle, \quad (76a)$$

sous la contrainte

$$\Theta_i(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i, u_{i+1}^{(k)}, \dots, u_N^{(k)}) + \Theta_i^\Sigma(u_i) \in -C_i. \quad (76b)$$

Dans le cas où la fonction  $J$  et les fonctions  $\Theta_i$  sont additives, on retrouve presque exactement l'algorithme de décomposition par prédiction du § 2.4. L'algorithme 4 en est bien une généralisation au cas où les hypothèses d'additivité sont remplacées par des hypothèses de différentiabilité.



### 3.4 Exercices

#### Application du PPA à un réseau de distribution d'eau

On a vu lors de la première séance que le problème de l'optimisation de  $N$  réseaux de distribution d'eau connectés par une usine de refoulement se mettait, après minimisation par rapport aux variables locales des sous-réseaux, sous une forme dans laquelle le couplage intervient par le critère et non par les contraintes :

$$\min_{\{(u_{i,1}, u_{i,2}) \in U_i^{\text{ad}}\}_{i=1}^N} J_{N+1} \left( \sum_{i=1}^N u_{i,1}, \sum_{i=1}^N u_{i,2} \right) + \sum_{i=1}^N J_i(u_{i,1}, u_{i,2}), \quad (77)$$

où l'appartenance aux ensembles  $U_i^{\text{ad}}$  permet d'assurer que les volumes fournis aux sous-réseaux par l'usine de refoulement sont positifs :

$$U_i^{\text{ad}} = \{(u_{i,1}, u_{i,2}) \in \mathbb{R}^2, 0 \leq u_{i,1} \leq \bar{v}_{i,1} \text{ et } 0 \leq u_{i,2} \leq \bar{v}_{i,2} - \bar{v}_{i,1}\}.$$

Pour simplifier les expressions, on utilisera les notations :  $u_i = (u_{i,1}, u_{i,2})$  et  $u = (u_1, \dots, u_N)$ .

Pour chacun des choix suivants, appliquer le principe du problème auxiliaire au problème (77) et interpréter l'algorithme obtenu.

**Choix No 1 :**

$$\boxed{\begin{aligned} J(u) &= \sum_{i=1}^N J_i(u_i) + J_{N+1} \left( \sum_{i=1}^N u_i \right) & \text{et} & \quad J^\Sigma(u) \equiv 0, \\ K^{(k)}(u) &= \sum_{i=1}^N J_i(u_i) & \text{et} & \quad \epsilon^{(k)} = 1. \end{aligned}}$$

**Choix No 2 :**

$$\boxed{\begin{aligned} J(u) &= \sum_{i=1}^N J_i(u_i) + J_{N+1} \left( \sum_{i=1}^N u_i \right) & \text{et} & \quad J^\Sigma(u) \equiv 0, \\ K^{(k)}(u) &= \sum_{i=1}^N J_{N+1} \left( u_i + \sum_{j \neq i} u_j^{(k)} \right) & \text{et} & \quad \epsilon^{(k)} = 1. \end{aligned}}$$

**Choix No 3 :**

$$\boxed{\begin{aligned} J(u) &= \sum_{i=1}^N J_i(u_i) + J_{N+1} \left( \sum_{i=1}^N u_i \right) & \text{et} & \quad J^\Sigma(u) \equiv 0, \\ K^{(k)}(u) &= \sum_{i=1}^N J_i(u_i) + \sum_{i=1}^N J_{N+1} \left( u_i + \sum_{j \neq i} u_j^{(k)} \right) & \text{et} & \quad \epsilon^{(k)} = 1. \end{aligned}}$$

Choix No 4 :

$$\begin{aligned} J(u) &= \sum_{i=1}^N J_i(u_i) + J_{N+1}\left(\sum_{i=1}^N u_i\right) & \text{et} & \quad J^\Sigma(u) \equiv 0, \\ K^{(k)}(u) &= \sum_{i \neq i_0} J_i(u_i) + J_{N+1}\left(u_{i_0} + \sum_{i \neq i_0} u_i^{(k)}\right) & \text{et} & \quad \epsilon^{(k)} = 1. \end{aligned}$$



### 3.5 Corrigés

#### Application du PPA à un réseau de distribution d'eau

On rappelle que l'application du Principe du Problème Auxiliaire à un problème général de la forme :

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

conduit à résoudre à l'itération  $k$  le problème auxiliaire :

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

**Choix No 1.** Les différences entre les dérivées de  $\epsilon^{(k)}J$  et  $K^{(k)}$  s'annulent sauf pour les termes concernant  $J_{N+1}$ , et la  $k$ -ème itération du **PPA** s'écrit alors :

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

Ce problème auxiliaire se décompose en  $N$  sous-problèmes dont la  $i$ -ème instance est :

