notes 1 – 8 décembre 2017

1 Recherche locale pour 3-SAT d'Uwe Schöning

Dans cette section, on utilise le vocabulaire suivant.

Littéral = soit une variable (x) (littéral positif) soit la négation d'une variable ($\neg x$) (littéral négatif).

Clause = ensemble de littéraux. La clause est satisfaite si au moins un des littéraux s'évalue à vrai.

Instance = ensemble de clauses. L'instance est satisfaite si toutes les clauses sont satisfaites. Est aussi appelé formule. Si chaque clause d'une formule contient exactement trois littéraux, on parle d'une formule 3-SAT.

Fait important : 3-SAT est NP-dur alors que 2-SAT est polynomial.

Data: Une formule 3-SAT F sur n variables, un entier positif d
Result: produit une assignation a qui valide la formule, s'il en existe une
for *ever* **do**
  Soit a une assignation arbitraire;
  for $3n$ *fois au plus* **do**
    if $F(a) =$*Vrai* **then**
      retourner *Vrai*;
    end
    Soit C une clause *arbitraire* de F, tel que $C(a) =$Faux;
    Soit x un littéral *aléatoire* de C;
    Changer la valeur $a[x]$ affectée à x
  end
end

Algorithm 1: Recherche locale de Uwe Schöning, 1999 [2]

Quand on change la valeur du littéral x, on valide cette clause, mais au risque d'invalider d'autres. Comment s'assurer qu'il y aura du progrès, au moins en espérance ? Cet algorithme ne s'arête jamais si la formule n'est pas satisfiable, mais sinon il trouve une assignation valide a^* au bout d'un certain temps, et on va analyser l'espérance de temps.

Pour cela on fixe une assignation valide arbitraire a^* et on mesure la *distance de Hamming* avec a, c'est-à-dire le nombre de variables où a et a^* diffèrent.

Lemme 1 *À chaque étape, la distance de Hamming* $d(a, a^*)$ *diminue de* 1 *avec probabilité au moins* $1/3$ *et augmente de* 1 *avec une probabilité au plus* $2/3$.

Preuve : Soit k le nombre de différences entre a et a^* sur les trois littéraux de la clause C choisie dans l'étape.

cas $k = 0$ impossible, car C est valide dans a^* et l'algorithme a choisit une clause qui n'est pas valide dans a.

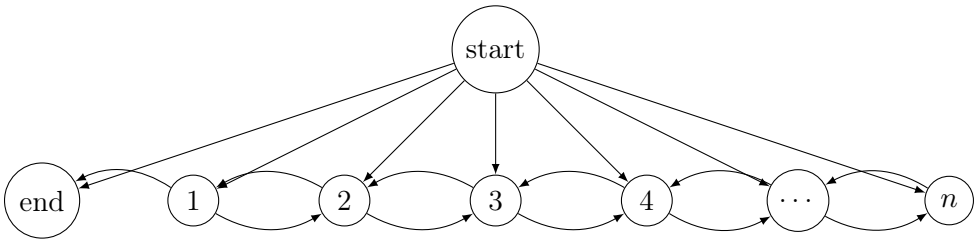

FIGURE 1 – Le comportement de l'algorithme correspond à une marche aléatoire dans ce graphe.

cas $k = 1$ alors avec probabilité 1/3 l'algorithme inverse l'unique variable où a et a^* différent dans C, et la distance diminue. Dans l'autre cas, donc avec probabilité 2/3 la distance augmente.

cas $k = 2$ **ou** $k = 3$ alors la probabilité que la distance diminue est encore plus grande.

□

Le comportement de l'algorithme décrit une marche dans le graphe de la figure 1. L'étiquette dans les sommets représente la distance de Hamming avec l'assignation valide a^*.

Pour une analyse pessimiste on peut associer une probabilité 1/3 aux arcs vers la gauche et 2/3 aux arcs vers la droite.

