Voisinage d'Arbre Evolutif Appliqué au Problème Maximum Parcimonie

Adrien Goëffon, Jean-Michel Richer and Jin-Kao Hao

LERIA - Université d'Angers, 2 bd Lavoisier, 49 045 Angers Cedex 01, France
Firstname.Name@univ-angers.fr

Abstract: *Le problème Maximum Parcimonie vise à reconstruire un arbre phylogénétique à partir de séquences ADN de manière à ce que le nombre de mutations génétiques survenues au cours de l'évolution soit minimal. Pour résoudre ce problème NP-complet, de nombreuses méthodes heuristiques ont été développées, pour la plupart basées sur la recherche locale. Ici, nous nous intéressons à l'influence de la relation de voisinage utilisée. Après avoir identifié les limites des voisinages couramment utilisés, nous introduisons le concept de voisinage évolutif. Nous montrons empiriquement qu'appliqué au problème Maximum Parcimonie, ce voisinage évolutif s'avère plus puissant et robuste que les voisinages classiques puisqu'il permet de trouver de meilleurs résultats à partir de n'importe quelle solution en moins de temps.*

Keywords: Problème Maximum Parcimonie, Voisinages d'Arbres, Recherche Locale

1 Introduction

La phylogénie peut être définie comme la reconstruction de l'évolution d'un ensemble d'espèces (ou taxons) associés à une séquence d'acides nucléiques (ADN) ou d'acides aminés (AA). Ces relations sont représentées par un arbre dit phylogénétique. Hillis [17] répertorie de nombreuses applications de la phylogénie : évolution génétique, classification et taxonomie, subdivisions des populations, variations géographiques, tests de paternité, parentés, hybridations, mise en évidence de nouvelles espèces, analyse des comportements reproducteurs, recherche virale...

Il faut remarquer que les hypothèses prises en compte dans la recherche d'un arbre d'évolution optimal ne peuvent se vérifier systématiquement lors de toute observation du monde du vivant. Par exemple on part de l'hypothèse que l'arbre le plus probable, ou le meilleur arbre, est celui qui minimise les mutations, alors qu'il est fort possible que ce ne soit pas exactement le cas dans la réalité. Il est néanmoins fortement intéressant d'obtenir une version mathématique d'un processus naturel, ce qui permet de fournir des informations que l'on n'avait pu inférer au préalable, et mettre en avant des caractéristiques originales que l'on peut confronter à l'observation.

Il existe actuellement plusieurs manières de reconstruire des arbres phylogénétiques :

Inspirées des méthodes de clustering de Sokal et Sneath [28], les *méthodes de distances* introduites en 1967 par Cavalli-Sforza et Edwards [4] et par Fitch et Margoliash [10] sont basées sur une matrice des distances observées entre les espèces deux à deux, ou bien calculée en fonction de séquences de caractères et d'un modèle de l'évolution. L'algorithme le plus connu reste le Neighbor-Joining de Saitou et Nei [25], amélioré par Gascuel [13].

Les *méthodes probabilistes* ont elles aussi recours à un modèle de l'évolution. Cependant, elles se basent sur l'analyse individuelle des caractères. La méthode du Maximum de Vraisemblance, introduite en 1981 par

Felsenstein [8], consiste à inférer la phylogénie la plus vraisemblable, c'est-à-dire maximisant la probabilité que les données se vérifient à partir de cette phylogénie et du modèle de l'évolution considéré. Bien que cette méthode connaisse de nombreux adeptes, reconnaissants de la fiabilité des arbres ainsi inférés, elle est celle qui requiert le plus d'effort calculatoire donc devient limitée pour un nombre important de taxons.

Enfin, les *méthodes cladistes* sont également basées sur une matrice de caractères donnée. La plus utilisée est celle du Maximum de Parcimonie, dont les premières réflexions sont à mettre au crédit de Edwards et Cavalli-Sforza [6]. Elle vise à retrouver la phylogénie qui minimise le nombre d'évènements évolutifs (score), sans recourir à un modèle de l'évolution. En outre, cette méthode permet d'attribuer à chaque ancêtre hypothétique (noeud interne de l'arbre), les états possibles pris pour chaque caractère.

Le problème Maximum Parcimonie (MP) est équivalent au problème de l'arbre de Steiner dans un hypercube. MP est NP-complet, comme l'ont montré Foulds et Graham en 1982 [11]. L'approche utilisée pour l'approximation du problème consiste à utiliser des algorithmes heuristiques dans le but de trouver le plus rapidement possible un arbre d'un score très proche de celui d'une solution optimale. De nombreux travaux de qualité traitent du problème MP, notamment ceux de Goloboff [15] ou Nixon [21]. Mais comme on peut le remarquer avec Roshan *et al.* [24], les logiciels actuels ne sont pas encore suffisamment performants et rapides dès lors que les instances à traiter contiennent des milliers d'espèces.

Considérant les très larges espaces de recherche, il se vérifie empiriquement que des heuristiques de recherche locale stochastique sont particulièrement adaptées au problème MP, à la condition d'utiliser un voisinage approprié [12].

Il existe majoritairement dans la littérature trois voisinages d'arbres : NNI, SPR et TBR. Chaque recherche locale associée gagne en efficacité ou en rapidité suivant le voisinage utilisé. Notre démarche consiste à combiner les propriétés de ces voisinages intéressants afin d'obtenir une recherche locale à la fois rapide, efficace et robuste. Nous introduisons ici le concept de voisinage évolutif en tant que nouvelle approche pour la résolution du problème MP. Tous les tests effectués montrent le réel gain apporté par cette technique d'un point de vue efficacité et temps de calcul, et surtout sa capacité à converger très vite vers une solution de confiance, en évitant les pièges des optima locaux.

Après avoir rappelé brièvement le problème MP, nous présentons dans la section 3 les voisinages d'arbres connus puis les méthodes de recherche locale qui leur sont souvent associées. Nous discutons ensuite des limites de ces voisinages et proposons une alternative pour y remédier. Ce voisinage évolutif est introduit puis décrit plus formellement dans la section 4. Des résultats expérimentaux sont ensuite présentés afin de comparer la performance et le comportement de ce voisinage par rapport aux deux voisinages classiques NNI et SPR.