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

On retrouve ainsi l'algorithme de décomposition par *prédiction*. En effet, les dérivées partielles de  $J_{N+1}$  par rapport aux  $u_i$  sont toutes égales entre elles car  $J_{N+1}$  ne dépend que de la somme des  $u_i$ . Notant :

$$p_{N+1}^{(k)} = \nabla J_{N+1} \left( \sum_{j=1}^N u_j^{(k)} \right),$$

les  $N$  sous-problèmes correspondant aux  $N$  réseaux élémentaires sont obtenus en ajoutant à leur critère le terme linéaire  $\langle p_{N+1}^{(k)}, u_i \rangle$ , comme dans la méthode de décomposition par les prix. Quant au calcul de  $p_{N+1}^{(k)}$ , il correspond précisément au calcul des multiplicateurs issus de la minimisation du coût de l'usine couplante à production globale fixée, comme dans la méthode de décomposition par allocation. On a donc obtenu l'algorithme de décomposition par prédiction de l'exercice du § 2, dans le cas où les contraintes couplantes sont affectées au  $(N+1)$ -ème sous-problème.

**Choix No 2.** Les différences entre les dérivées de  $\epsilon^{(k)}J$  et  $K^{(k)}$  s'annulent seulement pour les termes concernant  $J_{N+1}$ , et la  $k$ -ème itération du **PPA** s'écrit :

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

Ce problème auxiliaire se décompose en  $N$  sous-problèmes dont la  $i$ -ème instance est :

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

C'est donc un algorithme dual du précédent, dans lequel les fonctions  $J_i$  n'interviennent que sous forme linéarisée pour  $i$  variant de 1 à  $N$ . Le calcul des  $\nabla J_i(u_i^{(k)})$  correspond, au signe près, au calcul des multiplicateurs  $p_{i,t}^{(k)}$  lors de la minimisation de  $J_i$  sous les contraintes  $u_{i,t} - u_{i,t}^{(k)} = 0$ , et les  $N$  réseaux élémentaires sont donc traités comme dans l'algorithme de décomposition par allocation de ressources. La résolution du  $i$ -ème sous-problème auxiliaire conduit à comparer le coût *relaxé*<sup>28</sup> de l'usine couplante au coût, *linéarisé* autour du point  $u_i^{(k)}$ , du  $i$ -ème réseau élémentaire. Si l'on se rappelle que la fonction  $J_{N+1}$  est quadratique :

$$J_{N+1} \left( \sum_{i=1}^N u_i \right) = \frac{1}{2} \sum_{t=1}^2 a_{N+1,t} \left( \sum_{i=1}^N u_{i,t} \right)^2,$$

28. dans le sens où toutes les variables dont dépend  $J_{N+1}$  sont fixées à l'exception de  $u_i$

et si l'on ne tient pas compte de l'ensemble  $U_i^{\text{ad}}$ , les conditions d'optimalité des  $N$  sous-problèmes auxiliaires s'écrivent :

$$-p_{i,t}^{(k)} + a_{N+1,t} \left( u_{i,t} + \sum_{j \neq i} u_{j,t}^{(k)} \right) = 0 ,$$

soit :

$$u_{i,t}^{(k+1)} = \frac{p_{i,t}^{(k)}}{a_{N+1,t}} - \sum_{j \neq i} u_{j,t}^{(k)} .$$

Cette équation se met sous la forme :

$$\begin{aligned} u_{i,t}^{(k+1)} &= u_{i,t}^{(k)} + \frac{1}{a_{N+1,t}} \left( p_{i,t}^{(k)} - a_{N+1,t} \sum_{j=1}^N u_{j,t}^{(k)} \right) \\ &= u_{i,t}^{(k)} + \frac{1}{a_{N+1,t}} \left( p_{i,t}^{(k)} - p_{N+1,t}^{(k)} \right) . \end{aligned}$$

Cette formule ressemble à celle de l'étape de remise à jour dans la méthode de décomposition par allocation. Elle correspond cependant à une minimisation complète et non à une étape de gradient projeté.

**Choix No 3.** Les différences entre les dérivées de  $\epsilon^{(k)}J$  et  $K^{(k)}$  s'annulent, et la  $k$ -ème itération du **PPA** s'écrit :

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

Ce problème auxiliaire se décompose en  $N$  sous-problèmes dont la  $i$ -ème instance est :

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

Cet algorithme ressemble à celui correspondant au choix No 1, mais on utilise alors dans chacun des  $N$  sous-problèmes une version *relaxée* de la fonction  $J_{N+1}$  plutôt qu'une version *linéarisée*. Cet algorithme a toute les chances de mieux se comporter que le précédent, car chaque sous-problème a alors une vision moins locale du coût  $J_{N+1}$  de l'usine couplante que dans l'algorithme de prédiction correspondant au choix No 1.