On va analyser une marche aléatoire sur le graphe ci-haut, mais de manière pessimiste. Pour rendre les choses plus difficiles à l'algorithme, on considère le graphe dont les sommets sont les entiers $\mathbb{Z}$, et on cherche à déterminer la probabilité que l'algorithme atteint le sommet 0 du sommet j au bout de $3j$ étapes. Pire que ça, on veut déterminer la probabilité d'un tel chemin qui utilise exactement j pas du type $i \to i + 1$ et $2j$ pas de type $j \to j - 1$. Cette probabilité est au moins

$$q_j \geq \binom{3j}{j} \left(\frac{2}{3}\right)^j \left(\frac{1}{3}\right)^{2j}.$$

Utilisant la formule d'approximation de Stirling $n! = \Theta(\sqrt{n} \cdot (n/e)^n)$, on obtient

$$\binom{3j}{j} = \Theta\left(\frac{1}{\sqrt{j}} \frac{3^{3j}}{2^{2j}}\right).$$

Et ainsi la probabilité de succès de la boucle intérieure de l'algorithme de Schöning est pour une constante c au moins (la première constante $1/2^n$ est la probabilité de succès pour le choix $j = 1$)

$$\begin{aligned}
&\frac{1}{2^n} + c \cdot \sum_{j=1}^{n} \frac{1}{\sqrt{j}} \cdot \frac{1}{2^j} \binom{n}{j} \cdot \frac{1}{2^n} \\
\geq &\frac{c}{\sqrt{n}} \sum_{j=0}^{n} \binom{n}{j} \frac{1}{2^{n+j}} \\
= &\frac{c}{\sqrt{n}} \left(\frac{3}{4}\right)^n,
\end{aligned}$$

où la dernière égalité utilise l'expansion binomiale de $(1/2 + 1/4)^n$.

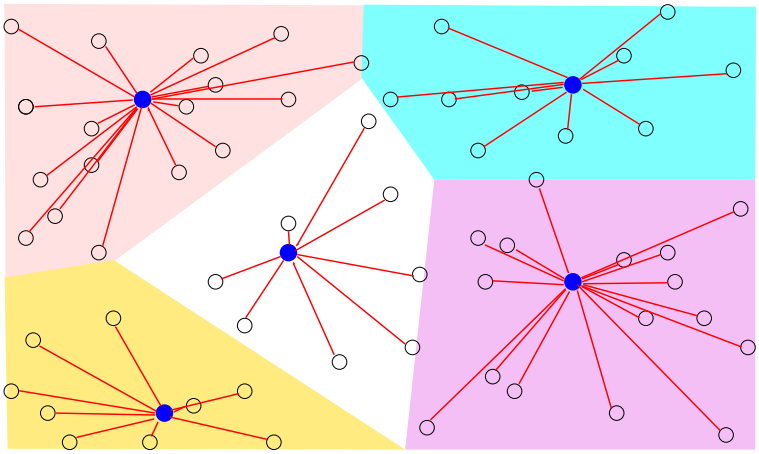

FIGURE 2 – Une instance du problème k-median et une solution. Illustration de Vinayaka Pandit.

Pour amplifier la probabilité de succès p, on peut répéter l'algorithme par exemple $20/p$ fois. En effet la probabilité que l'algorithme échoue à chaque fois est au plus

$$(1-p)^{20/p} \leq e^{-20} = 0,000000002.$$

Ceci montre que la complexité de l'algorithme de Schöning est $O^*((4/3)^n)$ et qu'il succède avec une grande probabilité.

2 Une 5-approximation pour k-median par recherche locale

On considère un espace muni d'une distance d. Soit C un ensemble de n points dans cet espace qui représentent des clients. Pour un ensemble $S \subseteq C$ de k stations, on associe chaque client à une station plus proche. On dénote par S_j la distance vers cette station, donc S_j est défini comme le minimum de $d(j,s)$ sur tout $s \in S$. Le coût de S est défini comme la somme sur ces distances, donc $\text{cost}(S) := \sum_{j \in C} S_j$.

Dans le problème k-médian on cherche un ensemble $S \subseteq C$ de cardinalité k qui minimise $\text{cost}(S)$.

Soit S un ensemble arbitraire de k points de C ;
while $\exists s \in S, s' \notin S : cost(S + s' - s) < (1-\epsilon)cost(S)$ **do**
  ajouter s' à S et enlever s de S ;
end

Algorithm 2: Recherche locale de Aryo, Garg, Khandekar, Meyerson, Mungala et Pandit, 2004 [1]

Soit O une solution optimale.

Cette recherche locale termine quand aucun changement n'arrive à améliorer l'objectif mieux que par un facteur $1-\epsilon$. Le nombre d'itérations est majorée par $\log_{1/(1-\epsilon)}(\text{cost}(S)/\text{cost}(O))$. Comme le logarithme du coût de tout ensemble S est polynomial en la taille de l'entrée, le nombre d'itération est polynomial également.

