Institut Géographique National
Laboratoire COGIT

Appariement dans GeOxygene
version 1.0

Date de la dernière modification	12 décembre 2007
Rédacteurs	Sébastien Mustière – IGN / Laboratoire COGIT
Diffusion	Libre
Contenu du document	Description rapide des outils d'appariement dans GeOxygene : appariement de réseaux et appariement de surfaces. Quelques repères pour paramétrer le processus.

TABLE DES MATIERES

1. Portée du document

Ce document présente les outils d'appariement : quels outils existent, comment les utiliser, comment les paramétrer.
Ce document n'aborde **pas** l'import des données avant leur appariement, ni le stockage persistant des résultats d'appariement, ni leur visualisation. Pour cela, se référer à la documentation générale.

2. Outils d'appariement disponibles et références

Différentes outils d'appariement sont disponibles :

- Un processus d'appariement de réseau, particulièrement conçu pour gérer le cas de réseaux avec des niveaux de détails différents.
    - Source principale pour toute référence à ce processus : [Mustière et Devogele 2008]
    - Autres références sur ce processus : [Devogele 1997],[Devogele, Trévisan et Raynal 1998], [Mustière 2006].

- Des outils d'appariement de surfaces.
    - Source principale pour toute référence à ce processus : [Bel Hadj Ali 2001]

A ces outils au cœur de l'appariement, s'ajoutent des outils pour le stockage, la gestion, la comparaison et la visualisation des liens d'appariement.

3. Appariement de réseaux [Mustière et Devogele 2008]

Package concerné : fr.ign.cogit.geoxygene.contrib.appariement.reseaux

Le but de l'algorithme est d'apparier deux réseaux, en particulier si ils ont des niveaux de détail différents. Il prend en entrée deux ensembles de populations d'objets formant un réseau (des lignes et éventuellement des points), et fournit en sortie un ensemble de liens d'appariement entre les objets.

3.1. Lancement de l'algorithme

Le lancement de l'algorithme se fait à partir de la méthode AppariementDeJeuxGeo, de la classe AppariementIO..

Tous les paramètres de l'algorithme (des données à traiter, aux seuils de distance en passant par les options de prétraitement) doivent être définis avant de lancer cette méthode (création d'un objet de la classe ParametresApp).

La méthode renvoie un ensemble de liens de la classe EnsembleDeLiens du package appariement.

3.2. Description de l'algorithme

La description du principe général de l'algorithme ci-dessous est issue de [Mustière 2006]. Pour une description plus fine, voir [Mustière et Devogele 2008]

3.2.1. Prétraitement

If the networks have very different structures, it is hard to directly compare them. We thus perform a pre-treatment of them, in order to give them a similar structure and to prepare the matching. Typical rep-treatment are mentioned below.

- *For both networks, our algorithm requires arcs and nodes, and topological relationships between them. But some of the networks only contain arcs and no explicit nodes. In this case, a first required pre-treatment is the creation of the graph structure (Figure 1). Different strategies are possible. The simplest one is to just create nodes at extremities of arcs and to compute topological relationships. In some cases, it may also be useful to fusion distinct but very close nodes coming from different arcs, in order to overcome topological problems in the data. Sometimes, it is also useful to build a planar graph from the arcs.*

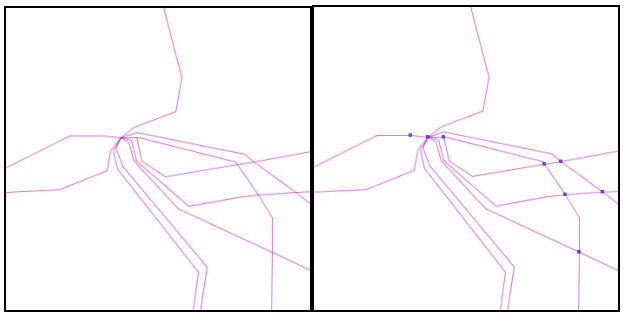

Figure 1. Nodes are added at the extremities of electric lines.