**Choix No 4.** Avec ce choix de noyaux, on obtient les sous-problèmes auxiliaires suivants :

$$\begin{aligned} i \neq i_0 &: \min_{u_i \in U_i^{\text{ad}}} J_i(u_i) + \left\langle \nabla J_{N+1} \left( \sum_{j=1}^N u_j^{(k)} \right), u_i \right\rangle , \\ i = i_0 &: \min_{u_{i_0} \in U_{i_0}^{\text{ad}}} J_{N+1} \left( u_{i_0} + \sum_{i \neq i_0} u_i^{(k)} \right) + \left\langle \nabla J_{i_0}(u_{i_0}^{(k)}), u_{i_0} \right\rangle . \end{aligned}$$

La résolution des sous-problèmes auxiliaires  $i \neq i_0$  se fait comme dans la méthode de décomposition par les prix. Le calcul de  $\nabla J_{i_0}(u_{i_0}^{(k)})$  correspond à la minimisation de  $J_{i_0}$  sous la contrainte  $u_{i_0} - u_{i_0}^{(k)} = 0$  et se fait donc comme dans la méthode de décomposition par allocation. Enfin, le sous-problème auxiliaire  $i_0$ , où l'on compare le coût relaxé de l'usine couplante au coût linéarisé du réseau élémentaire  $i_0$  permet la remise à jour des variables  $u_{i_0}$ .

On retrouve ainsi la variante de l'algorithme de décomposition par prédiction du § 2 que l'on avait introduite pour traiter le cas où le réseau élémentaire  $i_0$  était suffisamment complexe pour qu'on ne veuille pas le traiter comme dans la méthode par les prix.

## 4 Régularisation et Lagrangien augmenté

On va brièvement introduire une notion de **régularisation** particulièrement adaptée à l'optimisation et on montrera l'intérêt que présente cette notion dans le cadre de la dualité.

### 4.1 Régularisée de Moreau-Yosida

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

$$J_c(u) = \min_{v \in \mathcal{U}} \left( J(v) + \frac{1}{2c} \|v - u\|^2 \right). \quad (78)$$

Prenant  $v = 0$  dans (78), on déduit immédiatement de cette définition que l'on a :

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

A titre d'exemple, la régularisée de Moreau-Yosida de la fonction "valeur absolue"  $J(u) = |u|$  est la fonction :<sup>29</sup>

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

que l'on a représentée Figure 9 (pour  $c = 0.5$ ) :

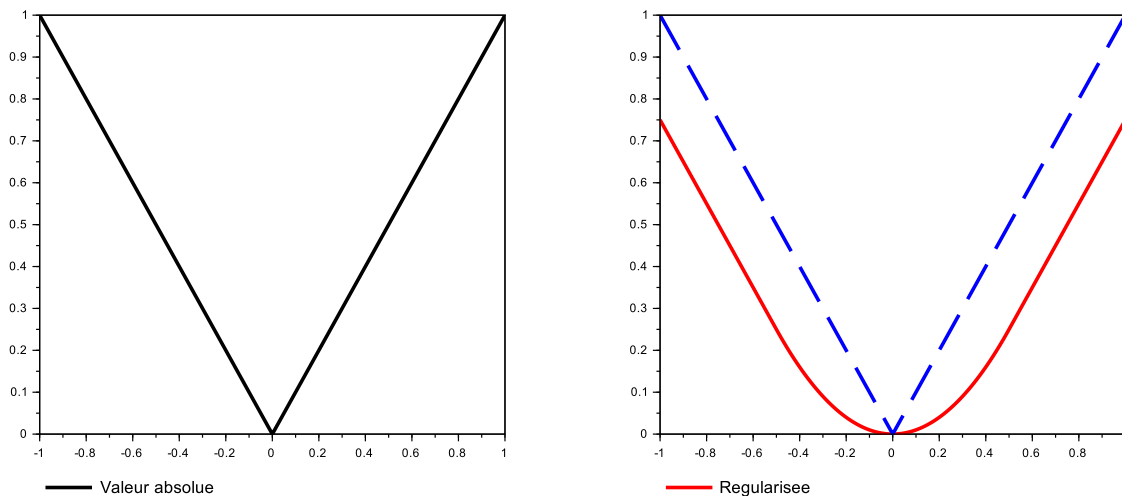


FIGURE 9 – Régularisation de la fonction "valeur absolue"

L'intérêt théorique de la régularisée de Moreau-Yosida provient essentiellement de ses propriétés, résumées dans le théorème suivant.

**Théorème 4.** Soit  $J : \mathcal{U} \rightarrow \mathbb{R}$  une fonction propre, convexe, semi-continue inférieurement (s.c.i.) et sous-différentiable. Alors :

1.  $\arg \min_{u \in \mathcal{U}} J(u) = \arg \min_{u \in \mathcal{U}} J_c(u)$ , et les deux fonctions  $J$  et  $J_c$  **coïncident** sur cet ensemble ;
2. la fonction  $J_c$  est propre, convexe, s.c.i., **différentiable à gradient lipschitzien**.

