

Synthèse des 2 jours d'étude
sur les problèmes de topologies
Parc national des Cévennes
Application Geotrek-Admin
Novembre 2021



Paris

Nantes

Toulouse

Bruxelles

Makina Corpus

Siège social
11 rue du Marchix
44000 Nantes

Établissement de Toulouse
52 Rue Jacques Babinet
31100 Toulouse

+33 (0)9 70 33 21 50



Table des matières

1	Contexte.....	3
2	Vocabulaire.....	3
3	Rappel des problématiques.....	3
3.1	Problème avec le filtre.....	3
3.2	Problème avec des clés étrangères.....	4
3.3	Problème des géométries cassées.....	4
3.3.1	Suppression d'un tronçon.....	4
3.3.2	Ouverture du réseau de tronçon.....	5
3.3.3	Problème suite à une découpe de tronçon parcouru.....	5
3.3.4	Cas particulier des linéaires qui deviennent des points.....	6
3.4	Problème des géométries modifiées.....	7
4	Conclusion des développements et des tests.....	8
5	Suite de l'étude.....	8

1 Contexte

Le Département des Côtes d'Armor a relevé une anomalie concernant des modifications non souhaitées d'itinéraires (<https://github.com/GeotrekCE/Geotrek-admin/issues/2515>).

Dans le cadre de la consultation et de la commande « 2020-DSI7-MB » portant sur « des prestations d'expertise de paramétrage et développement d'évolutions sur Geotrek-Admin », le Département a fait l'acquisition d'un pack de 40 heures (soit 5 jours) auprès de Makina Corpus.

Suite au retours effectués lors de ces 40 heures, le Parc National des Cévennes a commandé une expertise de 2 jours pour récapituler, cibler et déterminer les chantiers de correction des problèmes. Ce document fait suite au document du retours des 40 heures effectuées :
MakinaCorpus_CD22_Précisions-pack40h_20211022.pdf

2 Vocabulaire

- **PathAggregations** : partie d'une topologie, qui permet de rattacher, par rapport à un tronçon, l'emplacement d'un objet.
- **Topologie** : concerne tous les objets* s'appuyant sur la segmentation dynamique pour se rattacher automatiquement à un tronçon.
Par exemple, les contenus touristiques ne sont pas des topologies car ils sont rattachés via une zone tampon (buffer).

*POIs, Itinéraires, Services, Sentiers, Infrastructures, Signalétiques, Statuts Foncier, Statuts physique, Tronçon de compétence, Tronçon de gestion de travaux, Tronçon de gestion signalétique.

3 Rappel des problématiques

- Topologies linéaires cassées
- Géométries non conformes

Suite à cette étude de 2 jours, nous avons constaté et reproduit les éléments suivants :

3.1 Problème avec le filtre

Le filtre servant à identifier les topologies cassées n'indique pas forcément les géométries invalides.

En effet, il se peut que les topologies soient dites « cassées » sans altérer la géométrie finale. Par exemple si un tronçon B découpe un tronçon A parcouru par un itinéraire, les deux tronçons A et A' porteront le même ordre dans l'agrégation de l'itinéraire. La topologie sera considérée comme

cassée. Or, il s'agit apparemment du comportement historique de Geotrek. Tant que le bon ordre est remonté dans la requête SQL (partie aléatoire du problème), la géométrie n'est pas altérée.

Solution court terme : différencier les deux filtres

- 1 pour les problèmes topologiques (ordre non respecté, trou)
- 1 autre pour les géométrie cassées (linéaires différents de LineString (cas MultiLinestring et Point) → *important pour savoir si un itinéraire publié est cassé*

Estimé entre 0,5 et 1 jour

3.2 Problème avec des clés étrangères

Un mécanisme assez ancien de Geotrek prévoyait de supprimer la clé étrangère des agrégations, créée automatiquement avec Django par une clé étrangère de type différent.

Depuis la version [2.32.7 \(2020-03-02\)](#), cette fonction ne supprimait plus rien mais recréait une nouvelle clé étrangère à chaque lancement d'une migration.

En cours de correction ici : <https://github.com/GeotrekCE/Geotrek-admin/pull/2820>

Il se peut que des utilisateurs de Geotrek aient des comportements différents s'ils utilisent des versions de Geotrek avant ou après cette version.

Corrigé dans les 2 jours de l'étude

3.3 Problème des géométries cassées

Les géométries des topologies linéaires peuvent se casser selon certains cas de figure :

3.3.1 Suppression d'un tronçon

La suppression d'un tronçon parcouru par une ou plusieurs topologies linéaires alerte l'utilisateur sur les risques encourus, mais lui laisse tout de même la possibilité de casser les données.

Suite à l'étude, il faut soit :

- interdire catégoriquement la suppression d'un tronçon parcouru par un linéaire. Le risque qu'une personne ne faisant pas attention ou n'étant pas au fait des conséquences de cette action existera toujours .

