Recherche de motifs dans les permutations
Autour des permutations séparables

Mathilde Bouvel

20 décembre 2006

Groupe de travail combinatoire du LIX.

Plan

Recherche de "sous-structure"

Sous-structure et sous-structure commune :

- Mots : Pattern matching, distance d'édition
- Graphes : Isomorphisme de graphes et plus grand sous-graphe commun
- Permutations : recherche d'occurrence d'un motif, de plus grand motif commun

Recherche de "sur-structure" commune

Isomorphisme de graphes et sous-graphe commun

"Subgraph Isomorphism and Related Problems" G. Valiente

- G_1 et G_2 sont-ils isomorphes ?

$\hookrightarrow$ P ? NP-complet ? $\longrightarrow$ Problèmes iso-complets

- G_1 contient-il un sous-graphe isomorphe à G_2 ?

$\hookrightarrow$ NP-complet

- Trouver un plus grand sous-graphe commun à G_1 et G_2.

$\hookrightarrow$ NP-complet (APX-dur)

- Calcul de la distance d'édition entre G_1 et G_2.

$\hookrightarrow$ NP-complet (APX-dur)

Restriction à des classes de graphes

- G_1 et G_2 sont des arbres : trouver le plus grand sous-arbre commun est polynomial : $\mathcal{O}(n^4)$ [Zhang-Shasha]. Amélioration par Klein : $\mathcal{O}(n^3 \log n)$
- G_1 et G_2 sont des graphes planaires : problème du sous-graphe isomorphe toujours NP-complet, mais polynomial lorsque G_2 est fixé
- Résultats lorsque G_1 et G_2 sont des graphes d'intervalle, des graphes à degré borné, ...

Remarque : Plus grand sous-arbre commun $\equiv$ plus grand motif commun à deux permutations triables par pile (i.e. évitant 231) [Micheli Rossin 06]

Motifs dans les permutations : définition

Représentation linéaire des permutations : $\pi = \pi_1 \pi_2 \ldots \pi_n$

$\pi \in S_n$, $\tau \in S_k$ avec $k \leq n$

- La permutation π *contient* une occurrence du motif τ ssi $\exists$ $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ tels que $\pi_{i_1} \pi_{i_2} \ldots \pi_{i_k}$ est isomorphe en ordre à τ : $\pi_{i_p} < \pi_{i_q}$ ssi $\tau_p < \tau_q$
- dans le cas contraire, π *évite* τ
- Par exemple, **1**3**5**6**24** contient 132 et évite 321

Notation :
$S(\tau) =$ l'ensemble de toutes les permutations qui évitent τ

Recherche de motifs dans les permutations

- Recherche d'une occurrence d'un motif τ dans une permutation π

$\hookrightarrow$ *NP*-complet en général, polynomial si τ est séparable [BBL98, Ibarra97]

$\hookrightarrow$ polynomial lorsque τ est fixé

- Recherche d'un plus grand motif commun à deux permutations π et π'

$\hookrightarrow$ *NP*-dur en général, polynomial si π est séparable [BR06]

- Recherche d'un plus grand motif commun de la classe $\mathcal{C}$ à K permutations $\pi^1 \dots \pi^K$

$\hookrightarrow$ polynomial si $\mathcal{C} = \{$ séparables $\}$ et K est fixé [BRV06]

$\hookrightarrow$ *NP*-dur si $\mathcal{C} = \{$ séparables $\}$ et K est arbitraire [BRV06]

Classes de permutations à motifs exclus

- $S(\tau_1, \dots, \tau_k)$ = la classe des permutations ne contenant aucune occurrence de chacun des motifs $\tau_1, \dots, \tau_k$
- classe $\mathcal{C} \equiv$ stable par motif : si $\pi \in \mathcal{C}$ et $\tau \prec \pi$ alors $\tau \in \mathcal{C}$.
- Conjecture de Stanley-Wilf [Marcus Tardos 04] : $|S_n(\tau_1, \dots, \tau_k)| \leq c^n$

