Algorithme polynomial pour les points fixes de fonctions max-affines

1 Problème

On considère un système de type point fixe défini par un opérateur max-affine sur $\mathbb{R}^n$:

$$x_i = \max_{k \in K_i} [T^{(k)} x]_i, \quad i = 1, \dots, n, \tag{1}$$

où pour tout k et $y \in \mathbb{R}^n$, $T^{(k)} y$ est une fonction affine :

$$[T^{(k)} y]_i = \sum_{j=1}^{n} a_{ij}^{(k)} y_j + b_i^{(k)}. \tag{2}$$

L'objectif est de déterminer l'ensemble X^* des solutions de (1), éventuellement vide.

—

2 Reformulation comme programme linéaire

Considérons le problème d'optimisation suivant :

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \max_{i,k \in K_i} \left(x_i - [T^{(k)} x]_i \right) \\ \text{sous les contraintes} \quad & x_i \geq [T^{(k)} x]_i, \quad \forall i, k. \end{aligned} \tag{3}$$

En introduisant une variable z, ce problème peut être écrit comme un programme linéaire classique :

$$\begin{aligned} \min \quad & z \\ \text{sous les contraintes} \quad & x_i - [T^{(k)} x]_i \leq z, \quad \forall i, k, \\ & x_i - [T^{(k)} x]_i \geq 0, \quad \forall i, k. \end{aligned} \tag{4}$$

Ainsi, (3) peut être résolu efficacement avec des algorithmes de programmation linéaire.

—

3 Algorithme pour déterminer X^*

L'idée clé est d'utiliser l'optimisation pour identifier et supprimer les contraintes impossibles, ce qui conduit à un algorithme polynomial.

1. Initialiser les ensembles K_i.
2. Répéter :
   (a) Résoudre le programme linéaire (3) et noter la solution optimale $(x^\dagger, z)$ ainsi que l'indice $(i^\dagger, k^\dagger)$ correspondant à la contrainte la plus violée.
   (b) Si $z = 0$, alors $x^\dagger$ est une solution du système (1) pour les ensembles K_i restants. L'algorithme peut s'arrêter.
   (c) Si $z > 0$, alors aucune solution $x^* \in X^*$ ne peut satisfaire la contrainte $(i^\dagger, k^\dagger)$ à l'égalité. On peut donc supprimer $k^\dagger$ de $K_{i^\dagger}$ sans perdre de solutions.
   (d) Si pour un certain i, K_i devient vide, alors $X^* = \emptyset$ (pas de solution).
3. Répéter jusqu'à convergence.

Remarques.

- L'algorithme s'arrête après au plus $\sum_i |K_i| - n$ étapes, car à chaque itération au moins un élément de K_i est supprimé.
- À la fin de l'élagage, l'ensemble X^* est donné par toutes les solutions du système linéaire combinatoire :

$$x_i = [T^{(k_i)} x]_i, \quad k_i \in K_i \text{ (ensembles réduits)}.$$

- Chaque étape consiste en la résolution d'un programme linéaire de taille $O(n \cdot \sum_i |K_i|)$, donc l'ensemble de l'algorithme est polynomial.

—

4 Conclusion

Cet algorithme fournit une procédure efficace pour déterminer l'ensemble de solutions de points fixes pour les fonctions max-affines. Il repose sur une combinaison de :

- reformulation comme programme linéaire,
- identification des contraintes impossibles via l'optimisation,

- élagage progressif des ensembles K_i.

La méthode garantit l'arrêt en nombre fini d'itérations et permet de conclure soit à l'existence, soit à l'absence de solutions.