De plus, si  $J$  est fortement convexe, alors  $J_c$  est **aussi** fortement convexe.

<sup>29</sup>. l'écriture et la résolution des conditions d'optimalité (cas sous-différentiable) sont laissées au soin du lecteur...

La première propriété montre qu’il est équivalent de minimiser la fonction  $J$  ou la fonction  $J_c$ . La seconde propriété indique qu’il est plus efficace de mettre en œuvre un algorithme de minimisation sur  $J_c$  que sur  $J$ , dans la mesure où la fonction  $J_c$  est *plus régulière* que la fonction  $J$  (d’où le nom de “régularisée”). La régularisée de Moreau-Yosida présente donc un intérêt conceptuel en optimisation, intérêt que nous allons détailler maintenant.

## 4.2 Application à la minimisation

On considère le problème d’optimisation suivant, dans lequel la fonction  $J$  est convexe s.c.i. propre sous-différentiable, et où les contraintes sont représentées par un ensemble admissible  $U^{\text{ad}} \subset \mathcal{U}$ , que l’on suppose être convexe et fermé :

$$\min_{u \in U^{\text{ad}}} J(u) . \quad (79)$$

Le problème (79) s’écrit sous la forme équivalente suivante :

$$\min_{u \in \mathcal{U}} J(u) + \chi_{U^{\text{ad}}}(u) , \quad (80)$$

où  $\chi_{U^{\text{ad}}}$  est la *fonction caractéristique* de l’ensemble  $U^{\text{ad}}$  :

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

On notera que le fait que l’ensemble  $U^{\text{ad}}$  soit convexe fermé implique que la fonction  $\chi_{U^{\text{ad}}}$  est elle-même convexe s.c.i..

On effectue dorénavant un **changement de notation**, et on appelle  $J_c$  la régularisée de Moreau-Yosida de la fonction  $J + \chi_{U^{\text{ad}}}$  (plutôt que celle de  $J$ ). Il s’en suit que l’on a :

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

Du théorème 4, on déduit que :

$$\arg \min_{u \in U^{\text{ad}}} J(u) = \arg \min_{u \in \mathcal{U}} J_c(u) .$$

**Remarque 14.** La “disparition” de l’ensemble  $U^{\text{ad}}$  dans le terme de droite de l’égalité ci-dessus n’est bien sûr pas “miraculeuse” et provient de ce que la *définition* de  $J_c$  fait intervenir  $U^{\text{ad}}$ .

Plutôt que le problème (79), on est donc tenté de résoudre le problème équivalent suivant :

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

puisque ce nouveau problème consiste à minimiser une fonction *différentiable*, et n’est soumis à *aucune contrainte*. Il y a là bien sûr une difficulté<sup>30</sup> car chaque *évaluation* de la fonction  $J_c$  requiert la *résolution d’un problème d’optimisation* de même nature que le problème initial (79)!!! On va voir dans le paragraphe suivant que l’idée de régularisation conduit à une méthode parfaitement opérationnelle lorsqu’on l’applique au problème dual d’un problème d’optimisation sous contrainte.

**Remarque 15.** Il faut savoir que l’idée de régularisation appliquée de manière directe au problème primal (79) a donné lieu à tout un courant à la fois théorique et algorithmique en optimisation, désigné dans la littérature par les termes “méthode proximale”, “algorithme prox” ...

30. tout au moins si l’on utilise cette idée de régularisation de manière directe sur la fonction  $J$

### 4.3 Mise en œuvre sur le problème dual

On s'intéresse désormais au problème de minimisation suivant (dit "avec contrainte explicite") :

$$\min_{u \in U^{\text{ad}}} J(u) \quad \text{sous} \quad \Theta(u) \in -C, \quad (81)$$

où  $C$  est un cône convexe fermé d'un espace de Hilbert  $\mathcal{C}$ .

