



Graphab 3.0

Manuel d'utilisation

Marc Bourgeois, Céline Clauzel, Jean-Christophe Foltête, Xavier Girardet,
Paul Savary, Gilles Vuidel

02/12/2024

Contents

1	Introduction	5
1.1	A propos de Graphab	5
1.1.1	Auteurs	5
1.1.2	Conditions d'utilisation	5
1.2	Configuration requise	5
1.3	Installation et lancement du programme	6
2	Interface Graphique	7
2.1	Démarrer un projet Graphab	7
2.1.1	Identification du projet	7
2.1.2	Définition d'un habitat	8
2.1.3	Création d'un jeu de liens	9
2.2	Habitat	10
2.2.1	Création d'un habitat raster	10
2.2.2	Création d'un habitat vecteur	11
2.2.3	Capacité des taches	11
2.2.4	Méta-taches	13
2.2.5	Fonctions complémentaires (menu contextuel)	14
2.3	Jeu de liens	15
2.3.1	Création d'un jeu de liens	15
2.3.2	Conversion des distances	17
2.3.3	Fonctions complémentaires (menu contextuel)	18
2.4	Graphes	18
2.4.1	Création d'un graphe	18
2.4.2	Fusion de plusieurs graphes	19
2.4.3	Corridors	20
2.4.4	Partitionnement de graphe	22
2.4.5	Visualisation des graphes	24
2.4.6	Fonctions complémentaires (menu contextuel)	25
2.5	Métriques de connectivité	26
2.5.1	Famille de métriques et niveaux de calcul	26
2.5.2	Paramétrage des métriques	26
2.5.3	Paramétrage des métriques pondérées	27
2.5.4	Paramétrage multi-habitat	28
2.5.5	Affichage des résultats	28
2.5.6	Calcul des métriques par lot	29
2.6	Analyses	32
2.6.1	Interpolation de métrique	32
2.6.2	Ajout de taches	32
2.6.3	Modification de la carte de paysage	34
2.7	Jeu de données externe	35

2.7.1	Import de données externes	35
2.7.2	Générer des points aléatoires	36
2.7.3	Fonctions complémentaires (menu contextuel)	36
2.8	Affichage	37
2.8.1	Menu contextuel	37
2.8.2	Propriétés des éléments sur la carte	38
3	Interface en ligne de commande	39
3.1	Lancer Graphab en ligne de commande	39
3.2	Syntaxe	39
3.2.1	Définition	39
3.2.2	Séparateur	39
3.2.3	Paramètre optionnel	39
3.2.4	Intervalle et liste de valeurs	39
3.2.5	Séquençage des commandes	40
4	Références des commandes	41
4.1	Commandes générales	41
4.1.1	-help : affichage de l'aide	41
4.1.2	-metrics : affichage des métriques	42
4.2	Gestion du projet	43
4.2.1	-create : création d'un projet	43
4.2.2	-project : chargement d'un projet	43
4.2.3	-show : affichage des éléments d'un projet	43
4.2.4	-dem : chargement d'un MNT	44
4.3	Habitat	44
4.3.1	-habitat : création d'un habitat à partir de la carte de paysage	44
4.3.2	-vhabitat : création d'un habitat à partir d'une couche vecteur	45
4.3.3	-usehabitat : sélection des habitats	45
4.3.4	-removehabitat : suppression d'habitat	45
4.3.5	-mergehabitat : fusion d'habitat	46
4.3.6	-capa : capacité des taches	46
4.3.7	-metapatch : création d'un habitat méta-tache	47
4.4	Jeu de liens	48
4.4.1	-linkset : création d'un jeu de liens	48
4.4.2	-uselinkset : sélection des jeux de liens	49
4.4.3	-removelinkset : suppression de jeux de liens	49
4.5	Graphe	49
4.5.1	-graph : création d'un graphe	49
4.5.2	-usegraph : sélection des graphes	50
4.5.3	-removegraph : suppression de graphes	50
4.5.4	-mergegraph : fusion de graphes	50
4.5.5	-corridor : calcul de corridors	51
4.5.6	-cluster : partitionnement de graphe	51
4.6	Métrique	52
4.6.1	-gmetric : calcul d'une métrique globale	52
4.6.2	-cmetric : calcul d'une métrique par composante	54
4.6.3	-lmetric : calcul d'une métrique locale	54
4.6.4	Paramétrage des métriques	55
4.6.5	-interp : interpolation de métriques	56
4.7	Jeu de données et modèle	57
4.7.1	-dataset : import de jeu de données	57
4.7.2	-distance_matrix : matrice de distances d'un jeu de données ou habitat	57
4.7.3	-model : calcul d'un modèle	58

