

11 Principe du problème auxiliaire

On introduit 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 de simplifier l'étude de la convergence des algorithmes ;
- le **PPA** permet de décomposer les problèmes sans hypothèses d'additivité ;
- le **PPA** permet d'améliorer le conditionnement des problèmes.

On commence par exposer ce principe dans le cas sans contrainte explicite. On traite dans une deuxième partie le cas général. Ce chapitre ne constitue qu'une brève présentation du principe du problème auxiliaire. On en trouvera une présentation complète dans [10].

11.1 Optimisation sur un ensemble admissible

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

$$\min_{u \in U^{\text{ad}} \subset \mathbb{U}} J(u) + J^\Sigma(u) . \quad (100)$$

L'écriture de la fonction coût comme la somme des deux termes J et J^Σ correspond au fait que l'on considère que la partie du problème correspondant au terme J^Σ est "facile" à résoudre. Ainsi, lorsqu'il sera question de décomposer le problème (100), et que l'espace \mathbb{U} sera mis sous la forme d'un produit cartésien de N sous-espaces \mathbb{U}_i tels que l'ensemble U^{ad} soit aussi de la forme $U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}$ avec $U_i^{\text{ad}} \subset \mathbb{U}_i$, on supposera que la fonction J^Σ est additive par rapport à la décomposition de l'espace \mathbb{U} :⁴²

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

11.1.1 Principe de la méthode

Quand on veut décomposer le problème (100), seul le terme $J(u)$ est gênant car il n'est a priori pas additif par rapport à la décomposition de l'espace \mathbb{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 $u^{(k)}$:

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

En fait, le terme $J(u^{(k)}) - \langle J'(u^{(k)}), u^{(k)} \rangle$ étant constant, cette approximation revient à remplacer dans (100) le terme $J(u)$ par le terme $\langle J'(u^{(k)}), u \rangle$, qui est linéaire en u . On constate alors que :

- ce nouveau terme linéaire est additif par rapport à n'importe quelle décomposition de l'espace \mathbb{U} ,
- les conditions d'optimalité du problème résultant de cette approximation sont exactement celles du problème initial si l'on se place au point $u^{(k)} = u^\#$.

⁴²En effet, en l'absence du terme J , l'hypothèse d'additivité sur J^Σ fait que le problème (100) se décompose en N sous-problèmes *indépendants*, le i -ème de ces sous-problèmes étant : $\min_{u_i \in U_i^{\text{ad}}} J_i^\Sigma(u_i)$.

Cependant, cette linéarisation peut faire disparaître le caractère coercif de la fonction coût.

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 terme soit nul en $u^{(k)}$. Pour cela, on choisit une famille de fonctions $K^{(k)}$, appelées *noyaux de décomposition*. On suppose que ces fonctions sont fortement convexes et différentiables, et on ajoute au critère le terme :

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

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 :

$$\min_{u \in U^{\text{ad}}} K^{(k)}(u) + \left\langle \epsilon^{(k)} J'(u^{(k)}) - K^{(k)'}(u^{(k)}), u \right\rangle + \epsilon^{(k)} J^\Sigma(u). \quad (101)$$

Ce problème est appelé *problème auxiliaire au point $u^{(k)}$* associé au problème (100). Sa solution⁴³ est notée $u^{(k+1)}$, et les conditions d'optimalité associées s'écrivent :

$$\forall u \in U^{\text{ad}}, \left\langle K^{(k)'}(u^{(k+1)}) - K^{(k)'}(u^{(k)}) + \epsilon^{(k)} J'(u^{(k)}), u - u^{(k+1)} \right\rangle + \epsilon^{(k)} \left(J^\Sigma(u) - J^\Sigma(u^{(k+1)}) \right) \geq 0. \quad (102)$$

Partant d'un point quelconque $u^{(0)} \in U^{\text{ad}}$, on peut construire la suite $\{u^{(k)}\}_{k \in \mathbb{N}}$ engendrée par l'enchaînement des résolutions des problèmes auxiliaires (101). 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 (102) se simplifient en :

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

On retrouve au facteur $\epsilon^{(k)}$ près les conditions d'optimalité du problème initial (100). Ceci suggère, pour résoudre (100), la mise en œuvre de l'algorithme suivant.