- *Another typical required pre-treatment is the transformation of connections. For example, modelling of connections of electric networks may be very different in some cases. In the example of Figure 2, we first transform the connections of the electric network to make them similar to classical graph structure: a node is created in the centre of each transforming station, and all lines entering in the station are connected to this node. Matching will after be done with this transformed network.*

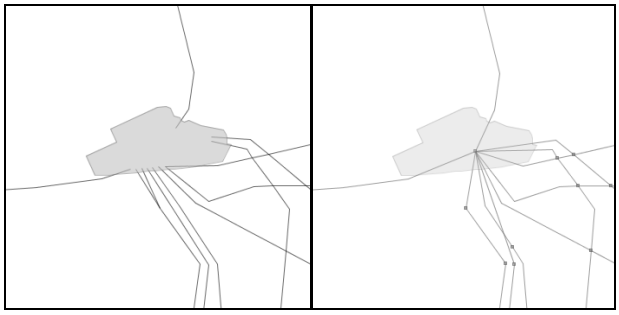

Figure 2. Electric lines arriving in the transforming station are connected before matching.

- *In some cases, there also may exist important differences in the exact location of the extremities of the network. Classically, this appends for hydrographical and railway networks. In these cases we split arcs and add extra nodes to networks, by projecting extremities of one network on the other one (Figure 3).*

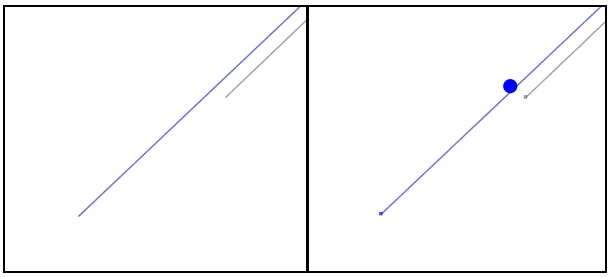

Figure 3. Networks may not stop at the same place: extra nodes are added before matching.

3.2.2. Appariement

Once networks have been prepared, they can be matched. We briefly describe the main steps of the algorithm.

- *The first step is a pre-matching of nodes. For each node of the less detailed database (Net1), we look for close nodes in the other database Net2 candidate for matching.*

- *The second step is a pre-matching of arcs. For each arc of Net1, we look for close arcs in Net2, candidate for matching (Figure 4). This pre-matching is based on the Hausdorf distance between lines.*

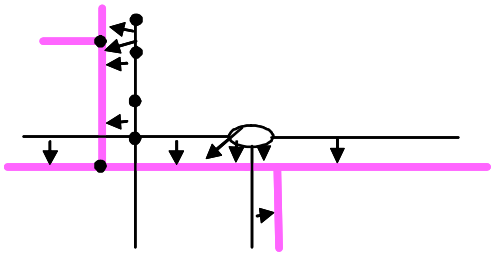

Figure 4. Pre-matching of arcs.

- *The third and most complex step is the actual matching of nodes. It is based on the combined analysis of the results of the two pre-matching steps.*

    - *We first look for 1-1 matching, i.e. one node of Net1 corresponding to one node of Net2. For each pair of nodes candidates for matching, we look if their respective connected arcs are also candidate for matching. When nodes and arcs matching are fully consistent, they are matched. For example, in Figure 5 the node Nc is matched to the node Nt because: 1/ these two nodes are candidate for matching, and 2/ all the 3 arcs connected to Nc are candidate for matching with arcs connected to Nt. This is the only case like that in the example.*

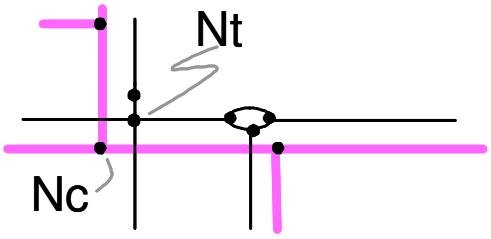

Figure 5. 1-1 node matching.