On fait pour le problème (81) les hypothèses "classiques" de convexité, coercivité, continuité et différentiabilité pour  $J$  et  $\Theta$ , ainsi qu'une condition de *qualification des contraintes*, par exemple :

$$0 \in \text{int}(\Theta(U^{\text{ad}}) + C) .$$

Le *Lagrangien* associé à ce problème est la fonction  $L$ , définie sur  $U^{\text{ad}} \times C^*$  à valeurs dans  $\mathbb{R}$  :

$$L(u, p) = J(u) + \langle p, \Theta(u) \rangle ,$$

où  $C^*$  est le cône dual de  $C$ . L'ensemble des points selle de ce Lagrangien est noté  $U^\# \times P^\#$ . Puis, on définit à partir du Lagrangien la *fonction duale*  $H$  définie sur  $C^*$  à valeurs dans  $\mathbb{R}$  par :

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

Il est facile de montrer que la fonction duale  $H$  est concave, semi-continue supérieurement (s.c.s.) et sur-différentiable. On note  $\widehat{U}(p)$  l'ensemble des valeurs de  $u$  qui réalisent le minimum de la fonction duale  $H$  à  $p$  donné. Le *problème dual* du problème (81) est :

$$\max_{p \in C^*} H(p) . \quad (82)$$

L'ensemble des solutions de ce problème est  $P^\#$ . Pour  $p^\# \in P^\#$ , on a toujours :

$$U^\# \subset \widehat{U}(p^\#) .$$

Une difficulté majeure est que l'inclusion ci-dessus peut dans certains cas être *stricte*, ce qui signifie que l'ensemble  $\widehat{U}(p^\#)$  contient des solutions "parasites" n'ayant rien à voir avec les solutions du problème d'optimisation initial (81). Cette difficulté est répertoriée dans la littérature sous le nom de *stabilité du Lagrangien*.

#### 4.3.1 Régularisation de la fonction duale et Lagrangien augmenté

L'idée que l'on va mettre en application est de régulariser la fonction duale  $H$  plutôt que la fonction coût  $J$  du problème initial. La fonction duale  $H$  est a priori simplement sur-différentiable, et sa régularisée sera donc différentiable à gradient lipschitzien. Adaptant au cas de la maximisation ce qui a été vu §4.2, on écrit un problème d'optimisation équivalent au problème (82) en régularisant la fonction  $H - \chi_{C^*}$  :

$$\begin{aligned} H_c(p) &= \max_{q \in \mathcal{C}} \left( H(q) - \chi_{C^*}(q) - \frac{1}{2c} \|q - p\|^2 \right) \\ &= \max_{q \in C^*} \left( H(q) - \frac{1}{2c} \|q - p\|^2 \right) . \end{aligned}$$

D'après le théorème 4, on sait que :

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

Remplaçant  $H$  par son expression résultant de la minimisation du Lagrangien  $L$ , et dans la mesure où les opérateurs min et max peuvent être inversés sans difficulté particulière puisque le terme quadratique de régularisation assure le caractère fortement concave en  $q$ , on obtient :

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

On définit alors le *Lagrangien augmenté* comme étant la fonction  $L_c : U^{\text{ad}} \times \mathcal{C} \rightarrow \mathbb{R}$ , ayant pour expression :

$$L_c(u, p) = J(u) + \max_{q \in C^*} \left( \langle q, \Theta(u) \rangle - \frac{1}{2c} \|p - q\|^2 \right) . \quad (83)$$

De cette définition, on déduit que la régularisée de la fonction duale  $H$  vérifie la relation :

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

**Remarque 16.** Il est important de noter que  $L$  et  $L_c$  n'ont pas les mêmes ensembles de définition : alors que le Lagrangien ordinaire  $L$  est défini sur  $U^{\text{ad}} \times C^*$ , où  $C^*$  est le cône dual du cône  $C$  entrant dans la définition de la contrainte, le Lagrangien augmenté  $L_c$  est défini sur  $U^{\text{ad}} \times \mathcal{C}$ , où  $\mathcal{C}$  est l'espace d'arrivée tout entier de la fonction  $\Theta$ .

**Propriétés théoriques du Lagrangien augmenté.** L'intérêt théorique du Lagrangien augmenté provient essentiellement des deux propriétés suivantes (données ici sans démonstration).

1. Le Lagrangien et le Lagrangien augmenté admettent le *même ensemble de points selle*  $U^\# \times P^\#$ .
2. Le Lagrangien augmenté  $L_c$  est *stable* :  $\forall p^\# \in P^\#, \arg \min_{u \in U^{\text{ad}}} L_c(u, p^\#) = U^\#$ .

La première propriété indique que, dans le cadre de la dualité, la solution du problème (81) peut être obtenue en recherchant les points selle du Lagrangien augmenté. Cette recherche se fait classiquement par un algorithme de type Uzawa, avec l'avantage ici que, une fois obtenue une variable duale optimale  $p^\#$  (pas forcément unique), l'ensemble  $\arg \min_{u \in U^{\text{ad}}} L_c(u, p^\#)$  ne contient aucune solution "parasite" (seconde propriété).<sup>31</sup> Enfin, comme la fonction  $H_c$  est différentiable à gradient lipschitzien, sa maximisation, et donc l'obtention d'une valeur optimale  $p^\#$  de la variable duale, peut être effectuée par des algorithmes performants.