Algorithme 7.

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 σ ; poser $k = 0$.
2. Calculer $u^{(k+1)} = \underset{u \in U^{\text{ad}}}{\operatorname{argmin}} \left(K^{(k)}(u) + \left\langle \epsilon^{(k)} J'(u^{(k)}) - K^{(k)'}(u^{(k)}), u \right\rangle + \epsilon^{(k)} J^\Sigma(u) \right)$.
3. Retourner à l'étape 2 en incrémentant k de 1 tant que $\|u^{(k+1)} - u^{(k)}\| > \sigma$.

11.1.2 Intérêts du PPA

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 $J^\Sigma \equiv 0$,

⁴³unique car on a supposé que les fonctions $K^{(k)}$ étaient fortement convexes

- le choix d'un noyau de décomposition purement quadratique $K(u) = \frac{1}{2} \|u\|^2$ conduit à un problème auxiliaire dont la solution se calcule analytiquement et correspond à l'algorithme de gradient projeté :

$$u^{(k+1)} = \text{proj}_{U^{\text{ad}}} \left(u^{(k)} - \epsilon^{(k)} J'(u^{(k)}) \right) .$$

- Le choix d'une famille de noyaux de décomposition approximant J au second ordre, à savoir $K^{(k)}(u) = \frac{1}{2} \langle u, J''(u^{(k)}) . u \rangle$, conduit au problème auxiliaire suivant :

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

dont la solution se calcule de manière analytique dans le cas où $U^{\text{ad}} = \mathbb{U}$:⁴⁴

$$u^{(k+1)} = u^{(k)} - \epsilon^{(k)} [J''(u^{(k)})]^{-1} . J'(u^{(k)}) ,$$

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 7 dans ce cas.

Décomposition. Le second intérêt du Principe du Problème Auxiliaire concerne la décomposition. On se place dans le cas où l'on peut mettre l'espace \mathbb{U} sous forme d'un produit cartésien $\mathbb{U}_1 \times \dots \times \mathbb{U}_N$ tel que les deux propriétés suivantes soient vérifiées :

- l'ensemble admissible U^{ad} est un produit cartésien : $U^{\text{ad}} = U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}$, avec $U_i^{\text{ad}} \subset \mathbb{U}_i$,
- la fonction J^Σ est additive par rapport à la décomposition de \mathbb{U} : $J^\Sigma(u_1, \dots, u_N) = \sum_{i=1}^N J_i^\Sigma(u_i)$.

Si l'on choisit une suite de noyaux de décomposition eux aussi additifs :

$$K^{(k)}(u_1, \dots, u_N) = \sum_{i=1}^N K_i^{(k)}(u_i) ,$$

le k -ème problème auxiliaire se met sous la forme :

$$\min_{(u_1, \dots, u_N) \in U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}} \sum_{i=1}^N \left(K_i^{(k)}(u_i) + \left\langle \epsilon^{(k)} J'_{u_i}(u_1^{(k)}, \dots, u_N^{(k)}) - K_i^{(k)'}(u_i^{(k)}), u_i \right\rangle + \epsilon^{(k)} J_i^\Sigma(u_i) \right) ,$$

$J'_{u_i}(u_1^{(k)}, \dots, u_N^{(k)})$ désignant le gradient partiel de J par rapport à u_i , évalué au point $(u_1^{(k)}, \dots, u_N^{(k)})$. Ce problème auxiliaire se scinde en N sous-problèmes indépendants, le i -ème d'entre eux étant :

$$\min_{u_i \in U_i^{\text{ad}}} K_i^{(k)}(u_i) + \left\langle \epsilon^{(k)} J'_{u_i}(u_1^{(k)}, \dots, u_N^{(k)}) - K_i^{(k)'}(u_i^{(k)}), u_i \right\rangle + \epsilon^{(k)} J_i^\Sigma(u_i) .$$

