# Synthèse du cours 9 : Algorithmes gloutons

16 novembre 2021

François Laroussinie

**NB :** Ces synthèses ont pour but de compléter les notes prises en cours. Elles ne les remplacent pas ! En particulier, la plupart des preuves n'y figurent pas. Rappel : il faut programmer les algorithmes vus en cours.

## 0.1 Le problème de l'allocation d'une ressource

On dispose d'une ressource (par exemple, un camion, un processeur,...), et on a un ensemble $\mathcal{E}$ de *requêtes* : chaque requête vise à utiliser cette ressource entre deux dates. **L'objectif est de satisfaire le plus grand nombre de requêtes possible.**

Une requête sera une paire $(d, f) \in \mathbb{N} \times \mathbb{N}$ avec $d < f$. Un sous-ensemble $\mathcal{F} \subseteq \mathcal{E}$ de requêtes est dit *compatible* lorsqu'une aucune requête n'en intersecte une autre, c'est-à-dire lorsque pour tout $(d_i, f_i), (d_j, f_j) \in \mathcal{F}$, on a soit $f_i \leq d_j$ (la requête $i$ se termine avant le début de la requête $j$, on dit alors que la requête $i$ précède $j$) ou $f_j \leq d_i$ (la requête $j$ précède la requête $i$).

Le problème s'énonce alors comme suit :

**Input :** un ensemble $\mathcal{E}$ de $n$ requêtes $(d_i, f_i)_{1 \leq i \leq n}$ avec $d_i < f_i$ pour tout $i$.
**Output :** un sous-ensemble compatible $\mathcal{F} \subseteq \mathcal{E}$ de taille maximale.

La taille du problème est $n$.

Dans la suite, on parlera des « solutions optimales pour un ensemble de requêtes $\mathcal{E}$ » pour désigner les sous-ensembles compatibles de taille maximale.

**Algorithme.** On va utiliser l'algorithme ci-dessous. La variable aux sert à s'assurer que la prochaine requête qui sera ajouté à la solution n'intersecte pas la dernière requête choisie.

```
Trier l'ensemble E par ordre de date de fin croissante (donc tel que f_i ≤ f_{i+1})
F = ∅
aux = 0
Pour i = 1, ..., n :
..... Si aux ≤ d_i Alors :
.......... F+ = {(d_i, f_i)}
.......... aux = f_i
Retourner F
```

La complexité de l'algorithme est donc $O(n \cdot \log n)$ pour la phase de tri et $O(n)$ pour le reste...

Il reste à prouver sa correction ! Il est clair qu'il renvoie bien un sous-ensemble compatible (c'est un invariant direct de l'algorithme) mais il faut montrer son optimalité. On va la montrer de deux manières. D'abord avec une preuve *adhoc* et ensuite en repartant des deux propriétés des algorithmes gloutons décrites précédemment...

**Preuve de correction 1.** On va montrer que l'algorithme donne une solution optimale, c'est-à-dire un sous-ensemble $\mathcal{F}$ de taille maximale. **Pour cela, on va supposer que ce n'est pas le cas** et donc que la ou les solutions optimales contiennent plus de requêtes que le sous-ensemble renvoyé par l'algorithme. Dans la suite, on note $\mathcal{F} = \{x_1, \dots, x_p\}$ le sous-ensemble compatible renvoyé par l'algorithme trié par date de fin croissante. Etant donnée la manière dont l'algorithme procède, on sait aussi que $x_1$ est la première requête choisie par l'algorithme pour $\mathcal{F}$, $x_2$ est la seconde, $x_3$ la troisième,...

On suppose à présent qu'une solution optimale comprend $q$ requêtes avec $q > p$. Parmi toutes les solutions optimales, on *choisit celle qui partage le plus des premiers choix fait par l'algorithme.* Soit $\mathcal{G}$ cette solution optimale $\{y_1, y_2, \dots, y_q\}$ aussi triée par date de fin croissante. On a donc $d(y_1) < f(y_1) \leq d(y_2 \leq f(y_2) < \dots$ où $d(y_i)$ désigne la date de début de la requête $y_i$ et $f(y_i)$ sa date de fin.

$\mathcal{G}$ peut aussi s'écrire $\{x_1, \dots, x_k, y_{k+1}, \dots, y_q\}$ et $x_{k+1} \neq y_{k+1}$ : $k$ est le nombre de premiers choix de l'algorithme partagés par $\mathcal{G}$. Et étant donnée hypothèse sur le choix de $\mathcal{F}$, il n'existe pas de solutions optimales qui contiennent les requêtes $x_1, \dots, x_{k+1}$.