4.8	Ajout/Suppression d'éléments	59
4.8.1	-delta : suppression d'un élément	59
4.8.2	-gremove : suppression de plusieurs éléments	59
4.8.3	-gtest : suppression et ajout itératif d'éléments	60
4.8.4	-addpatch : ajout itératif de taches	62
4.8.5	-remelem : suppression itérative d'éléments	63
4.9	Changements d'occupation du sol	64
4.9.1	-landmod : changements d'occupation du sol	64
4.9.2	-landmodgm : changements d'occupation du sol sur métrique globale	65
4.10	Options	66
4.10.1	-nosave	66
4.10.2	-distconv	66
4.10.3	-proc	66
4.10.4	-mpi	66
5	Exemples de lignes de commandes	67
5.1	Création d'un projet jusqu'au graphe	67
5.2	Habitat	67
5.3	Jeux de liens et graphes	68
5.3.1	Jeu de liens	68
5.3.2	Graphes	68
5.4	Métriques	68
5.4.1	Métrique globale	68
5.5	Multi-habitat	69
5.6	Afficher le projet	69
6	Capacités de traitement et limitations	71
6.1	Parallélisme	71
6.1.1	Un ordinateur : threads	71
6.1.2	Cluster : MPI	72
6.2	Gestion mémoire	72
7	Métriques	73
7.1	Métriques pondérées	75
7.1.1	Flux	75
7.1.2	Connectivité équivalente	76
7.1.3	Probabilité de Connectivité	76
7.1.4	Flux d'interaction	77
7.1.5	Fractions de delta Probabilité de Connectivité	78
7.1.6	Indice de centralité intermédiaire	79
7.1.7	Indice Intégral de Connectivité	79
7.1.8	Flux circuit	80
7.2	Métriques de surface	80
7.2.1	Taille moyenne des composantes	80
7.2.2	Taille de la plus grande composante	80
7.2.3	Probabilité de coïncidence de classe	81
7.2.4	Taille de composante attendue	81
7.3	Métriques topologiques	81
7.3.1	Lambda de Wilks	81
7.3.2	Indice d'Harary	82
7.3.3	Diamètre	82
7.3.4	Nombre de composantes	82
7.3.5	Degré du noeud	83
7.3.6	Coefficient de groupement	83
7.3.7	Centralité de proximité	83

7.3.8	Excentricité	84
7.3.9	Corrélation de connectivité	84

Chapter 1

Introduction

1.1 A propos de Graphab

Le programme Graphab est un outil de modélisation des réseaux écologiques fondé sur les graphes paysagers. C'est un outil intégré composé de 4 modules :

- création de graphes à partir d'une carte de paysage avec identification des taches d'habitat et des liens (distance euclidienne ou chemin de moindre coût)
- calcul de métriques de connectivité
- liens avec des données externes
- interface de géo-visualisation

Un plugin QGIS Graphab4QGIS permet d'utiliser Graphab directement depuis QGIS : <https://plugins.qgis.org/plugins/graphab4qgis/>

1.1.1 Auteurs

Le programme Graphab a été développé par Gilles Vuidel et Jean-Christophe Foltête au laboratoire [ThéMA \(Université de Franche-Comté – CNRS\)](#) avec les contributions de Anissa Bellil, Marc Bourgeois, Céline Clauzel, Stéphane Garnier, Xavier Girardet, François-Marie Martin, Yohan Sahraoui et Paul Savary. Le développement initial du logiciel a été financé par le ministère de l'écologie, du développement durable, des transports et du logement dans le cadre du programme [ITTECOP](#). Le logo de Graphab a été conçu par [Gachwell](#).