2 Le problème Maximum Parcimonie

Comme évoqué dans l'introduction, le problème MP consiste à partir d'un ensemble de séquences, à retrouver la phylogénie optimale au sens du critère de parcimonie, c'est-à-dire un arbre dont les feuilles sont associées aux séquences et qui minimise le nombre de mutations.

Afin de fixer plus précisément les idées, rappelons quelques définitions.

Definition 1. *La* distance de Hamming $H(x, y)$ *entre deux séquences* $x = (x_1, x_2, \ldots, x_k)$ *et* $y = (y_1, y_2, \ldots, y_k)$ *est égale à* $|\{i : x_i \neq y_i\}|$.

Definition 2. *Le* score de parcimonie *d'un arbre* $T = (V, E)$ *dont chaque noeud* v *est étiqueté par une séquence* s^v *de longueur* k *sur un alphabet* Σ *est la somme des distances de Hamming des séquences étiquetant chaque couple de noeuds séparés par une arête dans* T.

Etant donné un arbre T dont les feuilles sont bijectivement étiquetées par les séquences de S, Fitch a formalisé un algorithme polynomial [9] qui calcule des séquences hypothétiques (assignées aux noeuds internes de l'arbre) et le score de parcimonie de telle sorte que celui-ci soit minimal.

Le but du problème MP est de trouver un arbre dont le score de parcimonie est le plus faible parmi tous les arbres phylogénétiques possibles pour un ensemble S de séquences. MP peut alors être formulé comme un problème combinatoire de minimisation $(\mathcal{T}, f)$ tel que :

1. l'espace de recherche $\mathcal{T}$ est défini par l'ensemble de toutes les configurations possibles ($|\mathcal{T}| = \prod_{i=3}^{|S|}(2i-3)$ [26])
2. la fonction de coût $f : \mathcal{T} \rightarrow I\!N$ est telle que $\forall T \in \mathcal{T}$, $f(T) = \sum_{(x,y)\in E} H(x,y)$, i.e. le score de parcimonie de T.

3 Recherche locale et voisinages

3.1 La descente

La méthode de descente consiste à générer une première phylogénie, puis à rechercher une phylogénie voisine (au sens d'une relation de voisinage) dont le score est inférieur, et ainsi de suite jusqu'à ce que la phylogénie courante n'ait aucun voisin dont le score soit strictement inférieur. La solution finale est alors un optimum local, qui n'est pas nécessairement un optimum global.

Cette approche de descente, qui est la méthode de recherche locale la plus simple, dépend essentiellement de la relation de voisinage à laquelle elle est associée. Même s'il existe de nombreuses techniques pour tenter d'améliorer la qualité des solutions fournies par les algorithmes de descente, ces derniers sont à la base de toutes les meilleures méthodes de résolution actuelles.

3.2 Voisinages NNI, SPR et TBR

Une relation de voisinage structure l'espace de recherche sur lequel une méthode de recherche locale (par exemple descente) est appliquée. Les trois relations de voisinage d'arbres que l'on retrouve systématiquement dans la littérature sont NNI, SPR et TBR.

NNI (*Nearest Neighbor Interchange*) [30] consiste à échanger deux branches adjacentes de l'arbre. C'est un voisinage restreint de taille linéaire par rapport à la taille de l'arbre, car un arbre à N feuilles compte $2N-6$ voisins [23].

SPR (*Subtree Pruning Regrafting*) [29] est une stratégie qui coupe une branche et la réinsère à un autre endroit de l'arbre. A partir d'un arbre, il existe $2(N-3)(2N-7)$ réarrangements SPR possibles [2], c'est un voisinage de taille quadratique.

Enfin, TBR (*Tree-Bisection-Reconnection*) [29] est un voisinage plus large qui casse l'arbre en deux sous-arbres qui seront reconnectés à partir d'une de leurs arêtes. Ici, le nombre de voisins dépend de la topologie de l'arbre, mais il est d'au moins $(2N-3)(N-3)^2$ [2].

On peut remarquer que $NNI \subseteq SPR \subseteq TBR$ [20].

3.3 Propriétés et limites des voisinages existants

Une relation de voisinage réduite comme NNI possède l'avantage de favoriser la recherche à grande échelle en ne permettant que des modifications très locales sur l'arbre. Calculer la variation de coût engendrée par une transformation NNI est d'autant plus rapide que l'arbre résultant est très proche, et parcourir l'ensemble des voisins d'une configuration est également plus rapide qu'avec un voisinage plus large, car le nombre de voisins à explorer est plus petit.

En revanche, une recherche locale sur un tel espace de recherche aura une faible capacité à améliorer sensiblement le coût d'une solution sur quelques pas. De plus, étant donné le faible nombre de voisins, les optima locaux seront plus fréquents sur des solutions pas nécessairement proches de l'optimum en terme de coût.

A l'opposé, une relation de voisinage large comme TBR sera très coûteuse d'un point de vue calculatoire. Explorer tout le voisinage d'une configuration prend beaucoup de temps, et les arbres voisins subissent d'importantes modifications topologiques. Ainsi, moins d'information peut être conservée pour le recalcul du score de parcimonie, même si Goloboff [14] propose une méthode qui vise à réduire la complexité du recalcul du score.

3.4 Espace de recherche variable

Pour sortir des optima locaux, les méthodes actuelles proposent des alternatives, mais sans remettre en question les voisinages susmentionnés.

Ainsi, la méthode de Nixon [21], maintenant utilisée dans beaucoup de logiciels de MP, modifie la fonction d'évaluation par bruitage de la matrice de caractères lorsque la recherche locale s'enlise, afin de perturber la solution courante tout en continuant à se déplacer dans un espace de recherche possédant exactement la même structure (seul le poids des arêtes a changé, si l'on considère l'espace de recherche comme un graphe étiqueté par les variations de coût entre les arbres).

Une autre méthode utilisée par Ribeiro *et al.* [1], [22] consiste à considérer un ensemble de relations de voisinage imbriquées (par exemple $\{NNI, SPR\}$ ou bien $\{SPR, 2-SPR\}$, $k-SPR$ étant la relation de voisinage induite par k pas de SPR), et à les utiliser successivement. Il s'agit d'une application au problème de la métaheuristique VNS (Variable Neighborhood Search), proposée par Hansen et Mladenovic [16].

