Annexe B

Algorithmes Dijkstra et $A^\star$

Hugo Salou MPI*

Dernière mise à jour le 8 mars 2023

On s'intéresse, dans cette annexe, à l'algorithme $A^\star$. Cette annexe se situe à l'intersection des chapitres sur les graphes, et sur les jeux. L'algorithme $A^\star$ est une modification de l'algorithme de Dijkstra. Dans cette annexe, on prouvera la correction de l'algorithme $A^\star$.

On se place dans le contexte d'exécution d'un algorithme de calcul de plus cours chemin utilisant un tableau de distances μ, et le manipulant en n'effectuant que des opérations Relâcher. Notons le graphe $G = (V, E)$, le sommet source s. Notons également $d(\cdot, \cdot)$ la distance induite par les arêtes du graphe G. De plus, on notera $c(\cdot, \cdot)$ les coûts (positifs, non nuls) d'une arête de G. Notons $\ell(\cdot)$ les rongeurs des chemins.

Lemme 1 :

$$\forall (u, v) \in E, \quad d(s, v) \leqslant d(s, u) + c(u, v).$$

Preuve :
Soit $(u, v) \in E$. Soit γ_u un plus court chemin de s à u. Alors, $\gamma_u \cdot v$ est un chemin de s à v :

$$\ell(\gamma_u \cdot v) = \ell(\gamma_u) + c(u, v) = d(s, u) + c(u, v) \geqslant d(s, v).$$

□

Lemme 2 : Pour tout sommet u, la valeur de $\mu[u]$ est décroissant à mesure que l'algorithme s'exécute.

Preuve :
Soit $\underline{\mu}$ et $\bar{\mu}$ les valeurs de μ avant et après une opération Relâcher(x, y). Pour tout sommet $v \neq y$, $\bar{\mu}[v] = \underline{\mu}[v]$. De plus, par disjonction de cas,

- ou bien $\bar{\mu}[y] = \underline{\mu}[y]$, ok.
- ou bien $\bar{\mu}[y] = \underline{\mu}[x] + c(x, y)$ lorsque $\underline{\mu}[x] + c(x, y) \leqslant \underline{\mu}[y]$, donc $\bar{\mu}[y] \leqslant \underline{\mu}[y]$, ok.

□

Lemme 3 : Supposons que l'algorithme ait initialisé μ de la manière suivante :

$$\forall u \in V, \qquad \mu[u] = \begin{cases} +\infty & \text{si } u \neq s \\ 0 & \text{sinon.} \end{cases}$$

Alors, tout au long de l'exécution de l'algorithme, pour tout sommet u, $\mu[u] \geqslant d(s, u)$.

*Preuve :***Initialement** La propriété est vraie par hypothèse.

Hérédité Supposons vrai jusqu'à un certain état $\underline{\mu}$, pour une opération Relâcher(x, y). Pour tout sommet $v \neq y$, $\underline{\mu}[v] = \bar{\mu}[v] \geqslant d(s, v)$. De plus, par disjonction de cas,

- si $\bar{\mu}[y] = \underline{\mu}[y] \geqslant d(s, y)$;
- sinon si $\underline{\mu}[y] = \underline{\mu}[x] + c(x, y) \geqslant d(s, x) + c(x, y) \geqslant d(s, y)$ par hypothèse de récurrence, puis par lemme 1.

□

Corollaire : Si « à un moment » $\mu[u] = d(s, u)$, alors « pour toujours après » $\mu[u] = d(s, u)$. □

Lemme 4 : Si $(s, \ldots, u, v)$ est un plus court chemin de s à v tel que $\underline{\mu}[u] = d(s, u)$ « à un certain moment de l'exécution de l'algorithme. » Notons $\bar{\mu}$ obtenu par Relâcher(u, v).

Preuve :
On a

$$\bar{\mu} = \begin{cases} \underline{\mu}[v] & \text{si } \underline{\mu}[v] < \underline{\mu}[u] + c(u,v) \\ \underline{\mu}[u] + c(u,v) & \text{sinon.} \end{cases}$$

Par disjonction de cas,

- si $\underline{\mu}[v] < \underline{\mu}[u] + c(u,v) = d(s,u) + c(u,v) = d(s,v)$, et donc, en utilisant le lemme 3, $\bar{\mu}[v] = \underline{\mu}[v] = d(s,v)$.
- sinon, $\bar{\mu}[v] = \underline{\mu}[u] + c(u,v) = d(u,v) + c(u,v) = d(s,v)$.