1.1.2 Conditions d'utilisation

Le programme Graphab est disponible librement sous licence GPL. Les utilisateurs de Graphab sont invités à citer une des références suivantes [[Foltête et al.\(2012a\)](#), [Foltête et al.\(2021\)](#)] dans leurs travaux :

Foltête J.-C., Vuidel G., Savary P., Clauzel C., Sahraoui Y., Girardet X., Bourgeois M. 2021. Graphab: an application for modeling and managing ecological habitat networks. *Software Impacts*.8: 100065.

Foltête J.C., Clauzel C., Vuidel G., 2012. A software tool dedicated to the modelling of landscape networks. *Environmental Modelling & Software*, 38: 316-327.

1.2 Configuration requise

Graphab fonctionne sur tout ordinateur supportant Java 11 ou supérieur (PC sous Linux, Windows, Mac...). Toutefois, lorsqu'il s'agit de données très volumineuses, la quantité de mémoire vive (RAM) de l'ordinateur

peut limiter le nombre maximum de nœuds et de liens qui peuvent être traités en une seule fois avec Graphab. De plus, pour certaines métriques complexes, la puissance de votre processeur va déterminer la vitesse de leur calcul. Vous trouverez plus de détails à ce sujet dans la section 6 et dans [Foltête et al.(2012a)].

1.3 Installation et lancement du programme

Graphab est téléchargeable à cette adresse : <https://sourcesup.renater.fr/www/graphab>.

- Télécharger et installer Java 11 ou + (adoptium.net). Installer de préférence la version 64 bits de Java.
- Télécharger graphab-3.0.jar
- Lancer graphab-3.0.jar

Attention, les projets ne sont pas compatibles entre les versions 2.x et 3.x de Graphab !

Graphab peut être utilisé par deux interfaces : une interface graphique et une interface en ligne de commande. L'interface graphique permet de créer un projet, des graphes, visualiser les données, calculer des métriques, etc. L'interface en ligne de commande permet quant à elle, de lancer ces même calculs sur des ordinateurs distants sans interface graphique, ou bien lancer automatiquement plusieurs traitements à la suite.

Le chapitre 2 décrit l'ensemble des fonctionnalités du logiciel disponibles à partir de l'interface graphique. Les chapitres 3, 4, 5 détaillent comment utiliser Graphab en ligne de commande. L'ensemble des métriques disponibles sont détaillées dans le chapitre 7. Et le chapitre 6 revient sur les modes de parallélisation, les performances et la gestion mémoire.

Chapter 2

Interface Graphique

Une fois Graphab 3.0 lancé, le menu fichier donne accès à 4 rubriques :

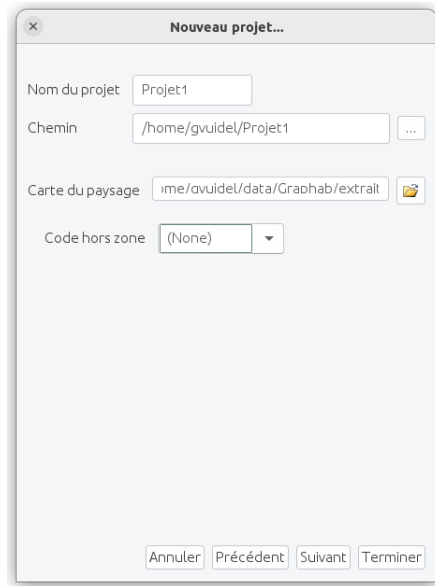
- Fichier / Nouveau projet : pour commencer la création d'un projet dans lequel toutes les données et résultats sont automatiquement enregistrés.
- Fichier / Ouvrir un projet : ouvre un projet déjà créé auparavant.
- Fichier / Préférences : pour changer certains paramètres du programme : langue anglais/français ; quantité maximale de mémoire utilisable ; nombre de processeurs à utiliser. Il est recommandé d'adapter la quantité de mémoire et le nombre de processeurs aux capacités de l'ordinateur utilisé et aux traitements prévus (voir à ce sujet la section [Capacités de traitement et limitations](#)).
- Fichier / Journal : affiche le journal des évènements.

2.1 Démarrer un projet Graphab

La mise en place d'un nouveau projet Graphab s'effectue à partir du menu : **Fichier / Nouveau projet**. L'utilisateur doit renseigner plusieurs fenêtres successives : identification du projet, import de la carte paysagère, création d'un habitat, création d'un jeu de liens. Un projet correspond à l'exploitation d'une seule carte paysagère initiale, mais il peut comporter plusieurs habitats, jeux de liens et graphes. Au terme de la phase de démarrage, le projet sera le support pour la création de multiples graphes et calculs de métriques de connectivité.