Estimé entre 1 et 5 jours (pas de trigger impacté, il suffit de mettre un `ON DELETE PROTECT` sur le `path_id` d'un `core_pathaggregation`)

- alerter l'utilisateur qu'en cas de suppression, les topologies linéaires seront aussi supprimées. C'est la méthode dure, mais elle a le mérite de maintenir l'intégrité des données.

Estimé entre 2 et 7 jours . Il faudra personnaliser le message d'erreur SQL car si la modification a lieu coté QGIS, il ne sera pas possible de CASCADER la suppression des linéaires (Coté Django, il sera possible de tout supprimer).

3.3.2 Ouverture du réseau de tronçon

L'ouverture du réseau de tronçons par le déplacement d'une extrémité du tronçon parcouru par un ou des topologies linéaires causera automatiquement des problèmes sur les géométries des linéaires concernés.

Suite à l'étude, il faut soit :

- interdire l'ouverture du réseau
Difficilement estimable en l'état, il faut écrire un trigger qui vérifiera si une jonction existe avant d'autoriser la suppression)

Cela va impliquer des problèmes comme par exemple dans le cas où un utilisateur souhaite déplacer ou modifier plusieurs tronçons à la suite. Mais le problème est bien réel et existe actuellement : même si on les reconnecte après coup, le fait de déconnecter, même temporairement, l'extrémité d'un tronçon connecté et parcouru par un linéaire provoque des anomalies de géométries irrécupérables.

- Une solution acceptable, mais potentiellement difficile à mettre en place, serait de déplacer le nœud complet (l'extrémité de tous les tronçons connectés à la fois)
Difficilement estimable en l'état, il faut modifier le JavaScript pour permettre cela, et vérifier que l'enregistrement des plusieurs tronçons à la suite, et la cascade de triggers qui en découle ne brisera pas les topologies. C'est une problématique auquel répond les topologies PostGIS. Il se peut, par contre, qu'aucune solution ne soit possible en gérant les modifications de tronçons les unes après les autres et pas « en même temps » (exemple : je souhaite déplacer un nœud, chaque tronçon relié au nœud sera mis à jour l'un après l'autre, le réseau sera donc ouvert un laps de temps, suffisant pour casser les topologies.
(il faudrait tester si l'utilisation de FOR EACH STATEMENT au lieu de FOR EACH ROW répond au problème)

3.3.3 Problème suite à une découpe de tronçon parcouru

Lors de la découpe d'un tronçon par un autre, les agrégations (ordre de passage) sont mises à jour (découpe et ajout), mais l'ordre de passage n'est pas mis à jour (dans certains cas, l'ordre de passage est dupliqué, dans d'autres il est nul).

La « magie » SQL fait que la plupart du temps les lignes remontent dans un ordre qui permet de ne pas casser la géométrie finale. Mais aucune indication dans la requête SQL à effectuer ne permet de fixer cet ordre (ORDER BY).

Il faudrait soit :

- réussir à fixer ces ordres pour éviter ce problème aléatoire

Difficilement estimable car il s'agit de modifier un trigger qui déclenche des modifications en cascade.

- soit utiliser une autre colonne, par exemple un `create_datetime` ou `update_datetime`. (colonnes non existantes pour le moment). La réalisation ne semble pas compliquée à réaliser.

Estimé entre 1 et 2 jours, mais il convient avant cela d'étudier l'impact sur l'existant (partie difficilement estimable).

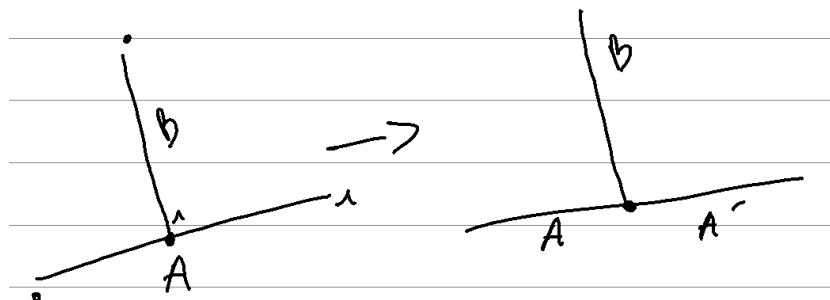
3.3.4 Cas particulier des linéaires qui deviennent des points

Il existe un cas particulier où après une découpe d'un de ses tronçons par un nouveau, une topologie linéaire devient un point. Cela se produit à cause d'un [trigger SQL](#) qui regarde si un point de passage est situé à une extrémité de tronçon (0 ou 1). Dans ce cas, toutes les parties linéaires sont supprimées et seul le ou les points de passage sont gardés. La géométrie finale devient donc un point / multipoint.