L'efficacité d'une méthode de type VNS semble toutefois limitée. Si l'on part du principe que la solution initiale est suffisamment éloignée de l'optimum, la première recherche locale (avec le voisinage le plus restreint de l'ensemble) rencontrera un optimum local plus rapidement. Même si prendre un voisinage étendu pour la suite de la recherche va permettre une amélioration de la solution, il est également probable qu'une grande partie de l'effort calculatoire va être perdu en envisageant plus de choix qui se révèleront inutiles (i.e. proposer des recombinaisons de plus en plus éloignées de la topologie courante et donc perdre une plus grosse partie de l'information acquise durant la première phase de la recherche locale).

4 Voisinage évolutif

4.1 Principe général

Afin de combiner les propriétés intéressantes des voisinages larges et faibles, nous proposons d'effectuer une recherche locale sur un espace de recherche qui s'élargit ou se rétracte en fonction de l'avancée de la recherche, et de la fréquence d'apparition de voisins pertinents.

Contrairement à la méthode VNS, partir du voisinage le plus large peut s'avérer pertinent, en construisant les bases de la topologie de la future solution. Evaluer plus de voisins (avec des modifications plus sensibles) en début de recherche va permettre d'améliorer grandement le coût des solutions dès les premiers pas de la recherche locale, grâce à une recherche plus intensive. En fin de recherche, on peut imaginer n'intervenir que très localement sur la topologie de l'arbre. On peut obtenir ce schéma en réduisant petit à petit l'étendue du voisinage exploré au fil de la recherche.

L'idée est donc de définir une relation de voisinage paramétrique qui évolue dans le temps, soit de manière prédéfinie, soit de manière réactive en prenant en compte les informations sur la qualité de l'ensemble des voisins visités. Nous allons maintenant définir un schéma simple qui utilise ce concept pour le problème MP.

4.2 Un exemple de voisinage évolutif

Utilisation de la propriété $NNI \subseteq SPR$ A titre d'exemple, nous prenons deux voisinages $\mathcal{N}^1$ et $\mathcal{N}^2$ tels que $\mathcal{N}^2 \subseteq \mathcal{N}^1$, de sorte que nous puissions définir plus simplement un voisinage paramétrique $\mathcal{N}_d$ qui généralise $\mathcal{N}^1$ et $\mathcal{N}^2$.

Prenons $\mathcal{N}^1 = \mathcal{N}^{SPR}$ et $\mathcal{N}^2 = \mathcal{N}^{NNI}$. Avec SPR, on dégrafe une branche de l'arbre et on la reconnecte ailleurs, sans contrainte particulière si ce n'est d'obtenir un arbre valide et distinct. On peut voir NNI comme un SPR particulier, où une branche doit être insérée sur une arête voisine d'où elle provient dans l'arbre courant.

Par extension, nous imaginons alors un voisinage de type SPR où la distance entre l'arête supprimée et l'arête insérée soit contrainte. Si cette distance est maximale, alors il s'agit du SPR, sans contrainte. Si celle-ci elle minimale, alors nous nous retrouvons dans le cas NNI.

Un voisinage paramétrique Soit $f^{SPR} : (\mathcal{T}, V, V) \rightarrow \mathcal{T}$ la transformation telle que $f^{SPR}(T, v_i, v_j)$ soit l'arbre obtenu en dégrafant dans $T = (V, E)$ le sous-arbre de racine v_i et en l'insérant entre v_j et son ascendant direct. Alors $\mathcal{N}^{SPR}(T) = \{T' \in \mathcal{T} | \exists (v_i, v_j) \in V^2, f^{SPR}(T, v_i, v_j) = T'\}$.

Pour contraindre SPR, nous introduisons un paramètre d, tel que $\mathcal{N}_d^{SPR}(T)$ représente l'ensemble des arbres obtenus par transformation $f^{SPR}(T, v_i, v_j)$ et dont v_i et v_j sont *distants* de d au maximum.

On note $\delta(v_i, v_j)$ la *distance* entre v_i et v_j, comme étant égale à la longueur du chemin élémentaire entre les ascendants respectifs de v_i et v_j, -1 si le chemin contient la racine (si l'on travaille sur des arbres enracinés). Ainsi, deux noeuds frères sont distants de 0, et la distance reste la même dans le cas d'arbres non enracinés.

Puisque l'on souhaite maîtriser la taille du voisinage durant la recherche, on définit le voisinage $\mathcal{N}_d^{SPR}(T)$ comme étant l'ensemble des voisins $\mathcal{N}^{SPR}(T)$ tels que la distance entre l'arête supprimée et l'arête insérée (qui est égale à la distance δ entre leurs deux noeuds fils) n'excède pas le paramètre d. En d'autres termes, $\mathcal{N}_d^{SPR}(T) = \{T' \in \mathcal{T} | \exists (v_i, v_j) \in V^2, f^{SPR}(T, v_i, v_j) = T' \wedge \delta(v_i, v_j) \leq d\}$.

4.3 Schéma simple de voisinage évolutif

Réduire d durant la recherche permet de débuter avec un voisinage quadratique (SPR) et de terminer avec le voisinage NNI, linéaire par rapport au nombre de noeuds. On calcule ainsi les valeurs initiales et finales de d :

$$\begin{cases} \mathcal{N}_{d_{init}}^{SPR} \equiv \mathcal{N}^{SPR} \\ \mathcal{N}_{d_{final}}^{SPR} \equiv \mathcal{N}^{NNI} \end{cases} \Rightarrow \begin{pmatrix} d_{init} \\ d_{final} \end{pmatrix} = \begin{pmatrix} \max_{V^2} \delta(v_i, v_j) \\ 1 \end{pmatrix}$$

d_{init} correspond au plus petit majorant des distances entre noeuds, qui est égale à la plus grande distance entre les feuilles de l'arbre deux à deux.

Si l'on réduit d de manière linéaire, et si M est le nombre d'itérations de recherche locale prévus, alors le paramètre d du voisinage $\mathcal{N}_d^{SPR}$ à la i-ème itération de recherche locale est égal à $\lfloor d_{init}\left(1-\frac{i}{M}\right)\rfloor$.

