Algorithmique avancée
Denis TRYSTRAM
Branch-and-Bound (et introduction aux algorithmes d'approximation)
ENSIMAG 2A - filière Alternance, octobre 2018

Présentation

Nous nous intéressons dans un premier temps à la résolution de problèmes *difficiles* par méthodes exactes, puis par des méthodes avec approximation garantie. Nous avons vu dans les autres chapitres que l'on ne peut espérer concevoir une solution en temps polynomial pour ce type de problèmes, par contre, nous allons montrer comment construire des solutions efficaces qui font beaucoup mieux que l'énumération exhaustive de toutes les solutions possibles. Bien sûr, ces stratégies restent exponentielles. C'est le cas de la méthode de séparation/évaluation (plus connues par son nom en anglais : *Branch-and-Bound*).

Le principe de la méthode est présenté ci-dessous pour la résolution du problème général suivant :
minimiser f(x) pour $x \in \Omega$ subset of $\mathbb{R}^n$
sous un ensemble de contraintes (C_i)

Il est possible de limiter l'explosion combinatoire en introduisant un système de bornes dans le parcours des configurations possibles du problème.

Un algorithme Branch-and-Bound consiste à énumérer systématiquement toutes les solutions candidates, où de larges ensembles de configurations inutiles sont éliminées en utilisant des bornes inférieures et supérieures (estimations de la fonction objectif que l'on cherche à optimiser).

Concrétement, une stratégie Branch-and-Bound divise le problème en un certain nombre de sous-problèmes. Supposons que l'on dispose d'une méthode qui fournit une borne inférieure du coût de n'importe quelle solution dans un sous-ensemble donné. Si la meilleure solution trouvée jusqu'à présent a un coût plus petit que cette borne, il est inutile d'explorer les solutions de ce sous-ensemble.

Le Branch-and-Bound repose sur deux procédures qui calculent (efficacement) une borne supérieure et inférieure de la valeur optimale de l'objectif dans chaque région.

- Une borne supérieure peut être déterminée par l'évaluation de n'importe quelle configuration dans la région considérée (par une heuristique glouton ou par optimisation locale).

- Une borne inférieure de la solution dans la région considérée est en général fournie par relaxation (ou par dualité).

Les figures suivantes 1, 3 and 5 illustrent le schéma de principe de la méthode.

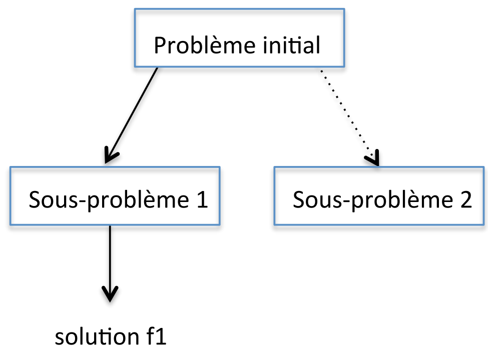

Figure 1: Principe du Branch-and-Bound. Première étape : on explore une cascade de sous-problèmes jusqu'à l'obtention d'une solution réalisable (ici, $f1$).

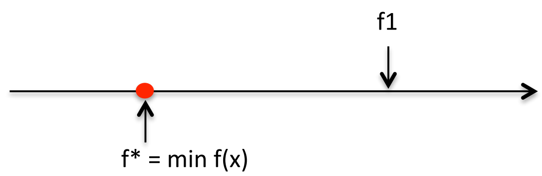

Figure 2: La première solution fournie un borne supérieure de l'optimal f^*.

Si l'exploration du sous-problème 4 conduit à une borne inférieure plus grande que $f3$, il est inutile de continuer sur cette branche.

Algorithmes d'approximation

Lorsque la solution exacte est trop coûteuse à obtenir, on cherche souvent à concevoir un algorithme plus rapide dont la solution n'est plus exacte, mais est *garantie* par rapport à la solution optimale (exacte). La définition suivante est considérée pour un problème de minimisation.

Définition. On dit que l'algorithme $\mathcal{A}$ est une ρ-approximation pour le problème considéré ssi

$f_{\mathcal{A}}(\mathcal{I}) \leq \rho f^*(\mathcal{I}), \forall$ instance $\mathcal{I}$ $(\rho > 1)$.