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 calculateur parallèle de type MIMD (Multiple Instructions Multiple Data), on préférera utiliser une version *séquentielle* (dite de type *Gauss-Seidel*) de l'algorithme dans laquelle les solutions $u_j^{(k+1)}$ des j -èmes sous-problèmes à l'itération k (avec $j < i$) servent à formuler le i -ème sous-problème à la même itération :

$$\min_{u_i \in U_i^{\text{ad}}} K_i^{(k)}(u_i) + \left\langle \epsilon^{(k)} J'_{u_i}(u_1^{(k+1)}, \dots, u_{i-1}^{(k+1)}, u_i^{(k)}, \dots, u_N^{(k)}) - K_i^{(k)'}(u_i^{(k)}), u_i \right\rangle + \epsilon^{(k)} J_i^\Sigma(u_i) .$$

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.

⁴⁴Dans le cas où U^{ad} est différent de \mathbb{U} , on ne peut plus calculer la solution analytique du problème auxiliaire (la projection de la solution sans contraintes n'étant pas identique à la solution avec contraintes).

Conditionnement. Le dernier intérêt que l'on mentionne 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 :

$$\min_{u \in \mathbb{U}} J(u) ,$$

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^\sharp .⁴⁵ Alors, le conditionnement de la matrice hessienne $J''(u)$ se détériore au voisinage de u^\sharp , et l'itération de l'algorithme de Newton "explose" lorsque $u^{(k)}$ tend vers u^\sharp .

Choisissant un scalaire $\delta > 0$, on applique le Principe du Problème Auxiliaire avec les noyaux :

$$\frac{\delta}{2} \|u\|^2 + \frac{1}{2} \langle u, J''(u^{(k)}) \cdot u \rangle ,$$

et avec $\epsilon^{(k)} = 1$. Le problème auxiliaire au point $u^{(k)}$ est quadratique, sa matrice hessienne est égale à $(\delta I + J''(u^{(k)}))$ et elle est partout définie positive dès que $J''(u^{(k)})$ est elle-même semi-définie positive. Le problème auxiliaire se résout alors de manière explicite et sa solution est :

$$u^{(k+1)} = u^{(k)} - [\delta I + J''(u^{(k)})]^{-1} \cdot J'(u^{(k)}) ,$$

que l'on peut interpréter comme l'itération d'une méthode de Newton "avec reconditionnement". On notera que le choix du coefficient δ résulte d'un compromis entre, d'une part le respect des conditions de convergence de l'algorithme (coercivité : δ doit être "grand"), et d'autre part le ralentissement induit par le terme quadratique dans le noyau (freinage : δ doit être "petit").

11.1.3 Théorème de convergence

Le théorème de convergence suivant est énoncé et démontré dans [10].

Théorème 13.

On suppose que les hypothèses suivantes sont vérifiées :

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

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

⁴⁵C'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 devient nul.

Alors, on a les résultats suivants :

- R1** le problème initial (100) admet au moins une solution u^\sharp , et chaque problème auxiliaire (101) 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 (100).

Si on fait de plus l'hypothèse :

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

alors on obtient que :

- R4** la suite $\{u^{(k)}\}_{k \in \mathbb{N}}$ converge vers l'unique solution u^\sharp du problème (100) ;
- R5** on dispose de la majoration a posteriori d'erreur :

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

11.2 Optimisation sous contraintes explicites

On se place maintenant dans le cadre de l'optimisation sous contraintes explicites, et on s'intéresse au problème suivant :

$$\min_{u \in U^{\text{ad}} \subset \mathbb{U}} J(u) + J^\Sigma(u) . \quad (103a)$$

sous les contraintes :

$$\Theta(u) + \Theta^\Sigma(u) \in -C \subset \mathbb{V} . \quad (103b)$$

Le lagrangien de ce problème est :

$$J(u) + J^\Sigma(u) + \langle p, \Theta(u) + \Theta^\Sigma(u) \rangle , \quad (104)$$

qui se présente comme la somme de deux lagrangiens L et L^Σ , à savoir :

$$L(u, p) = J(u) + \langle p, \Theta(u) \rangle \quad \text{et} \quad L^\Sigma(u, p) = J^\Sigma(u) + \langle p, \Theta^\Sigma(u) \rangle . \quad (105)$$

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 (103).