A chaque itération de recherche locale, le choix d'un voisin $f^{SPR}(T, v_i, v_j)$ (avec $\delta(v_i, v_j) \leq d$) se fait aléatoirement mais selon une distribution non uniforme. Pour des temps de calcul plus faibles et une meilleure efficacité, on choisit tout d'abord aléatoirement une distance d' (comprise entre 1 et d) et un noeud v_i. On recherche ensuite un noeud v_j en parcourant un chemin élémentaire aléatoire de longueur $d'+2$. Si durant le parcours on est bloqué sur une feuille de l'arbre, v_j prend la valeur de cette feuille même si $\delta(v_i, v_j) \leq d'$. Ainsi les feuilles de l'arbre ont une probabilité plus importante d'être sélectionnées, qui est fonction de leur distance avec la racine du sous-arbre à dégrafer.

Dans la section suivante, nous évaluons l'influence du voisinage évolutif $\mathcal{N}_d^{SPR}$ par rapport à leurs voisinages fixes associés $\mathcal{N}^{SPR}$ et $\mathcal{N}^{NNI}$.

5 Premières expérimentations

5.1 Benchmarks

Pour réaliser nos tests, nous avons utilisé des benchmarks aléatoires, mais également des instances issues de données réelles.

Les instances aléatoires ont été générées avec *dnatree* [19] utilisant le modèle de Kimura à 2 paramètres [18]. 300 instances, toutes générées suivant des paramètres différents, ont été utilisées durant les tests. Nous avons observé que la tendance des résultats variait très peu d'une instance à l'autre. Parmi elles et pour des raisons de lisibilité, nous en avons ici choisi 6 de tailles diverses, afin d'exhiber un échantillon représentatif. Elles comportent 100, 300 ou 500 séquences ADN, courtes (100 acides nucléiques) ou plus longues (1000). Le taux de transition-transversion est fixé à 2 et la probabilité de mutation par unité de temps à 5%. Leurs noms dans les tableaux de résultats sont composés du nombre de séquences et de leur longueur (100-100, 300-100, 500-100, 100-1000, 300-1000, 500-1000).

Dans le cadre d'une étude sur la diversité génétique d'une bactérie phytopathogène [7], le laboratoire de Pathologie Végétale de l'INRA d'Angers nous a fourni plusieurs jeux correspondant à des alignements de séquences courtes sur différents gènes de bactéries. L'instance reportée ici est le résultat de la concaténation des séquences de tous les gènes. Au final, il s'agit d'un ensemble de 44 séquences composées de 453 nucléotides (nommé *phyto* dans le tableau 5).

Enfin, nous avons testé notre voisinage sur l'instance *zilla* [5] majoritairement utilisée dans la littérature et réputée très difficile, composée de 500 séquences de 759 caractères.

5.2 Conditions d'expérimentation

Nous utilisons un algorithme de descente stricte sur lequel nous testons les trois voisinages : $\mathcal{N}^{SPR}$, $\mathcal{N}^{NNI}$ et $\mathcal{N}_d^{SPR}$ pour le Voisinage Evolutif tel qu'il est décrit dans la section 4. L'idée intuitive du voisinage évolutif $\mathcal{N}_d^{SPR}$ est de combiner l'efficacité de $\mathcal{N}^{SPR}$ et la rapidité de $\mathcal{N}^{NNI}$. Nous voulons vérifier ces points sur les jeux de tests utilisés. Puisque $\mathcal{N}^{NNI} \subseteq \mathcal{N}_d^{SPR} \subseteq \mathcal{N}^{SPR}$, la descente classique (qui retourne obligatoirement un optimum local) utilisant $\mathcal{N}^{SPR}$ retournera majoritairement des solutions de coût meilleur ou égal, car un

optimum local au sens d'un voisinage l'est également pour tous ses voisinages inclus, alors que la réciproque est fausse. Mais déterminer un optimum local nécessite en particulier d'avoir calculé tous ses voisins.

Le tableau 1 nous indique le nombre de voisins $|\mathcal{N}(T)|$ d'un arbre T en fonction du nombre de séquences N (voir section 3.2), pour les deux voisinages à taille fixe utilisés.

N	$\lvert\mathcal{N}^{NNI}(T)\rvert$	$\lvert\mathcal{N}^{SPR}(T)\rvert$	$\lvert\mathcal{T}\rvert$
44	82	6642	$4,6.10^{80}$
100	194	37 442	$3,3.10^{184}$
300	594	352 242	$3,4.10^{700}$
500	994	987 042	$1,0.10^{1280}$

Table 1. Taille des voisinages et de l'espace de recherche en fonction du nombre de séquences

On remarque clairement que l'effort calculatoire à fournir pour trouver un optimum local n'est pas le même.

Pour mesurer efficacement l'influence du voisinage sur la qualité de la solution retournée pour un effort calculatoire équivalent (et donc des temps de calcul plus raisonnables), nous fixons un nombre maximal M d'itérations de recherche locale. Dans nos expérimentations et pour les instances aléatoires moyennes (100 ou 300 séquences), on fixe M à 50 000, c'est-à-dire que 50 000 arbres au maximum seront évalués. Pour l'instance réelle, plus petite, un maximum de 10 000 itérations est utilisé pour mesurer l'efficacité des voisinages. Pour les instances larges (500 espèces), nous ferons varier le paramètre M.

Le pseudo-code suivant (Algorithme 1) montre la procédure utilisée pour nos tests. Il s'agit d'un algorithme classique de descente avec la méthode *First Improve*, où l'on parcourt le voisinage d'une solution courante (dans un ordre aléatoire) jusqu'à ce que l'on trouve un voisin qui l'améliore.