- *We then look for 1-n matching, i.e. one node of Net1 corresponding to several nodes and arcs of Net2. Without detailing this step, let's say that its principle is to group nodes and lines of Net2. These groups are then considered as hyper-nodes in the network and managed like the nodes: we look for groups where pre-matching of nodes and arcs are consistent (Figure 6).*

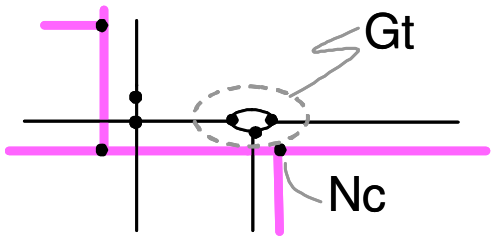

Figure 6. A 1-n node matching.

- *Finally, the last step concern line matching. Once nodes of have been matched, we consider arcs of Net1 one by one, and each arc is matched to a set of arcs of Net2. Roughly speaking, this arc of Net1 is matched to the shortest path in Net2 linking the nodes matched to the extremities of the arc of net1 (Figure 7).*

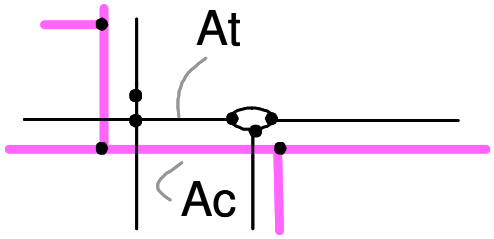

Figure 7 . Two arcs matched

3.3. Paramétrage

Avertissement
Le processus a de très nombreux paramètres (environ 40). Ceci s'explique par la diversité des prétraitements possibles, et par le fait que le processus est un outil de recherche, non encore complètement finalisé pour une mise en production. Cependant, ce nombre important de paramètre ne doit pas paraître un frein trop important à l'utilisation du processus. Tout d'abord, la plupart de ces paramètres sont inutiles en première approche, et servent à affiner précisément le processus in fine pour traiter les cas particuliers : **on peut ainsi considérer que seuls 2 paramètres sont à fixer impérativement : la distance maximale attendue entre les objets, et la précision moyenne de la moins précise des bases**. De plus, les paramètres sont organisés en différents groupes pour simplifier leur détermination. Enfin, tous les paramètres sont décrits en détail dans l'API, et ont une valeur par défaut.

Nous donnons ici quelques grandes lignes pour guider le paramétrage.

3.3.1. Généralités sur les paramètres

De manière générale, le réseau le moins détaillé est appelé réseau1, et le plus détaillé est appelé réseau 2.

On distingue parmi les paramètres,

- les paramètres essentiels qui spécifient quelles données traiter et les seuils de distance à utiliser,
- les paramètres secondaires qui permettent de raffiner le processus et la forme des résultats et qui peuvent en première approximation être laissés à leurs valeurs par défaut.

3.3.2. Les paramètres essentiels

Ces paramètres essentiels sont de deux types :

- Les paramètres qui spécifient quelles données traiter.
  - Leur nom commence par *population.*
  - Chaque réseau à traiter peut être constitué à partir d'un ou plusieurs ensembles d'objets avec une géométrie linéaire (les futurs arcs du réseau, par exemple une classe de tronçons de route et une classe de tronçons de chemin si on veut les traiter ensemble), ainsi qu'un ou plusieurs ensembles d'objets ponctuels (les futurs nœuds du réseau, comme une classe de carrefours ou de gares). Si il n'y a pas d'objet ponctuel, des noeuds seront créés automatiquement pour le calcul, mais aucun résultat ne sera ressorti sur ces nœuds.
  - Un paramètre particulier, *populationsArcsAvecOrientationDouble,* spécifie si les réseaux sont orientés (ex : hydrologie) ou non (ex : lignes électriques)