Considérons S la solution produite par l'algorithme. Pour faciliter l'analyse, supposons que S soit un minimum local, c'est-à-dire $\forall s \in S, s' \notin S : \text{cost}(S + s' - s) \geq \text{cost}(S)$. Nous allons

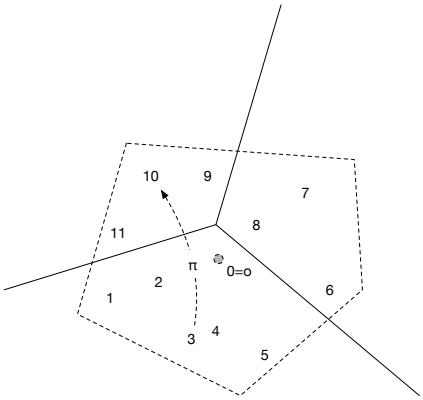

FIGURE 3 – Construction of the permutation π.

montrer $\text{cost}(S) \leq 5 \cdot \text{cost}(O)$, mais pour être précis il faudrait ajouter un facteur $1/(1-\epsilon)$.

Pour le montrer on va se servir des inégalités de la forme

$$\text{cost}(S + o - s) \geq \text{cost}(S)$$

pour certains stations $s \in S, o \in O$. Mais lesquels ?

Notations Soit $N_S(s)$ l'ensemble des clients qui sont affectés à la station s. Il faut penser à des cellules d'un diagramme de Voronoi défini par S. La solution optimale O défini un autre diagramme et on s'intéresse à l'intersection des cellules. Ainsi on dénote par N_s^o l'intersection entre $N_S(s)$ et $N_O(o)$.

On dit qu'une station $o \in O$ est capturée par une station $s \in S$ si $|N_s^o| > \frac{1}{2}|N_O(o)|$, autrement dit si $N_S(s)$ contient au moins la moitié de $N_O(o)$. On dit qu'une station $s \in S$ est *mauvaise* si elle capture une station de O et *bonne* sinon.

Permutation On définit une permutation π sur $N_O(o)$ avec la propriété suivante.

Si s ne capture pas o, alors pour tout $j \in N_s^o : \pi(j) \notin N_s^o$.

Une telle permutation est facile d'obtenir. On renumérote les disons D points de $N_O(o)$ de 0 à $D-1$, de telle sorte à ce que les points de $N_{s'}^o$ soient consécutifs pour tout s'. Puis π envoie un point juste $\lfloor D/2 \rfloor + 1$ unités plus loin, de manière circulaire modulo $|D|$.

Graphe des capturations Maintenant on construit un graphe bi-parti $H(S, O, E)$ où $(s, o) \in E$ ssi s capture o. Le degré entrant de tout nœud dans O est au plus 1.

Échanges considérés On va maintenant considérer k échanges $\langle s, o \rangle$ de la manière suivante. Si s n'est relié qu'à un sommet o alors on considère l'échange $\langle s, o \rangle$. Maintenant il reste disons ℓ stations dans O pas encore considérées dans un échange. Et les stations restants dans S ont soit degré 0 soit degré au moins 2. On va considérer les échanges entre les ℓ stations restantes de O avec des stations de S au degré 0 de telle sorte qu'aucune station ne soit considérée dans plus de 2 échanges. Ces échanges satisfont

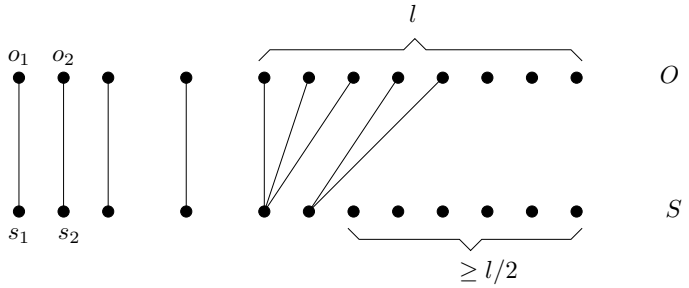

FIG. 5. *Capture graph $H = (S, O, E)$.*

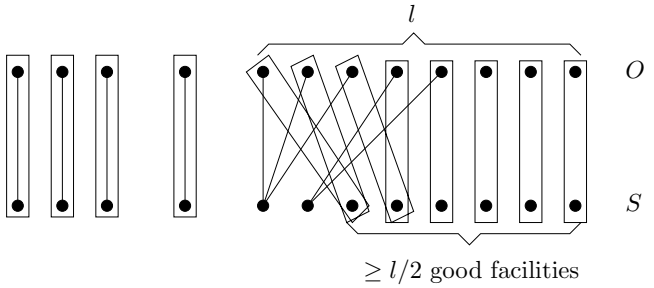

FIG. 6. *k swaps considered in the analysis.*

FIGURE 4 – Sélection des échanges considérés. Illustration par les auteurs de [1]

1. Toute station de O est considérée dans exactement un échange.
2. Une station de S qui capture plus d'une station de O n'est pas considérée dans un échange.
3. Chaque bonne station est considérée dans au plus 2 échanges.
4. Si un échange $\langle s, o\rangle$ est considéré alors s ne capture aucune station $o' \neq o$.