□

Lemme 5 : Soit $(s = x_0, x_1, x_2, \ldots, x_n)$ un plus court chemin. Si on effectue des opérations Relâcher(x_i, x_{i+1}) dans l'ordre $0 \to n-1$, possiblement entremêlés avec d'autres opérations Relâcher, alors pour tout $i \in [\![0, n]\!]$, $\mu_{\text{final}}[x_i] = d(s, x_i)$.

Preuve (par récurrence) :

- Initialement, $\mu[x_0] = d(s, x_0) = d(s,s)$.
- Et, pour tout les i inférieurs stricts, $\mu[x_i] = d(s, x_i)$, on conclut par le lemme 4.

□

(De ce lemme découle l'algorithme de Bellman-Ford.)

Corollaire : L'algorithme Dijkstra est correct.

Preuve :
Soit $t \in V$, un sommet du graphe. Soit $(s = x_0, x_1, \ldots, x_{p-1}, x_p = t)$ un plus court chemin de s à t. Montrons que $\mu_{\text{final}}[t] = d(s,t)$. En utilisant le lemme 5, il suffit de montrer que Dijkstra relâche les arêtes dans cet ordre. Supposons les sommets extraits todo dans l'ordre $x_0, \ldots, x_i$, pour $i \in [\![0, p-1]\!]$. Par l'absurde, supposons que Dijkstra sorte x_k de todo pour $k \in [\![i+2, p]\!]$. « À ce moment là, » on a

$$d(s, x_k) \leqslant \mu[x_k] \leqslant \mu[x_{i+1}] \leqslant d(s, x_{i+1}),$$

d'après le lemme 5, ce qui est absurde ($k > i+1$). □

Corollaire : L'algorithme $A^\star$ est correct.

Algorithme 1 Algorithme $A^\star$ (partiel)

```
Procédure Relâcher(u, v)
    si μ[v] > μ[u] + c(u, v) alors
        μ[v] ← μ[u] + c(u, v)
        π[v] ← u
        η[v] ← μ[v] + h(v)
```

Preuve :
Par l'absurde, supposons que non. Soit $t \in V$, un sommet du graphe, tel que $\mu_{\text{final}}[t] \neq d(s,t)$. Donc $d = \mu_{\text{final}}[t] > d(s,t) = d^\star$. Soit $(s = x_0, x_1, \ldots, x_{p-1}, x_p = t)$ un plus court chemin de s à t de longueur $d^\star$. L'algorithme commence par visiter $x_0 = s$ et on relâche les arêtes sortantes. Alors, $\mu[x_i] = d(s, x_1)$ et $\eta[x_1] = \mu[x_1] + \mu[x_1] + h(x_1) \leqslant d(s, x_1) + d(x_1, t) = d(s,t) = d^\star < d$ par hypothèse. « À ce state, » $\eta[t] = \mu[t] + h(t) \geqslant d + 0$. Ainsi, x_1 devrait être choisi avant t. À un tel moment, $\mu[x_1] = d(s, x_1)$, on relâche alors ses arêtes sortantes ; en particulier x_1 et x_2. Ceci assure alors que $\mu[x_2] = d(s, x_2)$, et $\eta[x_2] = \mu[x_2] + h(x_2) \leqslant d(snx_2) + d(x_2, t) \leqslant d(s,t) = d^\star < d$. « De proche en proche, » alors que l'on choisit x_{p-1} dans todo, on a $\mu[x_{p-1}] = d(s, x_{p-1})$. On relâche alors $\mu[x_p] = d(s, x_p) = d^\star$. Or, $d = \mu_{\text{final}}[t] \leqslant \mu_{\text{à ce moment}}[t]$. Absurde. □

Exemple (ré-entrée dans todo) :
Exécution de l'algorithme $A^\star$ sur l'entrée ci-dessus. La pile todo est vaut donc $\cancel{s}, \cancel{a}, \cancel{b}, \cancel{t}, \cancel{a}$.