- Les paramètres qui fixent les seuils de distance pour la recherche des objets à apparier (écarts maximaux supposés entre les objets homologues).
  - Leur nom commence par *distance*.
  - Les distances maximales entre nœuds et arcs (*distanceNoeudsMax* et *distanceArcsMax*), doivent de l'ordre des écarts maximaux espérés entre les réseaux (typiquement, ce peut être égal à deux fois la somme des précisions moyenne des bases). Il est conseillé en première approximation de donner la même valeur à ces deux paramètres. En cas d'hésitation, il est conseillé de mettre des paramètres plutôt trop pessimistes (trop forts) plutôt que trop optimistes.
  - Un autre paramètre, *distanceArcsMin*, est plus difficile à appréhender. Il peut être fixé en première approximation à la précision moyenne de la moins précise des bases.
  - Un dernier paramètre, *distanceNoeudsImpassesMax*, permet de raffiner le traitement des impasses et ne doit pas être utilisé en première approximation.

3.3.3. Les paramètres secondaires

Ces paramètres secondaires précisent les prétraitements à réaliser, la forme des exports, et permet de lancer diverses variantes du processus. On distingue :

- Ceux qui spécifient les pré-traitements topologiques.
  - Leur nom commence par *topologie*.
  - Ces paramètres permettent de rendre le graphe planaire, de corriger les erreurs de topologie aux connections, etc.

- IMPORTANT : L'appariement requiert des réseaux avec une topologie propre, ce qui explique l'intérêt de ces paramètres. Mais autant que possible, nous conseillons de faire ce nettoyage topologique a priori avant le traitement (dans un SIG classique du marché par exemple, qui sera plus optimisé pour faire cela que GeOxygene), et ensuite de ne pas utiliser ces paramètres.

- Ceux qui spécifient les pré-traitements « de projection ».
  - Leur nom commence par *projete*.
  - Dans certains cas, un redécoupage des réseaux en projetant les nœuds d'un réseau sur l'autre réseau peuvent être utiles (voir exemple Figure 3), en particulier si les données ont des niveaux de détail proche. Ce sont ces paramètres qui guident cela. Ils vont néanmoins un peu à l'encontre de la philosophie générale du processus la plupart du temps. Nous conseillons donc de ne pas les utiliser en première approximation.

- Ceux qui fixent diverses variantes du processus, créées au fur et à mesure de besoins ponctuels.
  - Leur nom commence par *variante*.

- Ceux qui permettent de débuguer les traitements.
  - Leur nom commence par *debug*.
  - Ils permettent essentiellement de fixer l'aspect graphique des liens d'appariement trouvés, ainsi que l'affichage du texte pendant le calcul.

3.4. Autour de l'appariement de réseaux

D'autres outils sont disponibles autour de l'appariement de réseaux, certains pour évaluer les écarts entre deux réseaux (classe Comparaison), d'autres pour recaler un réseau sur un autre après l'appariement (classe Recalage). Cette bibliothèque d'outils est embryonnaire et demande à être testée et étendue : il faut utiliser ces outils avec précaution car ils n'ont pas été testés intensément et ils ont été conçus pour des cas relativement simples.

4. Appariement de surfaces [Bel Hadj Ali 2001]

Package concerné : fr.ign.cogit.geoxygene.contrib.appariement.surfaces

4.1. Lancement du processus

Le processus se lance par la méthode *appariementSurfaces* de la classe AppariementSurfaces, après avoir fixé les paramètres en créant un objet *ParametresAppSurfaces*.