Combiner les ingrédients à l'analyse Considérons ces échanges un par un. Considérons $\langle s, o\rangle$. Le coût $\text{cost}(S - s + o)$ est difficile à exprimer, mais on peut le majorer en affectant les clients à des stations de $S - s + o$. Les clients en dehors de $N_S(s) \cup N_O(o)$ ne changent pas d'affectation. Les clients de $N_O(o)$ sont affectés à o tout simplement.

On s'intéresse aux clients j de $N_S(o')$ pour $o' \neq o$. Comme s ne capture pas o', nous avons $\pi(j) \notin N_S(s)$. Donc $\pi(j) \in N_S(s')$ pour une station $s' \neq s$. La distance de j vers la station la plus proche dans $S - s + o$ est majorée par $d(j, s')$. Qui elle par l'inégalité triangulaire est majorée par

$$d(j, s') \leq d(j, o') + d(\pi(j), o') + d(\pi(j), s') = O_j + O_{\pi(j)} + S_{\pi(j)}.$$

Comme nous avons

$$\text{cost}(S - s + o) - \text{cost}(S) \geq 0$$

nous avons aussi

$$\sum_{j \in N_O(o)} (O_j - S_j) + \sum_{j \in N_S(s) \setminus N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0.$$

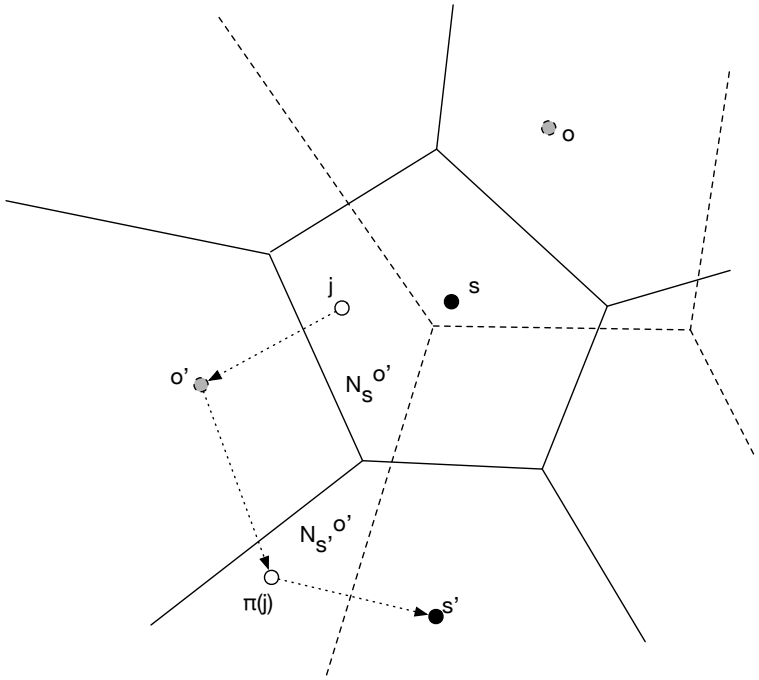

FIGURE 5 – Affecter clients $j \in N_S(s) \setminus N_O(o)$ à des stations $s' \in S - s$.

Maintenant si nous sommons sur les k échanges considérés par (1) le premier terme devient $\text{cost}(O) - \text{cost}(S)$. Par (3) le deuxième terme devient au plus

$$2 \sum_{j \in C} ((O_j + O_{\pi(j)} + S_{\pi(j)} - S_j).$$

Utilisant le fait que π est une bijection, ce terme est exactement $4 \cdot \text{cost}(O)$. Ceci montre que

$$\text{cost}(O) - \text{cost}(S) + 4 \cdot \text{cost}(O) \geq 0,$$

ce qui termine l'analyse.

Références

[1] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on Computing*, 33(3) :544–562, 2004.

[2] Uwe Schöning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 410–414. IEEE, 1999.