**Mise en œuvre opérationnelle du Lagrangien augmenté.** Dans cette approche duale, la bonne nouvelle est que, d'un point de vue pratique et contrairement au cas du problème de minimisation considéré au §4.2, il est possible de calculer *de façon explicite* le Lagrangien augmenté.

On considère pour commencer le cas des contraintes de type égalité, c'est-à-dire le cas  $C = \{0\}$ . Le cône dual est alors égal à l'espace tout entier :  $C^* = \mathcal{C}$ , de telle sorte que l'opération de maximisation figurant dans l'expression du Lagrangien augmenté s'écrit :

$$\max_{q \in \mathcal{C}} \langle q, \Theta(u) \rangle - \frac{1}{2c} \|p - q\|^2 .$$

Ceci correspond à l'optimisation d'une *forme quadratique* dont la solution est :

$$q^\# = p + c \Theta(u) .$$

---

31. On notera en particulier qu'il n'est pas nécessaire de supposer la fonction  $J$  fortement convexe (comme c'était le cas dans le cadre du Lagrangien ordinaire), de telle sorte que le Lagrangien augmenté permet d'obtenir la solution de problème d'optimisation tout ou partiellement linéaire.

Incorporant cette valeur dans (83), on obtient pour le Lagrangien augmenté l'expression suivante :

$$L_c(u, p) = J(u) + \langle p + c\Theta(u), \Theta(u) \rangle - \frac{1}{2c} \|c\Theta(u)\|^2 ,$$

et donc finalement :

$$L_c(u, p) = J(u) + \langle p, \Theta(u) \rangle + \frac{c}{2} \|\Theta(u)\|^2 . \quad (84)$$

Dans cette expression, les deux premiers termes correspondent au Lagrangien ordinaire. Le dernier terme en  $\|\Theta(u)\|^2$  est souvent interprété comme un terme de *pénalisation* dans la mesure où la contrainte traitée ici est de la forme  $\Theta(u) = 0$ . **Cette interprétation est erronée** : comme on l'a vu tout au long de ce chapitre, l'idée qui sous-tend le Lagrangien augmenté est celle de *régularisation* et non celle de pénalisation. On verra dans le cas des contraintes de type inégalité que l'expression obtenue pour le Lagrangien augmenté ne contient pas de terme de type pénalisation.

**Remarque 17.** On remarquera que, même dans le cas où les fonctions  $J$  et  $\Theta$  sont *additives* par rapport à une décomposition de l'espace  $\mathcal{U}$  :

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

le Lagrangien augmenté n'est pas additif par rapport à cette décomposition car le terme en  $\|\Theta(u)\|^2$  induit des termes de couplage de la forme  $\langle \Theta_i(u_i), \Theta_j(u_j) \rangle$ .

On passe maintenant au cas où  $C$  est un cône convexe fermé quelconque. Alors, l'opération de maximisation figurant dans l'expression du Lagrangien augmenté :

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

se met sous la forme équivalente :

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

les deux derniers termes ne dépendant pas de la variable  $q$ . La solution de ce problème quadratique (en fait sphérique) s'obtient à l'aide de la projection sur le cône dual  $C^*$  :

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

Avec cette valeur de  $q^\#$ , on a que :

$$\begin{aligned} \langle q^\#, \Theta(u) \rangle - \frac{1}{2c} \|p - q^\#\|^2 &= -\frac{1}{2c} \|p\|^2 - \frac{1}{2c} \|q^\#\|^2 + \frac{1}{c} \langle q^\#, p + c\Theta(u) \rangle \\ &= -\frac{1}{2c} \|p\|^2 + \frac{1}{2c} \|q^\#\|^2 , \end{aligned}$$

la dernière égalité provenant du fait que la projection orthogonale vérifie la propriété :

$$\langle \text{proj}_{C^*}(p + c\Theta(u)), p + c\Theta(u) \rangle = \langle \text{proj}_{C^*}(p + c\Theta(u)), \text{proj}_{C^*}(p + c\Theta(u)) \rangle .$$

On en déduit l'expression du Lagrangien augmenté dans le cas général :

$$L_c(u, p) = J(u) + \frac{1}{2c} \left( \|\text{proj}_{C^*}(p + c\Theta(u))\|^2 - \|p\|^2 \right) . \quad (85)$$

Moyennant une opération de projection sur le cône duale  $C^*$  (souvent facile à mettre en œuvre), le calcul de ce Lagrangien augmenté est donc effectif.