2.1.1 Identification du projet

Dans la première fenêtre, l'utilisateur doit nommer le projet et indiquer dans quel dossier celui-ci sera créé.



Le second élément concerne l'importation de la carte du paysage. Celle-ci doit être un fichier de type raster, au format Tiff (*.tif), AsciiGrid (*.asc) ou Idrisi (*.rst), dans lequel la valeur de chaque pixel correspond à une catégorie (occupation du sol ou autre type de classification). Ce raster définit l'emprise et la résolution spatiale pour l'ensemble du projet.

S'il est en format *.tif, sans extension Geotiff, le fichier doit être associé à un fichier de géoréférencement (*.tfw) structuré comme suit :

Exemple	
10.0	Taille du pixel en X
0.0	Rotation sur les lignes (0 pour les bonnes images)
0.0	Rotation sur les colonnes (0 pour les bonnes images)
-10.0	Taille du pixel en Y
821755.0	Coordonnée en X du pixel haut gauche
2342995.0	Coordonnée en Y du pixel haut gauche

S'il est en format *.rst, le fichier doit être associé à un fichier de géoréférencement (*.rdc) suivant la structure proposée dans le logiciel Idrisi.

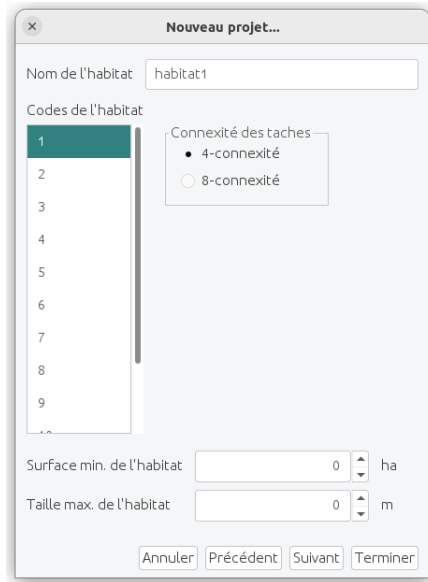
Important ! L'unité du système de coordonnées de l'image doit être en mètre. Si ce n'est pas le cas, les unités de surface et de distance seront erronées. Il est possible de reprojeter l'image dans une projection métrique (Lambert93, UTM, ...) à l'aide d'un SIG.

Code hors zone : code des pixels correspondant à l'absence de valeurs dans le fichier raster.

A partir de ces informations, le projet peut être créé en cliquant sur le bouton **Terminer**, ou bien on peut continuer par la définition de l'habitat en cliquant sur le bouton **Suivant**.

2.1.2 Définition d'un habitat

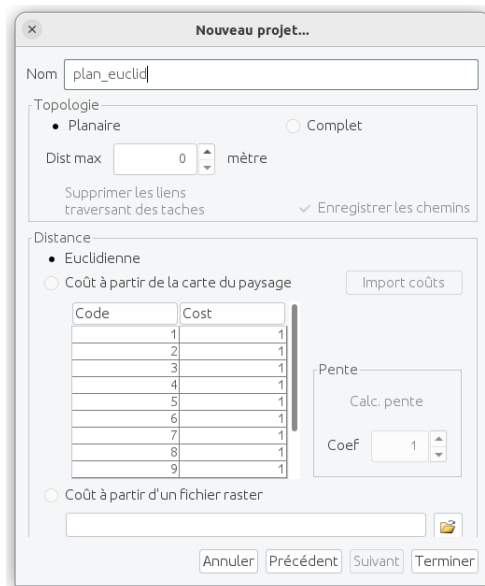
A partir de la carte du paysage, l'habitat peut être défini à partir de plusieurs paramètres détaillés dans la section [Création d'un habitat raster](#).



A partir de ces informations, le projet peut être créé avec un habitat en cliquant sur le bouton **Terminer**, ou bien il est possible de continuer par la définition d'un jeu de liens en cliquant sur le bouton **Suivant**.