Permutations séparables

- Permutations séparables $= S(2413, 3142)$
- Énumérées par les nombres de Schröder : $|S_n(2413, 3142)| = s_{n-1}$ [West 95]

$(s_n)_{n\geq 0} = 1, 2, 6, 22, 90, 394, 1806, 8558, 41586, 206098$

$S(x) = \sum_{n\geq 0} s_n x^n = \frac{1-x-\sqrt{x^2-6x+1}}{2x}$

$s_n = \sum_{i=0}^{n} \binom{2n-i}{i} c_{n-i}$

Caractérisation des permutations séparables

- Permutations séparables = celles possédant un arbre binaire de séparation
  - arbre binaire plan
  - feuilles $\pi_1, \pi_2, \ldots, \pi_n$ de gauche à droite
  - à chaque noeud interne correspond un intervalle
- Remarque : 2413 et 3142 n'ont pas d'arbre de séparation

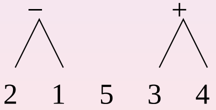
Caractérisation des permutations séparables

- Permutations séparables = celles possédant un arbre binaire de séparation
  - arbre binaire plan
  - feuilles π_1, π_2, . . ., π_n de gauche à droite
  - à chaque noeud interne correspond un intervalle
- Remarque : 2413 et 3142 n'ont pas d'arbre de séparation

2 1 5 3 4

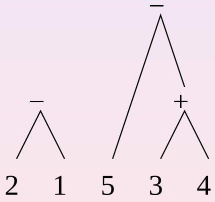
Caractérisation des permutations séparables

- Permutations séparables = celles possédant un arbre binaire de séparation
  - arbre binaire plan
  - feuilles π_1, π_2, . . ., π_n de gauche à droite
  - à chaque noeud interne correspond un intervalle
- Remarque : 2413 et 3142 n'ont pas d'arbre de séparation



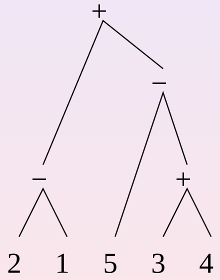
Caractérisation des permutations séparables

- Permutations séparables $=$ celles possédant un arbre binaire de séparation
    - arbre binaire plan
    - feuilles $\pi_1, \pi_2, \ldots, \pi_n$ de gauche à droite
    - à chaque noeud interne correspond un intervalle
- Remarque : 2413 et 3142 n'ont pas d'arbre de séparation



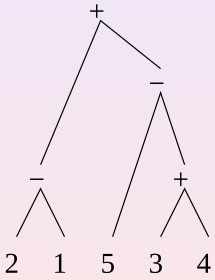
Caractérisation des permutations séparables

- Permutations séparables = celles possédant un arbre binaire de séparation
  - arbre binaire plan
  - feuilles π_1, π_2, ..., π_n de gauche à droite
  - à chaque noeud interne correspond un intervalle
- Remarque : 2413 et 3142 n'ont pas d'arbre de séparation



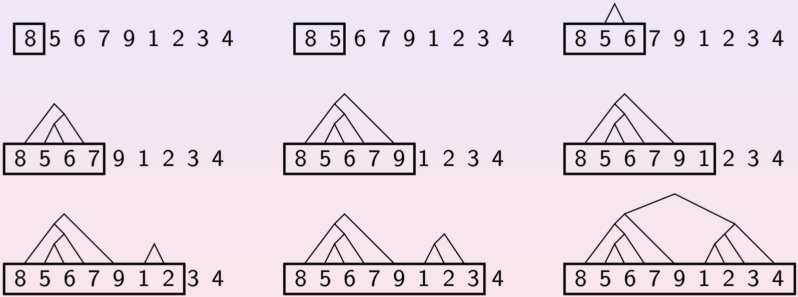
Caractérisation des permutations séparables

- Permutations séparables = celles possédant un arbre binaire de séparation
  - arbre binaire plan
  - feuilles $\pi_1, \pi_2, \ldots, \pi_n$ de gauche à droite
  - à chaque noeud interne correspond un intervalle
- Remarque : 2413 et 3142 n'ont pas d'arbre de séparation



