École polytechnique — Majeure informatique
Promotion 2004

Cours de conception et analyse d'algorithmes

PC 8 – algorithmes et bio-informatique

14 mars 2007

1. Alignement de familles de séquences

i. Proposer un algorithme en $O(n^3)$ pour trouver le meilleur alignement de 3 séquences de longueur n pour une fonction de distance entre caractères $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{N}$ donnée. δ mesure la distance entre les caractères, ainsi $\delta(\sigma, \sigma') = \delta(\sigma', \sigma) \geq 0$ et $\delta(\sigma, \sigma) = 0$.

ii. Proposer un algorithme qui étant donné deux mots u et v et une fonction de similarité δ, détermine en $O(n^2)$ les facteurs u' de u et v' de v de valeur d'alignement maximum. Ici la fonction δ représente la valeur de l'alignement, ainsi $\delta(-,-) = 0$, $\delta(\sigma, \sigma) > 0$, et $\delta(\sigma, \tau) < 0$ pour $\sigma \neq \tau$.

2. Sous-séquences maximales

i. Étant donnée une suite de valeur $X_1, \ldots, X_n$, avec $X_i \in \mathbb{R}$, donner un algorithme qui trouve en temps linéaire i et j tels que $\sum_{k=i}^{j} X_k$ soit maximale.

ii. Généraliser le problème précédent au cas de séquences circulaires.

iii. Généraliser le problème des sous-séquences maximales au cas où les variables X_i sont sur les sommets d'un arbre et où on souhaite trouver le sous-arbre dont la somme des X_i soit maximale.

3. Arbre d'évolution optimal pour la distance d'édition.

On dispose d'un certain nombre de séquences génétiques et on souhaite expliquer l'évolution de ces séquences par un arbre tel que deux séquences voisines dans l'arbre sont à faible distance. Pour cela on considère une distance d'édition pour laquelle l'ajout, la suppression ou le remplacement d'une lettre par une autre ont valeur 1. Par exemple $d(abab, bab) = 1, d(abab, baba) = 2$

Un arbre d'évolution pour les séquences $f_1, f_2, \ldots, f_m$ est un arbre dont les sommets sont $f_1, f_2, \ldots, f_m$, son coût est la somme des $m-1$ distances entre les séquences reliées par une arête

i. Donner un algorithme en temps polynomial qui résout le calcul de l'arbre d'évolution de coût minimal.

On complique un peu le problème en supposant donnés deux ensembles de séquences $F = \{f_1, f_2, \ldots, f_m\}$ et $G = \{g_1, g_2, \ldots, g_n\}$ on veut expliquer l'évolution de tous les éléments de F en s'aidant d'une partie de G. Pour cela on cherche un sous-ensemble G' de G et un arbre d'évolution dont l'ensemble des sommets est $F \cup G'$ dont le coût est minimal.

On se propose de montrer que le problème de décision associé est alors NP-complet. Pour cela on utilise SOMMET-COUVRANT. On associe à tout graphe $\Gamma = (X, E)$ ayant n sommets $x_1, x_2, \ldots x_n$ et m arêtes un ensemble F compose de m mots et un ensemble G de $n+1$ mots tous de longueur n et sur l'alphabet $\{0, 1\}$. A chaque arête $e_p = \{x_i, x_j\}$ de Γ on associe un mot f_p de F qui comporte $n-2$ 0 et deux 1 en positions i et j, à chaque sommet x_k de Γ on associe un mot g_k ayant $n-1$ 0 et un 1 en position k, on ajoute aussi le mot $g_0 = 0^n$ dans G.

ii. Montrer qu'à tout arbre dont l'ensemble des sommets est $F \cup G'$ $G' \subset G$ on peut associer un arbre d'évolution (éventuellement en ajoutant des éléments dans G') *de même coût* de façon telle que toutes les arêtes de l'arbre correspondent à deux séquences u et v telles que $d(u, v) = 1$.

iii. Montrer qu'un arbre dont toutes les arêtes correspondent à deux séquences u et v telles que $d(u, v) = 1$ détermine un ensemble de sommets couvrant toutes les arêtes dans Γ.

iv. En déduire que la recherche d'un arbre d'évolution conditionné de coût inférieur à k est un problème NP-complet.