2.1.3 Création d'un jeu de liens

La troisième fenêtre concerne la création d'un jeu de liens à partir de l'habitat défini précédemment. La création de ce jeu de liens constitue la dernière étape dans la mise en place initiale du projet. Cependant, il est possible de créer à tout moment de nouveaux jeux de liens dans le même projet (cf. [Jeu de liens](#))



Il est souvent plus simple et plus rapide de commencer par un jeu de liens planaire en distance euclidienne (paramètres par défaut).

A partir de ces informations, le projet peut être créé avec un habitat et un jeu de liens en cliquant sur le bouton **Terminer**.

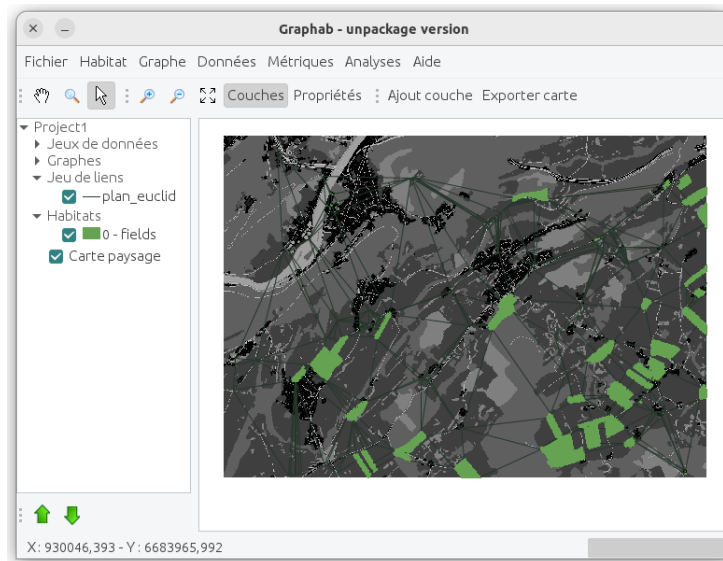


Figure 2.1: Le nouveau projet avec la carte de paysage, l'habitat et le jeu de liens

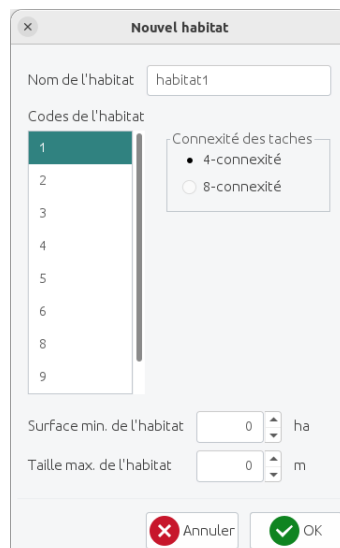
2.2 Habitat

Dans Graphab, un habitat correspond à un ensemble de zones (appelée taches d'habitat) représentant les zones d'habitat favorables pour une espèce, un groupe d'espèce ou un type générique.

Graphab 3 permet de créer plusieurs habitats dans un même projet. Ces habitats peuvent être créés à partir de la carte de paysage en raster (cf. [Création d'un habitat raster](#)) ou bien à partir d'une couche vectorielle (cf. [Création d'un habitat vecteur](#))

2.2.1 Création d'un habitat raster

La création d'un habitat à partir de la carte de paysage se fait par le menu **Habitat / Nouvel habitat raster**.



A partir de la carte du paysage, l'habitat peut être défini avec les paramètres suivants :

Nom de l'habitat : nom du nouvel habitat ; éviter les espaces et les caractères spéciaux. Il doit être unique dans le projet.

Code de l'habitat : valeurs des pixels correspondant à l'habitat utilisé pour la définition des taches. Plusieurs valeurs peuvent être sélectionnées en maintenant la touche **Ctrl**.

Surface min. de l'habitat : surface minimale en hectare pour qu'une tache d'habitat devienne un nœud du graphe.

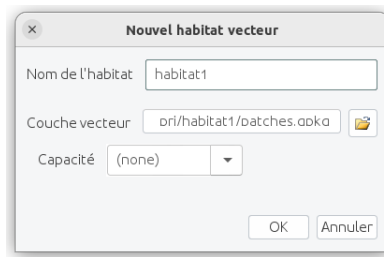
Taille max. de l'habitat : longueur en mètre au delà de laquelle la tache est redécoupée en plusieurs morceaux. Cette option est utile pour éviter des taches d'habitat couvrant un espace trop important.