Algorithm 1 Algorithme de descente avec nombre d'itérations fixé

Entrée: N séquences alignées définissant un problème de minimisation $(\mathcal{T}, f)$
Sortie: Le meilleur arbre trouvé

Générer un arbre initial $T \in \mathcal{T}$, aléatoirement ($\mathcal{A}$) ou selon une méthode des distances ($\mathcal{D}$)
nbIter = 1
Tant que le nombre d'itérations maximal n'est pas atteint (*nbIter* $\leq M$) et que T n'est pas de manière certaine un optimum local (tous les voisins de T n'ont pas déjà été évalués) **faire**
1. Générer un voisin $T' \in \mathcal{N}(T)$ de l'arbre courant T selon la relation de voisinage $\mathcal{N}$
2. Calculer le score de T' selon la fonction de coût f (score de parcimonie)
3. $T = T'$ si $f(T') < f(T)$
4. *nbIter* = *nbIter* + 1

Un voisin qui a déjà été évalué depuis la dernière amélioration ne peut pas être proposé à nouveau. Ainsi, il est possible que plus aucun voisin ne puisse être évalué avant les M itérations. Dans ce cas, la solution courante est un optimum local et nous la retournons.

Pour chaque instance courte ou moyenne, nous avons lancé 10 fois les descentes associées à toutes les combinaisons *Construction+Voisinage*. La méthode de construction $\mathcal{A}$ génère aléatoirement un arbre initial, tandis que $\mathcal{D}$ construit un premier arbre selon une méthode de clustering basée sur les distances de hamming entre séquences initiales (inspirée de UPGMA [27]), mais stochastique. Cette méthode construit des arbres de score bien meilleur qu'en prenant un arbre aléatoire, et ainsi on pourra observer le comportement de la descente avec $\mathcal{N}^{SPR}$, $\mathcal{N}^{NNI}$ ou $\mathcal{N}_d^{SPR}$ en fonction de la qualité de la solution initiale. Les 6 combinaisons reportées sont alors $\mathcal{A}$+$\mathcal{N}^{SPR}$, $\mathcal{A}$+$\mathcal{N}^{NNI}$, $\mathcal{A}$+$\mathcal{N}_d^{SPR}$, $\mathcal{D}$+$\mathcal{N}^{SPR}$, $\mathcal{D}$+$\mathcal{N}^{NNI}$ et $\mathcal{D}$+$\mathcal{N}_d^{SPR}$.

Pour les instances larges, nous avons lancé 20 fois chaque descente, à partir d'un arbre aléatoire pour les instances aléatoires (afin de mesurer la capacité des méthodes à converger vers des solutions proches depuis n'importe quel point de départ sur l'espace de recherche), et à partir d'un arbre construit selon une méthode de distances pour l'instance *zilla* (sur cette instance très difficile, il est nécessaire de débuter la recherche du meilleur arbre dès la construction de la solution initiale).

5.3 Résultats

Les tableaux de résultats contiennent, pour chaque instance et chaque méthode, le score minimum f_b, le score moyen f_m et son écart-type σ sur l'ensemble des essais, ainsi que le *temps* d'exécution moyen en secondes. Nous précisons également le score moyen des arbres initiaux f_{init} suivant la méthode de construction $\mathcal{A}$ ou $\mathcal{D}$.

		f_{init}	f_b	f_m	σ	*temps*
100-100	$\mathcal{A}$+$\mathcal{N}^{SPR}$	*1 567*	536	538,7	2,1	68
	$\mathcal{A}$+$\mathcal{N}^{NNI}$		544	568,8	21,6	8
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$		**534**	**534,0**	0	39
	$\mathcal{D}$+$\mathcal{N}^{SPR}$	*557*	**534**	534,6	0,5	60
	$\mathcal{D}$+$\mathcal{N}^{NNI}$		**534**	536,1	1,4	3
	$\mathcal{D}$+$\mathcal{N}_d^{SPR}$		**534**	**534,0**	0	10
100-1000	$\mathcal{A}$+$\mathcal{N}^{SPR}$	*12 939*	**4 080**	4 106,8	18,1	671
	$\mathcal{A}$+$\mathcal{N}^{NNI}$		**4 080**	4 178,8	197,1	109
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$		**4 080**	**4 080,0**	0	147
	$\mathcal{D}$+$\mathcal{N}^{SPR}$	*4 108*	**4 080**	4 081,8	2,0	539
	$\mathcal{D}$+$\mathcal{N}^{NNI}$		**4 080**	**4 080,0**	0	20
	$\mathcal{D}$+$\mathcal{N}_d^{SPR}$		**4 080**	**4 080,0**	0	52

Table 2. Comparaison entre SPR, NNI et le Voisinage Evolutif pour $N = 100$

Jeux aléatoires Sur les 2 instances composées de 100 séquences (tableau 2), on remarque que $\mathcal{N}_d^{SPR}$ retourne systématiquement une solution de score minimum (par rapport à l'ensemble des méthodes), quelle que soit la taille des séquences et la solution initiale de la recherche. Comme on pouvait s'y attendre, $\mathcal{N}^{NNI}$ est la méthode la plus rapide mais n'est pas du tout fiable lorsque la solution initiale est construite aléatoirement, et obtient de meilleurs résultats pour de longues séquences. Tous les voisinages semblent performants et fiables à partir d'un arbre construit selon une méthode de distances, mais cela est surtout dû à la relative facilité de cette instance (seulement 100 séquences), et donc à la bonne qualité de la solution initiale (pour 100-1000, la recherche locale ne permet qu'une amélioration de la solution construite de $-0,7\%$, contre $-68,5\%$ depuis un arbre aléatoire).

Lorsque le nombre de séquences est plus important, comme pour l'instance reportée dans le tableau 3, le voisinage $\mathcal{N}^{SPR}$ n'est plus approprié, puisqu'il retourne dans tous les cas des solutions de score très éloigné de ceux des solutions trouvées par $\mathcal{N}_d^{SPR}$. La qualité de la solution retournée avec $\mathcal{N}^{NNI}$ est fortement dépendante de l'arbre initial et peut converger très rapidement vers un optimum local, ce qui montre l'instabilité de la méthode en fonction des facteurs stochastiques. Ces résultats confirment que $\mathcal{N}^{NNI}$ n'est vraiment efficace (en terme de performance, robustesse et temps de calcul) que lorsque les séquences sont longues et à condition de débuter la recherche avec une *bonne* solution (ici construite suivant une méthode basée sur les distances). Notre voisinage évolutif $\mathcal{N}_d^{SPR}$ obtient de bons résultats depuis toute solution initiale, malgré le nombre réduit d'itérations comparé à la taille du problème.