4.2. Principe général

Le processus d'appariement de surfaces a été conçu dans une optique d'évaluation de la "qualité de données" où une des deux bases à apparier est considérée comme une référence, de meilleure qualité que l'autre. L'appariement est utilisé comme étape préliminaire à la comparaison des objets dans [Bel Hadj Ali 2001]. Les liens d'appariement sont qualifiés plus finement par la suite (e.g. avec des mesures de forme des objets), mais ceci n'est pas disponible dans le code présent.

Ces outils ont été mis au point pour apparier principalement des surfaces isolées (e.g. "bâtiments") plus que des partitions ou des surfaces connectées (e.g. "occupation du sol"), même si des tests relativement fructueux ont été effectués sur ce type d'objets.

Les principes généraux de l'approche sont les suivants:

- Apparier des surfaces en comparant les surfaces elles-mêmes plus que les lignes contours.
- Le choix des mesures utilisées s'appuie sur des principes probabilistes.
- L'optique générale est 1/ apparier avec des mesures simples (intersections de surfaces) puis 2/ qualifier ensuite avec des mesures plus fines (et éventuellement rejeter alors l'appariement), mais cette deuxième étape n'est pas disponible pour l'instant. Ceci explique la simplicité des mesures utilisées lors de l'appariement présenté ici.
- Ne tient compte que de la géométrie; les éventuelles relations de connexion entre surfaces sont ignorées.

4.3. Description du processus

1- Pré-appariement: Les surfaces des BD1 et BD2 sont associées en faisant des comparaisons 1-1 entre les objets des deux jeux de données. Deux objets sont associés si ils respectent la *mesure d'association* (voir définition des mesures ci-dessous).

2- Regroupement: des liens entre groupes de bâtiments (liens n-m) sont créés à partir des liens d'association (i.e. on identifie les parties connexes du graphe reliant les entités grâce au lien d'association)

3- Affinement des liens n-m. Pour 2 groupes G1 et G2 de bâtiments reliés, on recherche les meilleurs sous-groupes SG1 et SG2 de G1 et G2 maximisant soit la somme *Exactitude(SG1,SG2)* + *Complétude(SG1, SG2)*, soit minimisant *Distance surfacique (SG1, SG2) (*cette dernière option est conseillée par l'auteur en cas de bases avec des résolutions similaires). Des heuristiques sont proposées par l'auteur pour ne pas tester toutes les configurations possibles.

4- Vérification des liens. Ces appariements entre sous-groupes ne sont acceptés que si l'exactitude et la complétude dépassent un certain seuil (ou la distance surfacique). Ce seuil est fixé empiriquement par analyse des données dans les tests réalisés, de l'ordre de 0,5 pour l'exactitude et la complétude, ou 0,6 pour la distance surfacique.

Mesures utilisées

Soient A et B deux surfaces (chacune connexe ou non) :

- Association(A,B) = Vrai si
  Surface(A ∩ B) > min(R1,R2) ; avec R1 et R2 les résolutions des BDs contenant A et B
  et Surface(A ∩ B) > Surface(A) x 0,2
  et Surface(A ∩ B) > Surface(B) x 0,2
  Faux sinon
- Distance surfacique = (Surface(A∪B) – Surface(A ∩ B)) / Surface(A ∪ B)
- Exactitude(A,B) = Surface(A ∩ B) / Surface(A)
- Complétude(A,B) = Surface(A ∩ B) / Surface(B)

5. Références

Mustière S., Devogele T., 2008, Matching Networks with Different Levels of Detail. GeoInformatica. (accepté, à paraître en 2008).

Devogele T. 1997. Processus d'intégration et d'appariement de bases de données Géographiques. Application à une base de données routières multi-échelles, PhD Thesis, University of Versailles.

Devogele T., Trévisan J.,, Raynal L. 1998. Building a multi-scale database with scale-transition relationships. Advances in GIS research II, Proceedings of 7th International Symposium on Spatial Data Handling, pp.337-351.

Mustière S. 2006. Results of experiments on automated matching of networks. Proceedings of the ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data, Hanover, February 2006, pp.92-100