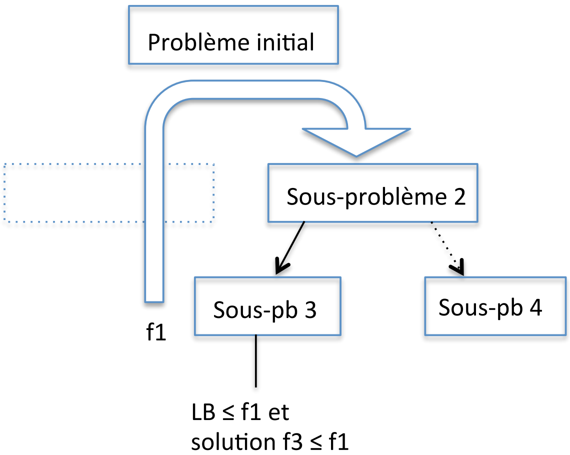

Figure 3: Principe du Branch-and-Bound. Seconde étape (exploration du sous-problème 3) : on continue à explorer l'arbre des configurations en cascade de sous-problèmes jusqu'à l'obtention d'une solution réalisable. Ici, la nouvelle solution $f3$ améliore la précédente $f1$.

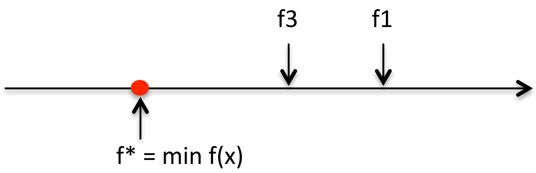

Figure 4: La seconde solution $f3$ améliore la borne supérieure $f1$.

Notons pour conclure que la méthode du Branch-and-Bound peut être utilisée comme un algorithme d'approximation si l'élagage d'une branche se fait avec une tolérance de ϵ. En effet, dans la situation où l'évaluation de la borne inférieure d'un sous-problème est plus grande que la valeur courante f, mais que $LB + \epsilon > f$, il est possible de couper cette branche. On obtient alors une solution approchée de l'optimal (figure 7).

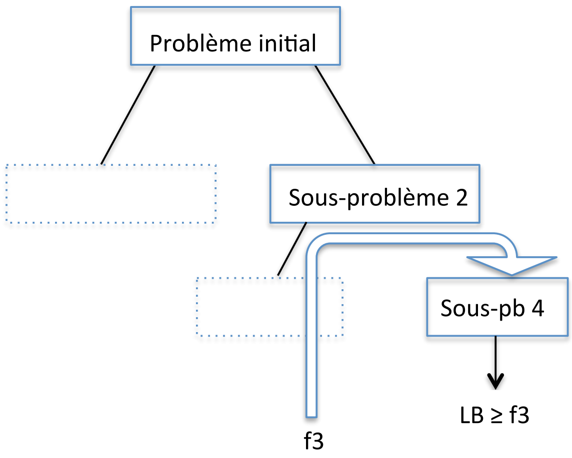

Figure 5: Principe du Branch-and-Bound. Troisième étape : l'évaluation de la borne inférieure du sous-problème 4 montre que toutes les solutions seront moins bonnes que la solution déjà trouvée : on arrête l'exploration dans cette branche.

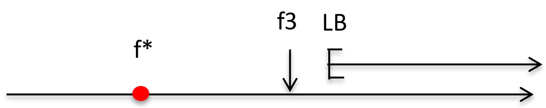

Figure 6: On arrête l'exploration dans cette branche car la borne inférieure du sous-problème est plus grande que la solution courante $f3$.

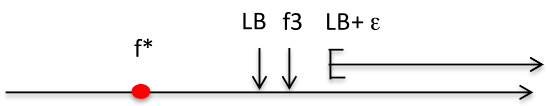

Figure 7: Principe du Branch-and-Bound avec approximation : la borne inférieure LB du sous-problème est plus petite que la solution courante, mais on arrête l'exploration dans cette branche car $LB > f3 - \epsilon$.