On sait aussi que $k < p$ car si $k = p$ alors l'algorithme aurait aussi essayer d'ajouter les requêtes $y_{k+1}$, $y_{k+2}$,... et que certaines d'entre elles auraient été ajoutées puisqu'elles sont compatibles avec les premières requêtes choisies... Maintenant on définit $\mathcal{G}'$ par $\mathcal{G}\backslash\{y_{k+1}\} \cup \{x_{k+1}\}$. On voit alors que $\mathcal{G}'$ est un sous-ensemble compatible puisque si l'algorithme choisit la requête $x_{k+1}$, c'est qu'elle est la requête ayant une date de fin minimale parmi celles compatibles avec $\{x_1, \dots, x_k\}$, et donc la requête $y_{k+1}$ a une date de fin supérieure ou égale à celle de $x_{k+1}$, et donc remplacer $y_{k+1}$ par $x_{k+1}$ ne pose pas de problème pour les autres requêtes de $\mathcal{G}$ : on a bien $f(x_{k+1}) \leq d(y_{k+2})$ (car $f(x_{k+1}) \leq f(y_{k+1})$).
$\mathcal{G}'$ est donc un sous-ensemble compatible et de même taille que $\mathcal{G}$, c'est donc une solution optimale et il partage un premier choix de l'algorithme de plus que $\mathcal{G}$, on a donc une contradiction avec les hypothèses de départ.

**Preuve de correction 2.** Pour cette seconde preuve, on procède en montrant d'abord que l'algorithme fait un « bon » premier choix, c'est à dire un choix qui permettra *in fine* d'obtenir un sous-ensemble optimal : autrement dit, un « bon choix » est un choix inclus dans une solution optimale.

**Lemme 1** *Soit $\mathcal{E}$ un ensemble de requête et soit $e$ une requête de $\mathcal{E}$ ayant une date de fin minimale. Il existe une solution optimale avec la requête $e$.*

**Preuve :** Soit $\mathcal{G} = \{y_1, \dots, y_q\}$ une solution optimale, où les $y_i$ sont triés par date de fin croissante, et donc telle que $d(y_i) < f(y_i) \leq d(y_{i+1}) \dots$ Etant donné l'algorithme, on sait que $f(e) \leq f(y_1)$, d'où $f(e) \leq d(y_2)$ et donc le sous-ensemble $\{e, y_2, \dots, y_q\}$ est compatible et constitue aussi une solution optimale qui contient le premier choix de l'algorithme. $\square$

**Lemme 2** *Soit $\mathcal{E}$ un ensemble de requêtes. Soit $\mathcal{G}$ une solution optimale pour $\mathcal{E}$ et soit $e \in \mathcal{G}$. Alors $\mathcal{G}' = \mathcal{G}\backslash\{e\}$ est une solution optimale pour les requêtes de $\mathcal{E}$ compatibles avec $e$, c'est-à-dire pour l'ensemble de requêtes $\mathcal{E}' = \{e' \in \mathcal{E} \mid d(e') \geq f(e) \vee d(e) \geq f(e')\}$.*

**Preuve :** Soit $|\mathcal{G}| = k$ et donc $|\mathcal{G}'| = k - 1$.
Supposons que le lemme soit faux. Alors il existe $\mathcal{G}''$ une solution optimale pour $\mathcal{E}'$ avec $|\mathcal{G}''| > |\mathcal{G}'| = k - 1$ . Il suffit de prendre $\mathcal{G}'' \cup \{e\}$ pour obtenir un sous-ensemble compatible

(car par hypothèse $\mathcal{G}''$ est compatible avec $e$) pour $\mathcal{E}$ et il est de cardinal strictement supérieur à $k$, donc $\mathcal{G}$ n'est pas optimal, et cela contredit les hypothèses de départ. □

Il reste à en tirer le théorème de correction... Pour cela il faut noter qu'une itération de l'algorithme consiste à choisir une requête ayant une date de fin minimale puis à passer à l'étape suivante où il va chercher une solution pour l'ensemble des requêtes compatibles avec son choix précédent, *etc.*

La preuve de correction se fait alors par induction sur la taille $n$ de $\mathcal{E}$ :

— $n = 1$ : l'algorithme renvoie l'unique requête et c'est bien évidemment la solution optimale !

— $n + 1$ : Soit $\mathcal{F}$ la solution renvoyée par l'algorithme, et soit $e$ la première requête choisie par l'algorithme. On sait par le lemme 1, qu'il existe une solution optimale $\mathcal{G}$ contenant $e$. Soit $k = |\mathcal{G}|$. Par le lemme 2, on sait que $\mathcal{G}' = \mathcal{G}\backslash\{e\}$ est une solution optimale pour l'ensemble $\mathcal{E}'$ des requêtes compatibles avec $e$ (donc avec $\mathcal{E}' = \{e' \in \mathcal{E} \mid d(e') \geq f(e) \vee d(e) \geq f(e')\}$). Et on a $|\mathcal{G}'| = k - 1$.
A quoi correspond l'ensemble $\mathcal{F}$ renvoyé par l'algorithme ? A $\{e\} \cup \mathcal{F}'$ où $\mathcal{F}'$ est l'ensemble renvoyé par l'algorithme pour l'ensemble $\mathcal{E}'$. Mais par hypothèse d'induction, on sait que l'algorithme est correct pour $\mathcal{E}'$ (car $|\mathcal{E}'| \leq n$) et donc $|\mathcal{F}'| = |\mathcal{G}'| = k - 1$ et donc $|\mathcal{F}| = k$ et c'est donc une solution optimale.