		f_{init}	f_b	f_m	σ	*temps*
300-100	$\mathcal{A}$+$\mathcal{N}^{SPR}$	*5 860*	1 579	1 647,9	32,3	115
	$\mathcal{A}$+$\mathcal{N}^{NNI}$		1 746	1 921,9	92,2	53
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$		1 304	1 310,8	7,0	51
	$\mathcal{D}$+$\mathcal{N}^{SPR}$	*1 375*	1 336	1 342,4	4,1	77
	$\mathcal{D}$+$\mathcal{N}^{NNI}$		1 303	1 305,6	3,7	54
	$\mathcal{D}$+$\mathcal{N}_d^{SPR}$		**1 302**	**1 303,4**	1,5	53
300-1000	$\mathcal{A}$+$\mathcal{N}^{SPR}$	*51 391*	17 043	17 305,0	211,3	1 128
	$\mathcal{A}$+$\mathcal{N}^{NNI}$		**14 209**	14 426,0	279,8	467
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$		**14 209**	**14 209,0**	0	479
	$\mathcal{D}$+$\mathcal{N}^{SPR}$	*14 294*	14 266	14 270,8	3,7	697
	$\mathcal{D}$+$\mathcal{N}^{NNI}$		**14 209**	**14 209,0**	0	82
	$\mathcal{D}$+$\mathcal{N}_d^{SPR}$		**14 209**	**14 209,0**	0	364

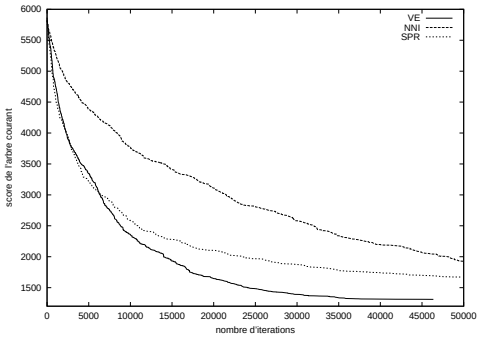

Table 3. Comparaison entre les voisinages pour $N = 300$ et Recherche Locale à partir d'un arbre aléatoire

La figure associée montre le score de l'arbre courant en fonction de l'avancée de la recherche, pour l'instance 300-100 et en partant d'un arbre aléatoire (la recherche reportée est celle retournant l'arbre de score médian). La recherche locale à voisinage évolutif (VE) domine clairement SPR et NNI.

		f_{init}	M	f_b	f_m	σ	*temps*
500-100	$\mathcal{A}$+$\mathcal{N}^{SPR}$	*9 165*	5.10^4	3 548	3 610,5	48,7	4'
	$\mathcal{A}$+$\mathcal{N}^{NNI}$			3 937	4 318,1	210,6	2'
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$			**2 305**	**2 435,1**	121,6	1'40
	$\mathcal{A}$+$\mathcal{N}^{SPR}$		10^5	2 940	3 049,4	47,8	8'
	$\mathcal{A}$+$\mathcal{N}^{NNI}$			2 808	3 042,1	199,6	4'
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$			**2 243**	**2 251,7**	10,2	3'
	$\mathcal{A}$+$\mathcal{N}^{SPR}$		$1,5.10^5$	2 753	2 795,5	28,1	11'
	$\mathcal{A}$+$\mathcal{N}^{NNI}$			2 389	2 559,1	192,8	6'
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$			**2 243**	**2 244,7**	1,4	4'
500-1000	$\mathcal{A}$+$\mathcal{N}^{SPR}$	*100 434*	5.10^4	36 505	37 595,6	595,1	39'
	$\mathcal{A}$+$\mathcal{N}^{NNI}$			28 478	30 072,0	1 374,8	20'
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$			**24 337**	**24 491,7**	184,7	16'
	$\mathcal{A}$+$\mathcal{N}^{SPR}$		10^5	31 615	32 029,7	319,4	75'
	$\mathcal{A}$+$\mathcal{N}^{NNI}$			**24 319**	24 460,3	188,9	29'
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$			**24 319**	**24 319**	0	33'
	$\mathcal{A}$+$\mathcal{N}^{SPR}$		$1,5.10^5$	28 961	29 490,7	407,6	116'
	$\mathcal{A}$+$\mathcal{N}^{NNI}$			**24 319**	24 460,3	188,9	29'
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$			**24 319**	**24 319**	0	44'

Table 4. Comparaison entre les voisinages pour $N = 500$ à partir d'une solution initiale aléatoire

Pour une instance très large (tableau 4), il se confirme que $\mathcal{N}^{SPR}$ donne dans tous les cas de mauvais résultats. Les scores retournés par $\mathcal{N}^{NNI}$ sont très inconstants, avec des écarts-types toujours très importants. Si un allongement du temps de recherche provoque une amélioration des performances de $\mathcal{N}^{SPR}$, ce n'est rapidement plus le cas pour $\mathcal{N}^{NNI}$, qui retourne très tôt un optimum local.

On remarque particulièrement l'efficacité du voisinage évolutif $\mathcal{N}_d^{SPR}$ dans le cas de larges instances, comme 500-1000 qui est constituée au total de 500 000 caractères. En 100 000 itérations, il parvient à retourner systématiquement (sur les 20 tests), une solution de score égal au meilleur score trouvé pour cette instance (24 319). Sur les 20 tests, $\mathcal{N}^{NNI}$ n'a trouvé qu'une seule fois un tel score.

Dans le tableau 4, les temps d'exécution sont donnés en minutes.

		f_{init}	f_b	f_m	σ	*temps*
phyto	$\mathcal{A}$+$\mathcal{N}^{SPR}$	*602*	**226**	229,0	7,0	44
	$\mathcal{A}$+$\mathcal{N}^{NNI}$		**226**	291,3	59,1	9
	$\mathcal{A}$+$\mathcal{N}_d^{SPR}$		**226**	**226,0**	0	9
	$\mathcal{D}$+$\mathcal{N}^{SPR}$	*231*	**226**	226,6	0,7	44
	$\mathcal{D}$+$\mathcal{N}^{NNI}$		**226**	226,9	0,7	5
	$\mathcal{D}$+$\mathcal{N}_d^{SPR}$		**226**	**226,0**	0	10

Table 5. Performance des voisinages sur un problème réel

Jeux réels Si l'on ajoute aux précédents résultats ceux obtenus avec l'instance réelle (tableau 5), de constitution différente des instances aléatoires, il apparait que le voisinage évolutif permet d'obtenir des solutions de score minimal avec une confiance maximale et avec de meilleurs temps de calcul qu'en utilisant le classique $\mathcal{N}^{SPR}$.