**Remarque 18.** On notera que l'expression (85) n'a pas grand-chose à voir avec un Lagrangien auquel on aurait ajouté un terme pénalisant les écarts à la contrainte, c'est-à-dire une expression de la forme :<sup>32</sup>

$$J(u) + \langle p, \Theta(u) \rangle + \frac{c}{2} \|\text{proj}_{C^*}(\Theta(u))\|^2 .$$

Pour conclure sur le côté opérationnel du Lagrangien augmenté, on va calculer ses gradients partiels par rapport à  $u$  et  $p$ . Pour cela, on introduit l'application  $\zeta_c : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}$ , définie par :

$$\begin{aligned} \zeta_c(p, \theta) &= \max_{q \in C^*} \langle q, \theta \rangle - \frac{1}{2c} \|p - q\|^2 \\ &= \frac{1}{2c} \left( \|\text{proj}_{C^*}(p + c\theta)\|^2 - \|p\|^2 \right) . \end{aligned}$$

On a bien sûr :

$$L_c(u, p) = J(u) + \zeta_c(p, \Theta(u)) . \quad (86)$$

Par un argument de dérivation des fonctions marginales (c'est à dire des fonctions résultant d'une optimisation : voir §1.3.4), on obtient les gradients partiels de la fonction  $\zeta_c$  :

$$\begin{aligned} \nabla_p \zeta_c(p, \theta) &= \frac{1}{c} \left( \text{proj}_{C^*}(p + c\theta) - p \right) \\ \nabla_\theta \zeta_c(p, \theta) &= \text{proj}_{C^*}(p + c\theta) , \end{aligned}$$

d'où l'on déduit à l'aide de (86) les gradients partiels du Lagrangien augmenté  $L_c$ . On a alors tous les ingrédients permettant de mettre en œuvre un algorithme numérique permettant de calculer un point selle du Lagrangien augmenté. À titre d'exemple, une itération de l'algorithme de Uzawa s'écrit :

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

**Décomposition du Lagrangien augmenté.** On a déjà remarqué que le Lagrangien augmenté ne conservait pas les éventuelles propriétés d'additivité du problème d'optimisation initial, et qu'on ne pouvait donc pas, même dans le cas simple  $C = \{0\}$ , lui appliquer l'algorithme de décomposition par les prix. D'un autre côté, on a vu que le principe du problème auxiliaire permettait une généralisation de la méthode de décomposition par les prix dans le cas où le Lagrangien n'est pas additif. On peut montrer que cette généralisation s'applique au cas du Lagrangien augmenté. Pour plus de détails, on consultera le §5.4 du polycopié du cours.

### 4.3.2 Interprétation géométrique

On rappelle la définition de la *fonction de perturbation*  $\Phi : \mathcal{C} \rightarrow \mathbb{R}$  :

$$\Phi(v) = \left\{ \min_{u \in U^{\text{ad}}} J(u) \quad \text{sous} \quad \Theta(u) - v \in -C \right\} ,$$

---

32. le terme  $\text{proj}_{C^*}(\Theta(u))$  est nul lorsque  $\Theta(u) \in -C$



qui correspond au coût optimal du problème dans lequel on a introduit une perturbation  $v \in \mathcal{C}$  de la contrainte. Un résultat fondamental de la théorie de la dualité est l'*interprétation marginaliste* du multiplicateur  $p^\sharp$  des conditions de Karush–Kuhn–Tucker, à savoir :

$$-p^\sharp \in \partial\Phi(0) .$$

Dans ce paragraphe, on se limite au cas des contraintes de type égalité  $\Theta(u) = 0$  afin de rendre plus simple l'interprétation des différentes méthodes. On a déjà vu (second exercice du §1.5 de ces notes de cours) que l'algorithme de Uzawa appliqué au Lagrangien ordinaire :

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

s'interprétait à l'aide de la fonction duale de la façon suivante :

- $v^{(k+1)} \in \arg \min_{v \in \mathcal{C}} \Phi(v) + \langle p^{(k)}, v \rangle ,$
- $p^{(k+1)} = p^{(k)} + \rho v^{(k+1)} ,$

ce qui d'un point de vue géométrique revient à ausculter la fonction  $\Phi$  avec des droites de pente  $-p^{(k)}$  jusqu'à trouver la droite de pente  $-p^\sharp$  qui tangente la fonction  $\Phi$  au point  $v^\sharp = 0$ . On a montré que, dans le cas où la fonction  $\Phi$  n'était pas fortement convexe, voir non convexe, cet algorithme ne permettait pas d'obtenir la solution du problème d'optimisation initial.

Ce même algorithme de Uzawa appliqué au Lagrangien augmenté s'écrit :

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

et son interprétation à l'aide de la fonction duale est en conséquence :

- $v^{(k+1)} \in \arg \min_{v \in \mathcal{C}} \Phi(v) + \langle p^{(k)}, v \rangle + \frac{c}{2} \|v\|^2 ,$
- $p^{(k+1)} = p^{(k)} + \rho v^{(k+1)} .$

D'un point de vue géométrique, ce nouvel algorithme revient à ausculter la fonction  $\Phi$  avec des paraboles de la forme  $-\langle p^{(k)}, \cdot \rangle - \frac{c}{2} \|\cdot\|^2$ , jusqu'à trouver celle qui tangente  $\Phi$  au point  $v^\sharp = 0$ . On comprend géométriquement que cet algorithme va fonctionner même dans le cas où la fonction  $\Phi$  est linéaire ou non convexe (pourvu que le coefficient  $c$  soit suffisamment grand pour que l'on puisse "loger la parabole dans la non-convexité"). On a représenté à la figure 10 les comportements respectifs de l'algorithme de Uzawa appliqué au Lagrangien ordinaire et au Lagrangien augmenté dans deux cas où l'hypothèse de forte convexité fait défaut.