L'écriture des fonctions de coût, de contraintes et du lagrangien comme la somme de deux termes correspond au fait que l'on considère que les parties J^Σ et Θ^Σ (et donc le lagrangien L^Σ) "ne posent pas de difficulté" en optimisation. Ainsi, quand on se préoccupera de décomposition, on supposera comme au § 11.1 que l'espace \mathbb{U} admet une décomposition $\mathbb{U}_1 \times \dots \times \mathbb{U}_N$ telle que l'ensemble U^{ad} se mette sous la forme d'un produit cartésien $U_1^{\text{ad}} \times \dots \times U_N^{\text{ad}}$. On verra aux § 11.2.2 et § 11.2.2 que, suivant la nature de la décomposition, il faudra supposer que le lagrangien L^Σ est additif par rapport à u (ce qui revient à supposer les fonctions J^Σ et Θ^Σ additives) ou que le lagrangien L^Σ se décompose en N sous-lagrangiens indépendants.

11.2.1 Extension du Principe du Problème Auxiliaire

L'extension du Principe du Problème Auxiliaire au cas des problèmes de point selle passe, comme au § 11.1, par les deux points suivants :

1. on suppose que les fonctions J et Θ (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 Λ défini sur $\mathbb{U} \times \mathbb{V}$ à valeurs dans \mathbb{R} , *fortement convexe* en u et *fortement concave* en p ,⁴⁶ ainsi qu'une constante $\epsilon > 0$.

Le problème auxiliaire au point $(u^{(k)}, p^{(k)})$ associé au problème (103) consiste à trouver, dans $U^{\text{ad}} \times C^*$, le point selle du lagrangien auxiliaire $\mathfrak{L}^{(k)}$ défini par :⁴⁷

$$\mathfrak{L}^{(k)}(u, p) = \Lambda(u, p) + \langle (\epsilon L'_u - \Lambda'_u)(u^{(k)}, p^{(k)}), u \rangle + \langle (\epsilon L'_p - \Lambda'_p)(u^{(k)}, p^{(k)}), p \rangle + \epsilon L^\Sigma(u, p), \quad (106)$$

L'_u et L'_p (resp. Λ'_u et Λ'_p) étant les gradients partiels de L (resp. Λ) par rapport aux variables u et p . L'unique point selle $(u^{(k+1)}, p^{(k+1)})$ de ce problème est caractérisé par les conditions d'optimalité :

$$\begin{aligned} \forall u \in U^{\text{ad}}, \quad & \langle \Lambda'_u(u^{(k+1)}, p^{(k+1)}) + (\epsilon L'_u - \Lambda'_u)(u^{(k)}, p^{(k)}), u - u^{(k+1)} \rangle + \epsilon (L^\Sigma(u, p^{(k+1)}) - L^\Sigma(u^{(k+1)}, p^{(k+1)})) \geq 0 \\ \forall p \in C^*, \quad & \langle \Lambda'_p(u^{(k+1)}, p^{(k+1)}) + (\epsilon L'_p - \Lambda'_p)(u^{(k)}, p^{(k)}), p - p^{(k+1)} \rangle + \epsilon (L^\Sigma(u^{(k+1)}, p) - L^\Sigma(u^{(k+1)}, p^{(k+1)})) \leq 0. \end{aligned}$$

Partant d'un point $(u^{(0)}, p^{(0)})$ appartenant à $U^{\text{ad}} \times C^*$, on construit la suite $\{(u^{(k)}, p^{(k)})\}_{k \in \mathbb{N}}$ engendrée par l'enchaînement des résolutions des problèmes auxiliaires (106). 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 (104), et donc que \bar{u} est une solution du problème (103). Comme au § 11.1, ceci suggère l'algorithme de résolution suivant.

Algorithme 8.