Calcul d'un arbre binaire de séparation

Algorithme en temps linéaire [BBL98] :

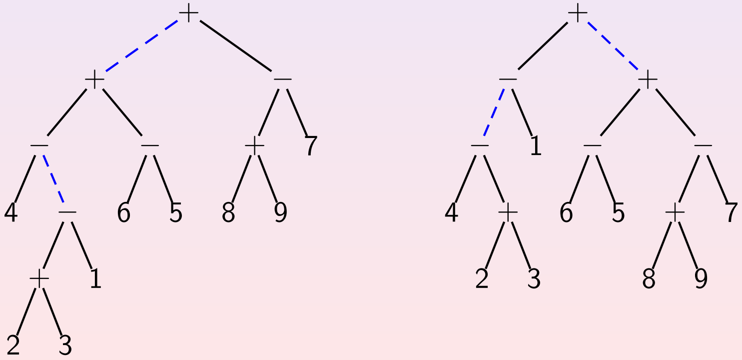


Remarque : on teste aussi en temps linéaire si une permutation est séparable ou non.

Arbre binaire de séparation

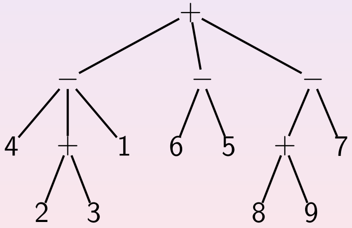
Une permutation séparable peut avoir plusieurs arbres binaires de séparation.

Exemple : $\pi = 4\ 2\ 3\ 1\ 6\ 5\ 8\ 9\ 7$



Arbre de séparation contracté

Fusion des noeuds père et fils ayant même signe.
Exemple : $\pi = 4\ 2\ 3\ 1\ 6\ 5\ 8\ 9\ 7$



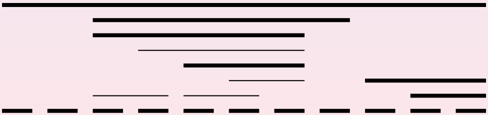
Représentant unique de toute permutation séparable par un arbre de Schröder signé (signe à la racine seulement).

Décomposition en intervalles communs

Généralisation à toutes les permutations de l'arbre de séparation : arbre de décomposition en intervalles communs.

Traduction sur les permutations de la décomposition modulaire des graphes.

Exemple : $\pi = 5\ 1\ 10\ 9\ 6\ 7\ 8\ 11\ 2\ 4\ 3$



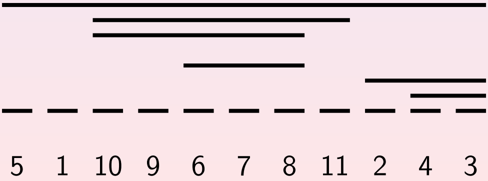
5 1 10 9 6 7 8 11 2 4 3

Décomposition en intervalles communs

Généralisation à toutes les permutations de l'arbre de séparation : arbre de décomposition en intervalles communs.

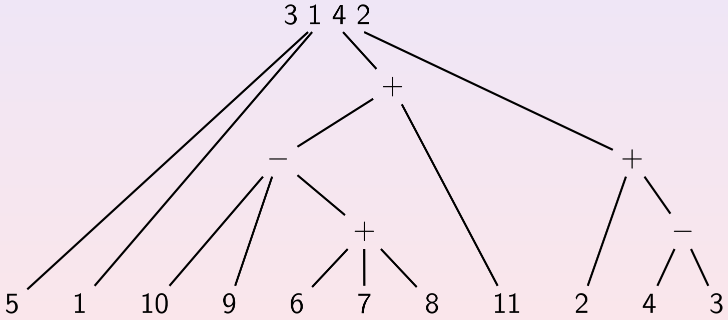
Traduction sur les permutations de la décomposition modulaire des graphes.

Exemple : $\pi = 5\ 1\ 10\ 9\ 6\ 7\ 8\ 11\ 2\ 4\ 3$



Arbre de décomposition

Exemple : $\pi = 5\ 1\ 10\ 9\ 6\ 7\ 8\ 11\ 2\ 4\ 3$



Noeuds linéaires ($+$ ou $-$) et noeuds premiers étiquetés par des permutations simples ($=$ sans intervalles communs non-triviaux)

Arbre de décomposition

Cet arbre est calculable en temps polynomial [Uno Yagiura 00].