On peut considérer que les instances précédentes ne comportent pas de difficulté particulière, puisqu'à partir d'un certain nombre de lancers, un score minimum est ressorti des expérimentations. Avec l'instance *zilla* bien connue, trouver le score minimum calculé à ce jour (16 218 [21]) en un faible nombre d'itérations et avec un unique arbre de départ (une *réplication*) est peu probable. Ce test sera alors une bonne manière de valider nos résultats, et surtout de comparer les performances de notre voisinage évolutif avec $\mathcal{N}^{NNI}$ dans le cas d'un démarrage à partir d'une solution construite ($\mathcal{D}$). C'est en effet la seule de ces méthodes qui offre sur les jeux utilisés précédemment d'aussi bons résultats que le voisinage évolutif ($\mathcal{N}^{NNI}$ est plus rapide, mais le résultat est moins fiable car il dépend toujours de la solution initiale et reste sujet aux aléas stochastiques).

Nous testons les trois voisinages pour différents nombres maximaux d'itérations M : 10^5, 2.10^5 et 3.10^5. Les temps d'exécution sont ici donnés en minutes.

		f_{init}	M	f_b	f_m	σ	*temps*
zilla	$\mathcal{D}$+$\mathcal{N}^{SPR}$	*18 353*	10^5	17 089	17 116,8	24,8	145'
	$\mathcal{D}$+$\mathcal{N}^{NNI}$			16 556	16 778,9	119,2	19'
	$\mathcal{D}$+$\mathcal{N}_d^{SPR}$			**16 306**	**16 356,5**	47,8	27'
	$\mathcal{D}$+$\mathcal{N}^{SPR}$		2.10^5	16 785	16 816,8	36,6	278'
	$\mathcal{D}$+$\mathcal{N}^{NNI}$			16 556	16 778,9	119,2	19'
	$\mathcal{D}$+$\mathcal{N}_d^{SPR}$			**16 282**	**16 297,9**	9,8	57'
	$\mathcal{D}$+$\mathcal{N}^{SPR}$		3.10^5	16 590	16 645,3	57,5	395'
	$\mathcal{D}$+$\mathcal{N}^{NNI}$			16 556	16 778,9	119,2	19'
	$\mathcal{D}$+$\mathcal{N}_d^{SPR}$			**16 277**	**16 296,3**	18,0	77'

Table 6. Descentes sur l'instance *zilla* à partir d'une solution construite

Pour $\mathcal{N}^{SPR}$ et $\mathcal{N}^{NNI}$, nous avons lancé 20 fois chaque méthode mais uniquement sur 3.10^5 itérations. Les valeurs pour 10^5 et 2.10^5 itérations sont données par les résultats intermédiaires des recherches locales. $\mathcal{N}^{NNI}$ retourne systématiquement un optimum local en moins de 100 000 itérations (50 000 en moyenne), c'est pour cette raison que les résultats sont identiques pour un M supérieur. Sur cette instance difficile, on remarque immédiatement la puissance du voisinage évolutif. En 200 000 ou 300 000 itérations, l'écart entre les arbres retournés et le meilleur arbre connu à ce jour varie entre $0,36\%$ et $0,69\%$, ($0,49\%$ en moyenne), tandis qu'il varie entre $2,08\%$ et $4,33\%$ ($3,31\%$ en moyenne, soit près de 7 fois plus) pour les voisinages connus.

La figure 1 montre le comportement des trois recherches ayant retourné le *score médian* pour les trois méthodes appliquées à l'instance *zilla* sur respectivement 100 000 et 300 000 itérations. Nous rappelons que l'arbre initial est construit à partir de la méthode basée sur les distances.

On remarque clairement la supériorité du voisinage évolutif $\mathcal{N}_d^{SPR}$, noté VE sur les figures, par rapport à SPR et NNI.

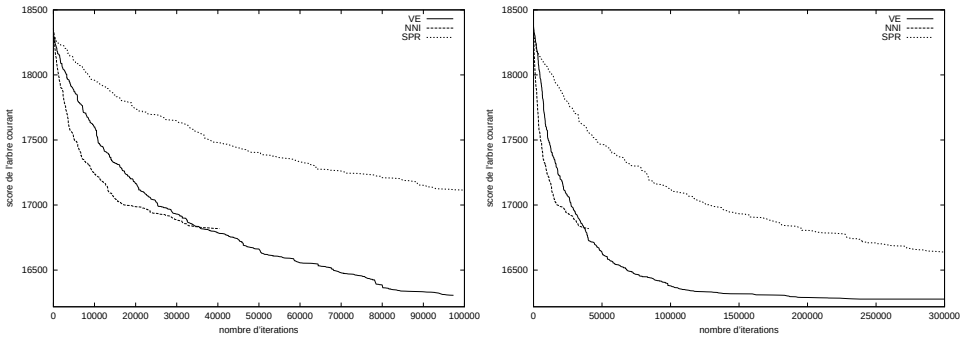

Figure 1. Score de la solution courante sur resp. 10^5 et 3.10^5 itérations (instance *zilla*)

6 Conclusion et perspectives

Pour résoudre le problème MP en reconstruction phylogénétique, les heuristiques de recherche locale SPR et NNI sont parmi les méthodes les plus populaires. Bien qu'elles soient très performantes et rapides pour des petites instances (comportant moins de 100 espèces), elles apparaissent peu fiables lorsqu'on les applique à des instances plus grandes. Dans le cas de voisinages restreints comme NNI, il faut effectuer plusieurs recherches à partir d'arbres distincts (réplications), ou bien détériorer par moment la solution courante pour explorer plusieurs zones de l'espace de recherche et ainsi éviter les pièges des optima locaux. Dans le cas de voisinages larges comme SPR ou TBR, il est nécessaire de les combiner à d'autres méthodes (algorithmes génétiques [15], superarbres [3], ...) car ils ne sont pas suffisamment efficaces utilisés seuls pour de larges instances.

