

Résolution de programmes linéaires par batch sur GPU

Stage de recherche en optimisation numérique – RTE / Mines Paris – PSL

Contexte

Dans le secteur de l'énergie, une grande partie des problèmes d'optimisation se modélisent par programmation linéaire (PL), du calcul à court terme du dispatch des générateurs (*unit commitment*) à l'étude de l'évolution à long terme des systèmes énergétiques. Les programmes linéaires rencontrés sont souvent de grande taille, et peuvent comporter des variables binaires.

Chaque programme linéaire est généralement écrit dans un modèleur, puis résolu par un solveur dédié, le plus souvent basé sur l'algorithme du simplexe ou sur une méthode de points intérieurs [3]. Ces algorithmes s'appuient sur des routines performantes d'algèbre linéaire creuse pour passer à l'échelle.

Cependant, la résolution d'un problème d'optimisation est rarement une fin en soi. Les processus opérationnels consistent souvent à résoudre un grand nombre de problèmes linéaires en faisant varier certains paramètres. Par exemple, pour les exercices prospectifs de long terme (Bilan prévisionnel, Futurs énergétiques), RTE modélise le fonctionnement du système électrique français sur une année. Le problème annuel est divisé en une succession de problèmes hebdomadaires, et plusieurs réalisations de l'année future (jusqu'à 1 000) sont considérées pour tenir compte des incertitudes (consommation, production renouvelable, maintenances). Il est donc courant de devoir résoudre $52 \times 1\,000$ problèmes linéaires de structure similaire.

Lorsque tous les problèmes ont la même structure (mêmes nombres de variables et de contraintes, même position des coefficients dans la matrice des contraintes), leur résolution par batch se prête bien à l'architecture des GPU modernes, qui exécutent massivement en parallèle une même instruction sur des données différentes (parallélisme *Single Instruction Multiple Data*, ou SIMD). Le nouveau solveur MadIPM [2] permet une telle résolution, en s'appuyant sur le solveur linéaire NVIDIA cuDSS.

Sujet

Partie 1 : adaptation de MadIPM aux problèmes Antares

Dans un premier temps, le stage consistera à prendre en main MadIPM et à l'adapter à la résolution des problèmes d'optimisation issus d'Antares, l'outil de simulation développé par RTE [1]. Un benchmark précis sera mené pour quantifier les gains apportés par la résolution sur GPU. Cette étude permettra ensuite d'améliorer la robustesse du solveur, en cherchant d'abord un *scaling* adapté aux problèmes traités par Antares, puis un algorithme permettant de retrouver la matrice de base associée à la solution (*crossover*).

Partie 2 : exploitation de la structure des solutions

Une fois la robustesse assurée, nous nous intéresserons de plus près à la structure de la solution afin d'améliorer les performances du solveur. L'algorithme de points intérieurs utilisé dans MadIPM peut en effet être interprété comme une méthode de Newton appliquée aux conditions de stationnarité du problème (conditions de Karush-Kuhn-Tucker), lissées par un terme d'homotopie.

Plutôt que de résoudre indépendamment tous les problèmes d'un même batch, nous chercherons à exploiter la structure de la solution. Si l'on ne perturbe que le second membre des contraintes et/ou le vecteur de coûts, la solution primale-duale est une fonction linéaire par morceaux des perturbations [4], chaque cassure correspondant à un changement de l'ensemble des contraintes actives (*active set*). Cette structure peut être utilisée directement au sein de l'algorithme de points intérieurs pour accélérer la résolution d'un scénario à l'autre. Une extension consistera à étudier les perturbations de la matrice des contraintes, un problème encore ouvert : la solution n'est alors plus linéaire par morceaux, mais rationnelle par morceaux.

Cette seconde partie, plus théorique, permettra de monter en compétence sur l'analyse de sensibilité des problèmes d'optimisation [5].

Profil

Nous cherchons une personne motivée par l'optimisation numérique et le calcul haute performance, ayant idéalement suivi un master d'optimisation et/ou une formation d'école d'ingénieurs. De bonnes bases en programmation (Julia, Python...) sont attendues ; des notions de programmation GPU (CUDA) et d'algèbre linéaire numérique seront un plus. Le but du stage est de développer des compétences avancées en optimisation numérique. Il pourra être poursuivi par une thèse de doctorat à Mines Paris – PSL.

Encadrement

- RTE, direction R&D, feuille de route Modélisation Équilibre Offre-Demande : Thomas Bittar (thomas.bittar@rte-france.com) et Antoine Oustry (antoine.oustry@rte-france.com)
- Mines Paris – PSL : François Pacaud (francois.pacaud@minesparis.psl.eu)

Informations pratiques

- **Entité d'accueil** : RTE, direction R&D, service CLER
- **Lieu** : RTE, bâtiment Window, Puteaux (La Défense)
- **Période** : février/mars 2027 à juillet/août 2027 (5 à 6 mois)
- **Gratification** : selon la grille RTE
- **Candidature** : CV et lettre de motivation à envoyer aux trois encadrants

Références

- [1] Antares Simulator : <https://antares-simulator.org/>
- [2] MadIPM : <https://github.com/madsuite-org/MadIPM.jl>
- [3] Wright, S. J. *Primal-dual interior-point methods*. Society for Industrial and Applied Mathematics, 1997.
- [4] Adler, I., Monteiro, R. D. C. A geometric view of parametric linear programming. *Algorithmica* 8, 161–176 (1992). <https://doi.org/10.1007/BF01758841>
- [5] Pacaud, F. Sensitivity analysis for parametric nonlinear programming: A tutorial. arXiv:2504.15851 (2025).