On peut "binariser" les noeuds linéaires comme sur les arbres de séparation.

Pour une permutation séparable, arbre de séparation = arbre de décomposition.

Caractérisation des permutations séparables : celles dont l'arbre de décomposition n'a pas de noeuds premiers.

Algorithmique avec les permutations séparables

- "Perfect sorting"
- Recherche d'occurrence d'un motif
- Recherche de plus grand motif commun à deux permutations
- Recherche de plus grand motif séparable commun à K permutations

Problèmes difficiles en général, faciles sur les séparables.

"Perfect sorting" et bioinformatique

"Perfect Sorting by Reversals Is Not Always Difficult" S. Bérard, A. Bergeron, C. Chauve, C. Paul (2006)

- Motivation : trouver des scénarios d'évolution probables à partir de génômes connus.
- Problème :
  - Permutations signées : $1\ \overline{3}\ \overline{2}\ 5\ 4\ 6$
  - Renversement d'intervalles communs : renverser [2..5] donne $1\ \overline{4}\ \overline{5}\ 2\ 3\ 6$
  - Objectif : transformer π en $1\ \dots n$ ou $\overline{n} \dots \overline{1}$ en un nombre de renversements minimal
- Solution : problème exponentiel en général, polynomial sur les séparables (et même un peu plus), en utilisant l'arbre de décomposition

Recherche d'une occurrence d'un motif dans une permutation

- Motif quelconque : problème *NP*-complet
- Motif séparable : problème résolu en temps polynomial
  - Algorithme de Bose, Buss et Lubiw (BBL) : $\mathcal{O}(kn^6)$ en temps, $\mathcal{O}(kn^4)$ en espace
  - Algorithme d'Ibarra : $\mathcal{O}(kn^4)$ en temps, $\mathcal{O}(kn^3)$ en espace
- Exploiter la structure d'arbre de séparation (précalcul)
- Outil clé $=$ programmation dynamique

Algorithme BBL

- Entrée : motif τ séparable + arbre de séparation T ; permutation π
- Tableau de prog. dynamique : $M(V, i, j, a, b) = 1$ ou 0 selon qu'il existe ou non une occurrence du sous-motif de τ sous le noeud V dans $\pi_i \dots \pi_j$ utilisant des valeurs entre a et b

Exemple : V représente 21, $\pi = 6\ 4\ 2\ 5\ 3\ 1$

- $M(V, 2, 4, 3, 5) = 0$
- $M(V, 2, 5, 3, 5) = 1$
- $M(V, 2, 4, 2, 5) = 1$

Algorithme BBL

Équations de programmation dynamique :

- Feuilles : $M(V, i, j, a, b) = 1$ ssi il existe $h \in \{i, i+1, \ldots, j\}$ tel que $a \leq \pi_h \leq b$
- V positif : $M(V, i, j, a, b) = Max\{M(V_L, i, h-1, a, c-1) \cdot M(V_R, h, j, c, b) : i < h \leq j, a < c \leq b\}$
- V négatif : $M(V, i, j, a, b) = Max\{M(V_L, i, h-1, c, b) \cdot M(V_R, h, j, a, c-1) : i < h \leq j, a < c \leq b\}$

Complexité de l'algorithme BBL

- Complexité en espace : $\mathcal{O}(kn^4)$
- Pour la complexité en temps :
  - $\mathcal{O}(kn^5)$ pour les feuilles
  - $\mathcal{O}(n^2)$ pour un noeud interne à partir des tableaux des fils
  - $\mathcal{O}(kn^6)$ au total

Adapter l'algorithme de Bose, Buss et Lubiw

- Plus grand motif commun : NP-dur dans le cas général
- Si une permutation est séparable, utiliser la structure d'arbre de séparation
- Programmation dynamique : $M(V, i, j, a, b) =$ un plus grand motif commun à $\pi[V]$, π séparable et à π' quelconque, indices dans π' entre i et j, valeurs dans π' entre a et b
- Complexité en espace : $\mathcal{O}(nn'^4 min(n, n'))$
- Complexité en temps : $\mathcal{O}(nn'^6 min(n, n'))$, voire $\mathcal{O}(nn'^6)$

Comment ça marche ?

