Révisions

Exercice 1 – st-connexité en SAT

Soit $G = (V, E)$ un graphe, et s, t deux sommets. On cherche à tester si s et t sont dans deux composantes connexes différentes.

1. Si uv est une arête de G, que dire des composantes connexes de u et v ?
2. Proposer une modélisation du problème en SAT.
3. Que pouvez-vous dire des clauses obtenues ? En déduire la complexité de l'algorithme issu de cette approche.

Exercice 2 – k-centres

Si $\{P_1, \ldots, P_i\}$ est un ensemble de points du plan et P est un point, la distance $\text{dist}(P, \{P_1, \ldots, P_i\})$ est la distance minimum entre P et un point de $\{P_1, \ldots, P_i\}$, c'est-à-dire $\min_{1 \leq j \leq i} \text{dist}(P, P_j)$.

On se donne un entier k et un ensemble de n points $\mathcal{P}$ dans le plan. On cherche à sélectionner k points $P_1, \ldots, P_k$ dans $\mathcal{P}$, qui minimisent la quantité $\max_{P \in \mathcal{P}} \text{dist}(P, \{P_1, \ldots, P_k\})$. Cette quantité est appelée le *coût* de $\{P_1, \ldots, P_k\}$.

1. Lorsque $k = 1$, donner un algorithme permettant de résoudre le problème. Quelle est sa complexité ?

On considère l'algorithme suivant : on choisit un point P_1 quelconque. Puis, pour $i = 2, \ldots, k$, on choisit P_i comme un point de $\mathcal{P}$ qui est le plus loin de l'ensemble $\{P_1, \ldots, P_i\}$.

2. Quel coût est calculé par l'algorithme pour $k = 3$ si on considère l'ensemble de points $\{A, B, C, D, E, F\}$ dont les distances sont les suivantes ? On suppose qu'on choisit $P_1 = A$.
   Préciser les trois points choisis (et expliquez comment vous les avez trouvés).

	A	B	C	D	E	F
A	0	1	4	3	2	2
B	1	0	1	2	3	4
C	4	1	0	3	2	1
D	3	2	3	0	2	3
E	2	3	2	2	0	1
F	2	4	1	3	1	0

3. Comment s'appelle ce type d'algorithme ?
4. Quelle est la complexité de cet algorithme ?
5. On note $\{P_1, \ldots, P_k\}$ les points sélectionnés par notre algorithme et $\{P_1^*, \ldots, P_k^*\}$ une solution optimale, de coût d^*. On suppose maintenant par l'absurde que la solution renvoyée par l'algorithme a un coût strictement supérieur à $2d^*$, c'est-à-dire qu'il existe un point P à distance plus grande que $2d^*$ de tous les P_i.
   (a) Montrer que si $i \neq j$ alors $\text{dist}(P_i, P_j) > 2d^*$.
   (b) En déduire que pour tout i, il existe un unique j tel que $\text{dist}(P_i, P_j^*) \leqslant d^*$.
   (c) Montrer que P est à distance au plus d^* d'un P_j^*.
   (d) Déduire une contradiction. Quel facteur d'approximation pouvez vous en déduire pour l'algorithme décrit ci-dessus ?

Exercice 3 – Directed Feedback Vertex Set dans les Tournois

Un graphe orienté est un graphe $G = (V, E)$ où les arêtes uv sont orientées de u vers v. On appelle u l'origine de l'arête et v la destination de uv.

Un tournoi est un graphe orienté tel que pour toute paire de sommets u, v, uv ou vu est une arête. Un triangle est un cycle orienté de longueur 3. Dans le problème Directed Feedback Vertex Set dans les Tournois (DFVST), on se donne un tournoi T et un entier k et on désire déterminer si la suppression de k sommets laisse un graphe sans cycle orienté.

1. Montrer que dans un tournoi, tout cycle orienté de longueur au moins 4 contient un triangle orienté comme sous graphe.
2. En déduire un algorithme FPT pour DFVST paramétré par k. Quelle est sa complexité ?

On va essayer d'améliorer la complexité de cet algorithme en utilisant une méthode appelée *compression itérative*. On définit le problème Feedback Vertex Set dans les Tournois Quasi-Transitifs (FVSTQT) comme le problème où on se donne un tournoi T, un ensemble de sommets X tel que $T[V \setminus X]$ est acyclique et un entier k et on veut savoir s'il existe k sommets de $V \setminus X$ dont la suppression laisse un tournoi acyclique. (Autrement dit, on veut trouver un feedback vertex set de T qui est disjoint de X).

Soit T un tournoi et $u_1, \ldots, u_n$ un ordre arbitraire des sommets de T. Pour tout i, on note $V_i = \{u_1, \ldots, u_i\}$. On considère les graphes $T_1, \ldots, T_n$ où T_i est la restriction de T aux sommets $u_1, \ldots, u_i$. Soit i un entier. Dans la suite **on suppose qu'on nous donne un ensemble** X_i de sommets de taille au plus k tel que $T_i[V \setminus X_i]$ est acyclique.

3. Combien y a-t-il de sous-ensembles de X_i ?
4. On considère le tournoi T_{i+1}, c'est à dire le tournoi T_i où u_{i+1} a été ajouté. Montrer que $T_{i+1}[V_{i+1} \setminus (X_i \cup \{u_{i+1}\})]$ est acyclique.
5. Montrer que si (T_{i+1}, k) est une instance positive de DFVST alors il existe $Y \subseteq X_i \cup \{u_{i+1}\}$ de taille au plus k tel que $(T_i[V_{i+1} \setminus Y], k - |Y|)$ est une instance positive de FVSTQT.
6. En déduire que si FVSTQT peut être résolu en temps polynomial n^c, alors on peut résoudre DFVST en temps $2^{k+1}n^{c+1}$.

La fin de l'exercice consiste à montrer que FVSTQT peut être décidé en temps polynomial. Soit T un tournoi et X un ensemble de sommets tels que $T[V \setminus X]$ est acyclique.

7. Montrer que l'on peut numéroter les sommets $w_1, \ldots, w_r$ de $T[V \setminus X]$ tel que, pour tout $i < j$, l'arc $w_i w_j$ est orienté de w_i vers w_j.
8. Soit $t = abc$ un triangle de T. Montrer que si tous les sommets de t sont dans X alors l'instance est négative.
9. Montrer que si X contient a, b mais pas c, alors (T, X, k) est positive si et seulement si $(T - c, X, k - 1)$ est positive.
10. On suppose maintenant que X contient seulement le sommet a de t. Comme abc est un triangle orienté, on a $b = w_i$ et $c = w_j$ avec $i < j$. Montrer que toute solution de (T, X, k) contient tous les sommets de l'ensemble $\{w_i, w_{i+1}, \ldots, w_j\}$ sauf au plus un.
11. En déduire un algorithme glouton qui résout FVSTQT en temps polynomial.