L'objectif ici était de comprendre l'influence du voisinage utilisé en fonction des instances et de l'avancée de la recherche locale, et de proposer une alternative qui combine les propriétés intéressantes des voisinages SPR et NNI. Nous avons alors introduit la notion de voisinage évolutif, et effectué une série d'expérimentations montrant un gain d'efficacité sensible par rapport à SPR et NNI, notamment sur des instances difficiles. De plus, sa robustesse entraîne un gain de temps, car elle permet de minimiser le nombre de réplications. En effet, l'arbre initial influe très peu sur la qualité des solutions retournées par le voisinage évolutif.

Afin d'améliorer encore les performances de notre voisinage évolutif, il est prévu d'étudier plus précisément les propriétés du sous-voisinage améliorant et du sous-voisinage détériorant d'un arbre courant. Le but de cette recherche locale évolutive sera d'utiliser un voisinage qui se construit à partir des propriétés des configurations déjà explorées.

Remerciements: Ce travail est partiellement supporté par Ouest Genopole®. Nous remercions le laboratoire de Pathologie Végétale de l'INRA d'Angers pour leur collaboration, ainsi que Olaf R. P. Bininda-Emonds pour nous avoir fourni l'instance *zilla*.

References

[1] A. A. Andreatta et C. C. Ribeiro. Heuristics for the phylogeny problem. *Journal of Heuristics* 8:429-447, 2002.

[2] B. L. Allen et M. Steel. Subtree Transfer Operations and their Induced Metrics on Evolutionary Trees. *Annals of Combinatorics* 5(1):1-15, 2000.
[3] O. R. P. Bininda-Emonds, Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life. *Computational Biology*, volume 4. Kluwer Academic Publishers, Dordrecht, the Netherlands
[4] L. L. Cavalli-Sforza et A. W. F. Edwards. Phylogenetic analysis: models and estimation procedures. *Evolution* 32: 550-570, 1967.
[5] M. W. Chase *et al.* Phylogenetics of seed plants: an analysis of nucleotide-sequences from the plastid gene rbcL. *Annals of the Missouri Botanical Garden*, 80:528-580, 1993.
[6] A. W. F. Edwards et L. L. Cavalli-Sforza. The reconstruction of evolution. *Annals of Human Genetics* 27: 105-106, 1963.
[7] E. Fargier, A. Goëffon et C. Manceau. L'étude de la diversité génétique au sein de l'espèce *Xanthomonas campestris* révèle une grande homogénéité entre les différents pathovars. *6ème congrès de la Société Française de Phytopathologie*, Toulouse, 2005.
[8] J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution* 17: 368-376, 1981.
[9] W. Fitch. Towards defining course of evolution: minimum change for a specified tree topology. *Systematic Zoology* 20:406-416, 1971.
[10] W. M. Fitch et E. Margoliash. Construction of phylogenetic trees. *Science* 155: 279-284, 1967.
[11] L. R. Foulds et R. L. Graham. The Steiner problem in phylogeny is NP-complete. *Advances in Applied Mathematics* 3:43-49, 1982.
[12] G. Ganapathy, V. Ramachandran et T. Warnow. On contract-and-refine transformations between phylogenetic trees. SODA 2004: 900-909, 2004.
[13] O. Gascuel. On the optimization principle in phylogenetic analysis and the minimum evolution criterion. *Biology and Evolution* 17:401-405, 2000.
[14] P. A. Goloboff. Character optimisation and calculation of tree lengths. *Cladistics* 9: 433-436, 1993.
[15] P. A. Goloboff. Analyzing Large Data Sets in Reasonable Times: Solutions for Composite Optima. *Cladistics* 15:415-428, 1999.
[16] P. Hansen et N. Mladenovic. An introduction to Variable neighborhood search. Metaheuristics, Advances and Trends in Local Search Paradigms for Optimization. Edited by S. Voss et al., 433-458, Kluwer Academic Publishers, Dordrecht, 1999.
[17] D. Hillis, C. Moritz et B. Mable. Molecular Systematics, Sinauer, Boston, 1996.
[18] M. Kimura. A simple model for estimating evolutionary rates of base of base substitutions through comparative studies of nucleotide sequence. *Journal of Molecular Evolution* 16:111-120, 1980.
[19] M. K. Kuhner et J. Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Molecular Biology and Evolution*, 11:459-468, 1994 (Erratum 12:525).
[20] D. R. Maddison. The discovery and importance of multiple islands of most-parsimonious trees. *Syst. Zool.* 43(3):315-328, 1991.
[21] K. C. Nixon. The Parsimony Ratchet, a New Method for Rapid Parsimony Analysis. *Cladistics* 15:407-414, 1999.
[22] C. C. Ribeiro et D. S. Vianna. A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure. *International Transactions in Operational Research* 12:1-14, 2005.
[23] D. F. Robinson. Comparison of labeled trees with valency three. *J. Combin. Theory*, 11:105-119, 1971.
[24] U. W. Roshan, B. M. E. Moret, T. Warnow et T. L. Williams. Rec-I-DCM3: A Fast Algorithmic Technique for Reconstructing Large Phylogenetic Trees. *Proceedings of the IEEE Computational Systems Bioinformatics conference (CSB)*, 2004.
[25] N. Saitou et M. Nei. The neighbor-Joining Method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4:406-425, 1987.
[26] E. Schröder. Vier Kombinatorische Probleme. *Z. Math. Phys.* 15:361-376, 1870.
[27] R. R. Sokal et C. D. Michener. A statistical method for evaluating systemaic relationships *University of Kansas Science Bulletin* 38:1409-1438, 1958.
[28] R. R. Sokal et P. H. A. Sneath. Principles of Numerical Taxonomy. W. H. Freeman, San Francisco, 1963.
[29] D. L. Swofford et G. J. Olsen. in D.M. Hillis and C. Moritz (Ed.) Phylogeny Reconstruction. *Molecular Systematics*, chapter 11:411-501, 1990.
[30] M. S. Waterman et T. F. Smith. On the similarity of dendograms. *Journal of Theoretical Biology* 73:789-800, 1978.