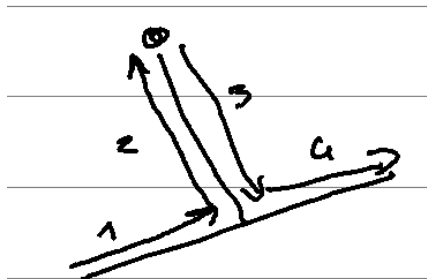
Il faudra fixer ce problème, mais avant cela, il convient de déterminer pourquoi ce comportement particulier a été mis en place, et si la correction n'impliquera pas un retour en arrière sur une autre problématique. (cela fait penser au cas des points de passage, et au cas des ponctuels associés à plusieurs tronçons à une intersection de type pâte d'oie ou carrefour.)

Reproduction :

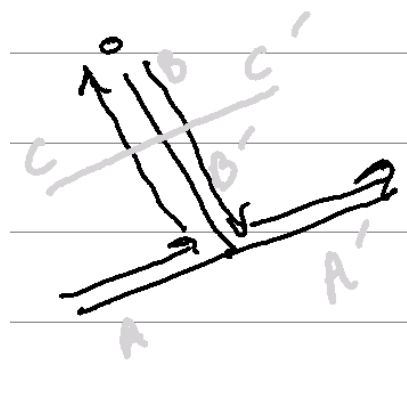
on trace deux tronçons qui se découpent :



On dessine l'itinéraire en mettant un point de passage à l'extrémité du tronçon B



On découpe le tronçon B avec un nouveau tronçon C



L'itinéraire est maintenant un point

Difficilement estimable, nécessite une étude d'impact poussée.

3.4 Problème des géométries modifiées

Actuellement, l'affichage de la géométrie à modifier peut ne pas correspondre à ce qui a été enregistré précédemment. Si un plus court chemin existe, la bibliothèque d'affichage de la carte va corriger le tracé et prendre le plus court chemin.

Plusieurs pistes :

- Insérer automatiquement des points de passage sur chaque tronçon parcouru (solution évoquée précédemment)
 - sur les longs itinéraires, le nombre de point de passage enlèvera de la flexibilité de modification (plusieurs dizaines ou centaines ajoutés automatiquement)
- Utiliser des points de passage spécifique à l'affichage

- problématique du code à développer côté JS et Leaflet, et s'assurer que les développements fonctionneront aussi sous Leaflet 1.x. Il se peut aussi que ce ne soit pas possible avec la librairie Leaflet.Snap utilisée.
- Lors d'une modification, afficher la géométrie originale pour que l'utilisateur soit averti d'un éventuel changement à la sauvegarde (prévention modification non souhaitée)
- Insérer automatiquement des points de passage obligatoires à l'extrémité du tronçon découpé emprunté par le linéaire (idem solution 1 mais avec moins de points de passage ajoutés, moins contraignant pour l'utilisateur, mais plus compliqué à mettre en place)

4 Conclusion des développements et des tests

Nous avons remonté plusieurs cas de figure qui peuvent provoquer des problèmes topologiques ou géométriques. Certains peuvent être corrigés rapidement, d'autres ont un impact si grand qu'il est difficile de quantifier le travail à effectuer.

Un cas problématique est remonté :

le fait de modifier plusieurs tronçons à la suite en les déconnectant / reconnectant n'a jamais fonctionné comme il faut. La solution la plus simple serait de bloquer cette possibilité, mais il faudrait trouver une explication ou une parade pour que l'utilisateur comprenne pourquoi il n'a pas le droit de faire cela.

Une solution qui permettrait de gérer ce cas de figure serait de **gérer les topologies avec PostGIS** mais :

- cela nécessite beaucoup de travail (éventuellement une v3)
- Django ne peut pas utiliser les topologies PostGIS nativement. On devra l'implémenter pour le projet. Cependant, l'utilisation des topologies revient régulièrement dans bon nombre de projet SIG, et pourrait être l'objet du développement d'une bibliothèque Django spécifique mais réutilisable.
(beaucoup de travail certes, mais potentiellement intéressant pour la communauté des développeurs Django dont certains pourraient s'intéresser ou contribuer au projet)

5 Suite de l'étude

Définition des chantiers à mettre en œuvre parmi les propositions suscitées, leur priorisation et leur organisation dans des sprints de développements.

La liste ci-dessous est triée de la plus petite à la plus grande difficulté estimée.

- Blocage de la suppression d'un tronçon parcouru par un linéaire
- Affichage de la géométrie originale lors d'une modification d'un linéaire topologique



**MAKINA
CORPUS**

- Alertes déconnexion réseau topologique
- Correction du cas du linéaire qui devient point après découpe par un nouveau tronçon
- Insertion de point de passage obligatoire (invisible / sur chaque nouvelle intersection / sur chaque tronçon parcouru selon la faisabilité)
- Correction des ordres des agrégations