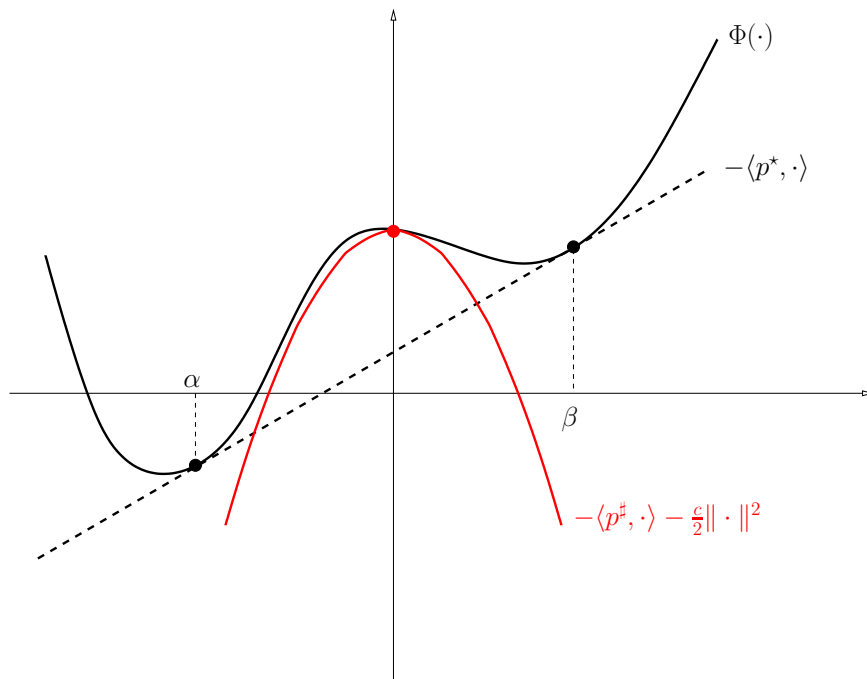
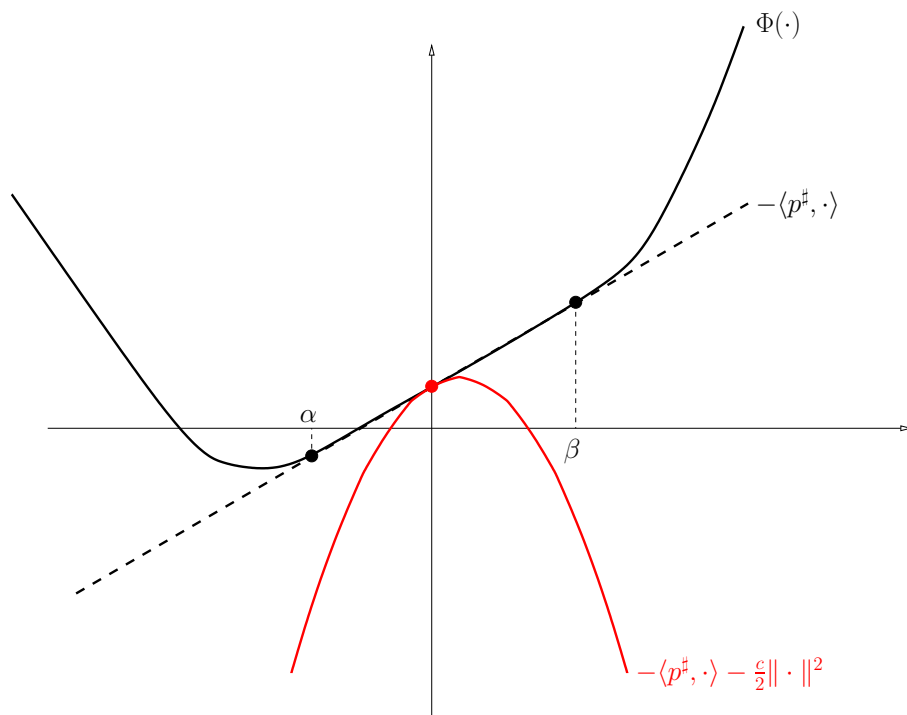


FIGURE 10 – Interprétation géométrique du Lagrangien augmenté