Produit de booléens $\longrightarrow$ concaténation de motifs

- Concaténation positive :
  $p \oplus p' = p_1 \cdots p_k (p'_1 + k) \cdots (p'_{k'} + k)$
- Concaténation négative :
  $p \ominus p' = (p_1 + k') \cdots (p_k + k') p'_1 \cdots p'_{k'}$

Remarque clé : la propriété d'être un plus grand motif commun est héréditaire.

Équations de programmation dynamique

- Feuille V :
  $M(V, i, j, a, b) = 1$ ou motif vide selon qu'il existe $h \in \{i, i+1, \dots, j\}$ tel que $a \leq \pi'_h \leq b$ ou pas
- Noeud V positif :
  $M(V, i, j, a, b) =$ Max de $\{M(V_L, i, j, a, b), M(V_R, i, j, a, b)\} \cup \{M(V_L, i, h-1, a, c-1) \oplus M(V_R, h, j, c, b) : i < h \leq j, a < c \leq b\}$
- Noeud V négatif :
  $M(V, i, j, a, b) =$ Max de $\{M(V_L, i, j, a, b), M(V_R, i, j, a, b)\} \cup \{M(V_L, i, h-1, c, b) \ominus M(V_R, h, j, a, c-1) : i < h \leq j, a < c \leq b\}$

Utilisation de l'arbre de décomposition

Algorithme comme avant, en ajoutant les noeuds premiers

- Pour un noeud premier d'étiquette σ, on concatène les motifs trouvés dans les fils selon l'ordre de valeurs donné par σ :

V premier ayant d fils $V_1, \ldots, V_d$. Pour calculer $M(V, i, j, a, b)$, on :

- coupe $\{i, \ldots j\}$ en d intervalles d'indices $I_1, \ldots, I_d$ de gauche à droite
- coupe $\{a, \ldots b\}$ en d intervalles de valeurs $A_1, \ldots, A_d$ (rangés par ordre croissant)
- associe l'intervalle d'indices I_k à l'intervalle de valeurs A_{σ_k}
- σ-concatène les plus grands motifs communs aux V_k et à π' utilisant l'intervalle d'indices I_k et l'intervalle de valeurs A_{σ_k} dans π'

Complexité

- Différence dans l'analyse de complexité : Pour un noeud premier d'arité d, calcul d'une case du tableau à partir des tableaux des fils en $\mathcal{O}(n'^{2d-2})$.
- Borne optimale : $d \leq n \rightarrow$ algorithme *a priori* non polynomial.
- Mais l'algorithme est polynomial pour la recherche de plus grand motif commun à une permutation dont **l'arbre de décomposition a tous ses noeuds premiers d'arité bornée par une constante** d et à une permutation quelconque.

Recherche de plus grand motif commun séparable

Problème : Trouver un motif **séparable** de taille maximale qui apparaisse dans $\pi^1, \ldots, \pi^K$

- Si K est fixé, problème résolu en temps polynomial par programmation dynamique
- Si K est arbitraire, problème NP-dur
- Ce n'est pas une bonne approximation d'un plus grand motif commun à $\pi^1, \ldots, \pi^K$

Plus généralement, pour toute classe $\mathcal{C}$ de permutations à motifs exclus, un motif commun à $\pi^1, \ldots, \pi^K$ qui est dans $\mathcal{C}$ et de taille maximale n'est pas une bonne approximation d'un plus grand motif commun à $\pi^1, \ldots, \pi^K$ ($\sqrt{OPT}$)

Programmation dynamique pour K permutations

Motif cherché séparable : obtenu par concaténations positives et négatives de motifs séparables plus petits.