0. Choisir un noyau de décomposition Λ et un coefficient $\epsilon > 0$.
1. Choisir un point $(u^{(0)}, p^{(0)}) \in U^{\text{ad}} \times C^*$ et un seuil de convergence σ ; poser $k = 0$.
2. Calculer le point selle $(u^{(k+1)}, p^{(k+1)})$ du lagrangien auxiliaire (106).
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 8 est alors remplacée par les deux sous-étapes suivantes.

- 2.1 Calculer $u^{(k+1)} = \underset{u \in U^{\text{ad}}}{\operatorname{argmin}} \mathfrak{L}^{(k)}(u, p^{(k)})$.
- 2.2 Calculer $p^{(k+1)} = \underset{p \in C^*}{\operatorname{argmax}} \mathfrak{L}^{(k)}(u^{(k+1)}, p)$.

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.

⁴⁶La 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.

⁴⁷ce point selle est unique grâce aux hypothèses faites sur Λ

11.2.2 Méthode de décomposition à deux niveaux

On se place dans le cadre de la *variante séquentielle* de l'algorithme 8 (phases de minimisation et de maximisation emboîtées). Pour le choix du noyau Λ , 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 raisonnable de choisir une dépendance en u du noyau aussi générale que possible, de la forme $K(u)$.

Ceci conduit au choix de noyau de décomposition suivant :

$$\Lambda(u, p) = K(u) - \frac{1}{2\alpha} \|p\|^2, \quad (107)$$

où K est une fonction fortement convexe différentiable et où α est une constante positive soulignant la forte concavité en p du noyau Λ . Avec ce choix, l'étape 2.2 de la variante séquentielle de l'algorithme 8 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.

Algorithme 9.

0. Choisir un noyau de décomposition K et des coefficients α et ϵ tous deux positifs .
1. Choisir un point $(u^{(0)}, p^{(0)}) \in U^{\text{ad}} \times C^*$ et un seuil de convergence σ ; poser $k = 0$.
- 2.1 Résoudre $\min_{u \in U^{\text{ad}}} K(u) + \langle J'(u^{(k)}) - K'(u^{(k)}), u \rangle + \epsilon J^\Sigma(u) + \epsilon \langle p^{(k)}, \Theta'(u^{(k)}) \cdot u + \Theta^\Sigma(u) \rangle$
et noter $u^{(k+1)}$ une 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 :⁴⁸

- **Uzawa** : $J^\Sigma = 0$, $\Theta = 0$, $K = J$;
- **Arrow-Hurwicz** : $J^\Sigma = 0$, $\Theta^\Sigma = 0$, $K = \frac{1}{2} \|u\|^2$.

Pour ce qui est de la décomposition, seule la phase de minimisation en u est concernée, et on est ramené au cas du § 11.1. On suppose donc qu'il existe une décomposition (u_1, \dots, u_N) de la variable u telle que :

- U^{ad} se met sous la forme d'un produit cartésien par rapport à cette décomposition ;
- les fonctions J^Σ et Θ^Σ sont additives par rapport à cette décomposition.

Il suffit alors de choisir un noyau K 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 :

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

⁴⁸pourvu que l'on puisse prendre $\epsilon = 1$

11.2.3 Méthode de décomposition à un niveau

On se place dans le cadre de l'algorithme 8 original (recherche de point selle). Afin de pouvoir réinterpréter le calcul du point selle dans cet algorithme comme un problème d'optimisation sous contraintes, on choisit un noyau de décomposition Λ de la forme d'un lagrangien, à savoir :

$$\Lambda(u, p) = K(u) + \langle p, \Omega(u) \rangle , \quad (108)$$

où K est une fonction fortement convexe différentiable et où Ω est une fonction différentiable à valeurs dans l'espace \mathbb{V} . On notera que le noyau Λ ainsi choisi n'est pas fortement concave en p . Le choix du noyau de décomposition consiste donc à se donner, outre la fonction K , un opérateur de contraintes Ω de même nature que Θ . L'algorithme qui en résulte est le suivant.

Algorithme 10.