Tableau de programmation dynamique M de dimension $4K$: $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K)$ contient un plus grand motif commun séparable τ à $\pi^1, \ldots, \pi^K$ t.q. τ a une occurrence dans π^q entre les indices i_q et j_q et utilisant des valeurs entre a_q et b_q.

Équation de programmation dynamique

Remplissage du tableau par $\sum_q (j_q - i_q) + (b_q - a_q)$ croissant :

- si $\exists q \in [1..K]$ tel que $i_q = j_q$ or $a_q = b_q$ alors :
  - si $\forall q \in [1..K], \exists h_q \in [i_q..j_q]$ tel que $\pi^q_{h_q} \in [a_q..b_q]$, alors $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K) \leftarrow 1$
  - sinon $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K) \leftarrow \epsilon$
- $M(i_1, j_1, a_1, b_1, \ldots, i_K, j_K, a_K, b_K) \leftarrow Longest(S_\oplus \cup S_\ominus \cup S)$ avec
  - $S_\oplus = \{M(i_1, h_1 - 1, a_1, c_1 - 1, \ldots, i_K, h_K - 1, a_K, c_K - 1) \oplus M(h_1, j_1, c_1, b_1, \ldots, h_K, j_K, c_K, b_K) :$ $i_q < h_q \leq j_q, a_q < c_q \leq b_q, \forall q \in [1..K]\}$
  - $S_\ominus = \{M(i_1, h_1 - 1, c_1, b_1, \ldots, i_K, h_K - 1, c_K, b_K) \ominus M(h_1, j_1, a_1, c_1 - 1, \ldots, h_K, j_K, a_K, c_K - 1) :$ $i_q < h_q \leq j_q, a_q < c_q \leq b_q, \forall q \in [1..K]\}$
  - $S \;\; = \{1\}$ si $\forall q \in [1..K], \exists h_q \in [i_q..j_q]$ tel que $\pi^q_{h_q} \in [a_q..b_q]$,
    $= \{\epsilon\}$ sinon

Avec un nombre K inconnu de permutations

- Le précédent algorithme est exponentiel
- En fait, le problème est NP-dur
- Preuve : réduction à partir de "Independant Set"
- Remarque : la preuve reste valable même si les permutations en entrée sont elles-même séparables

Mauvaise approximation du plus grand motif commun

$\mathcal{C}$ une classe de permutations à motifs exclus

Il existe une suite de permutations $\sigma_n \in S_n$ telles que $|\pi_n| = o(n^{0.5+\epsilon})$ avec π_n un plus grand motif de $\mathcal{C}$ qui a une occurrence dans $|\sigma_n|$

Conséquence : Rechercher le plus grand motif commun à K permutations en regardant les motifs de $\mathcal{C}$ fournit un taux d'approximation au mieux $\sqrt{OPT}$

Preuve

- Pour tout $\pi \in S_k$, le nombre de permutations $\sigma \in S_n$ qui contiennent π est au plus $(n-k)!\binom{n}{k}^2$
- Stanley-Wilf : il existe c tel que pour tout k, $|\mathcal{C}_k| \leq c^k$
- $\mathcal{C}$ est stable par motif : si $\pi \in \mathcal{C}$ et $\tau \prec \pi$, alors $\tau \in \mathcal{C}$

Conséquence : au plus $c^k(n-k)!\binom{n}{k}^2$ permutations de taille n qui contiennent un motif de $\mathcal{C}$ de taille au moins k.

Par l'absurde : si la taille minimale d'un motif de $\mathcal{C}$ contenu dans une permutations de taille n est $k = \lceil n^{0.5+\epsilon} \rceil$, alors $c^k(n-k)!\binom{n}{k}^2 = o(n!)$

Conclusion

- Problèmes de recherche de motifs dans les permutations : difficiles dans le cas général
- Problèmes *NP*-durs, mais sont-ils *NP* ?
- Restriction à des classes particulières de permutations :
  - les séparables, avec arbres de séparation : tout est plus simple
  - deux généralisations utilisant les arbres de décomposition : perfect sorting et recherche de plus grand motif commun
  - amélioration des complexités ?
- Les permutations à motifs exclus ?