0. Choisir les fonctions K et Ω 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 σ ; poser $k = 0$.
2. Résoudre
$$\min_{u \in U^{\text{ad}}} K(u) + \langle \epsilon J'(u^{(k)}) - K'(u^{(k)}), u \rangle + \epsilon J^\Sigma(u) + \langle p^{(k)}, (\epsilon \Theta'(u^{(k)}) - \Omega'(u^{(k)})) \cdot u \rangle$$
 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 $\|u^{(k+1)} - u^{(k)}\| + \|p^{(k+1)} - p^{(k)}\| > \sigma$.

L'utilisation de cet algorithme en décomposition passe par le fait que l'on soit capable de caractériser les cas où le calcul d'un point selle se décompose en N calculs de points selle indépendants. Pour ce qui est de la partie "facile" L^Σ du problème, on suppose que les espaces \mathbb{U} et \mathbb{V} se décomposent sous la forme $\mathbb{U}_1 \times \dots \times \mathbb{U}_N$ et $\mathbb{V}_1 \times \dots \times \mathbb{V}_N$, de telle sorte que :⁴⁹

- U^{ad} se met sous la forme d'un produit cartésien par rapport à la décomposition de \mathbb{U} ;
- la fonction J^Σ est additive par rapport à la décomposition de \mathbb{U} ;
- la fonction Θ^Σ est bloc-diagonale par rapport aux décompositions de \mathbb{U} et de \mathbb{V} .

Notant (u_1, \dots, u_N) et (p_1, \dots, p_N) les composantes de $u \in \mathbb{U}$ et de $p \in \mathbb{V}$ suivant la décomposition de ces espaces, l'hypothèse faite sur Θ^Σ implique :

$$\langle p, \Theta^\Sigma(u) \rangle = \sum_{i=1}^N \langle p_i, \Theta_i^\Sigma(u_i) \rangle ,$$

où $\Theta_i^\Sigma : \mathbb{U}_i \longrightarrow \mathbb{V}_i$ désigne la composante de Θ^Σ correspondant au sous-espace \mathbb{V}_i . Cette dernière écriture, associée à l'additivité de J^Σ , conduit à la décomposition du calcul du point selle de L^Σ .

Toujours dans l'optique de la décomposition, on choisit le noyau Λ tel que :

- la fonction K soit additive par rapport à la décomposition de \mathbb{U} ;
- la fonction Ω soit bloc-diagonale par rapport aux décompositions de \mathbb{U} et de \mathbb{V} .

⁴⁹On notera que ces deux décompositions comptent le même nombre de "morceaux".

On constate alors que, avec ces hypothèses et ces choix, le problème de minimisation sous contraintes en u de l'algorithme 10 se scinde en N sous-problèmes d'optimisation sous contraintes indépendants ne portant chacun que sur les variables u_i et p_i . Notant Θ_i et Ω_i les composantes de Θ et Ω correspondant au sous-espace \mathbb{V}_i et utilisant le caractère bloc-diagonal de la fonction Ω_i , le i -ème sous-problème issu de l'optimisation sous contraintes dans l'algorithme 10 s'écrit :⁵⁰

$$\min_{u_i \in U_i^{\text{ad}}} K_i(u_i) + \left\langle \epsilon J'_{u_i}(u^{(k)}) - K'_i(u_i^{(k)}), u_i \right\rangle + \epsilon J_i^\Sigma(u_i) + \langle p^{(k)}, \epsilon \Theta'_{u_i}(u^{(k)}) \cdot u_i \rangle - \left\langle p_i^{(k)}, \Omega'_i(u_i^{(k)}) \cdot u_i \right\rangle ,$$

sous la contrainte :

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

11.2.4 Convergence

Pour tout ce qui concerne la convergence des méthodes de décomposition à un et deux niveaux, ainsi que pour les différentes variantes possibles pour ces méthodes, on se reportera à [10].

⁵⁰où $\Theta'_{u_i}(u^{(k)})$ représente la jacobienne partielle en u_i de la fonction Θ au point $u^{(k)}$