Connexité des taches :

- 4-connexité : une tache est constituée du pixel central et de ses 4 voisins s'ils ont la même valeur ;
- 8-connexité : une tache est constituée du pixel central et de ses 8 voisins s'ils ont la même valeur.

2.2.2 Création d'un habitat vecteur

Il est aussi possible de créer un habitat à partir d'une couche vectorielle, indépendante de la carte de paysage, à partir du menu **Habitat / Nouvel habitat vecteur**.



Paramètres :

Nom de l'habitat : nom du nouvel habitat ; éviter les espaces et les caractères spéciaux.

Couche vecteur : couche au format vectoriel contenant les taches d'habitat. L'emprise de la couche doit être incluse dans l'emprise de la carte de paysage.

Capacité : attribut de la couche renseignant la capacité de chaque tache. Ce paramètre est optionnel et peut être laissé à **(None)**, dans ce cas la capacité correspondra à la surface en m^2 des taches.

2.2.3 Capacité des taches

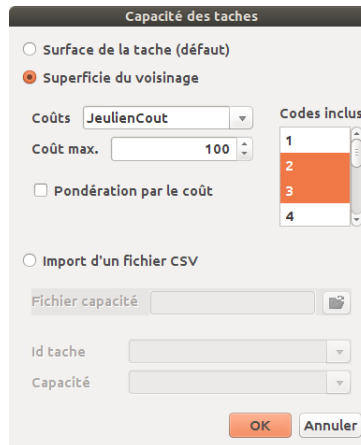
La capacité d'une tache traduit sa « qualité intrinsèque », considérée comme un indicateur de son potentiel démographique. Ainsi une tache avec une forte capacité pourra accueillir une forte population et inversement. La capacité intervient directement dans le calcul de certaines métriques de connectivité « de surface » et « pondérées » (voir le chapitre [Métriques](#)).

Par défaut (lors de la création de l'habitat), la capacité d'une tache est égale à sa superficie en m^2 . Cependant, il est possible de remplacer la superficie par n'importe quel autre indicateur de qualité. En effet, dans certains cas, la présence d'une espèce n'est pas liée à la superficie de la tache mais à la superficie d'autres types d'occupation du sol autour de la tache. Par exemple la présence d'amphibiens dans un étang de reproduction ne dépend souvent pas de la taille du plan d'eau mais de la superficie de l'habitat terrestre autour de ce plan d'eau.

Avec le menu **Habitat / Capacité des taches**, il est possible de modifier la capacité de l'ensemble des taches d'un habitat par deux méthodes.

Capacité calculée comme une fonction du voisinage

La première méthode consiste à définir la capacité des taches comme une fonction de la composition de leur voisinage et de procéder à ce calcul directement dans Graphab.



L'utilisateur doit sélectionner plusieurs critères : type de distance, distance maximale, catégories paysagères.

Coûts : il s'agit de la métrique spatiale (euclidienne ou ensemble de coûts) correspondant à un des jeux de liens disponibles dans le projet. Il est à noter que l'usage des coûts dans cette procédure revient à définir un voisinage anisotropique autour des taches, ce qui peut différer fortement d'une fonction de type buffer dans un SIG. Pour des raisons de cohérence, il est conseillé d'utiliser le même type de distance que celui qui a permis de définir les liens du graphe.

Coût max. : distance maximale du voisinage autour de la tache.

Codes inclus : l'utilisateur peut choisir une ou plusieurs catégories paysagères, autre que la catégorie « habitat », à prendre en compte dans le calcul de la capacité.

L'option **Pondération par le coût** permet d'introduire une pondération en fonction de l'éloignement à la tache, par le biais d'une fonction exponentielle négative. De cette façon, les surfaces sélectionnées compteront plus si elles sont proches de la tache et inversement.

Les valeurs de capacité ainsi calculées remplacent les surfaces des taches pour tous les calculs qui suivront, mais il est possible de revenir à l'état initial par le menu **Habitat / Capacité des taches** et en cochant l'option **Surface de la tache**.

Capacité définie à partir de données externes

L'autre possibilité est d'ouvrir un tableau de données au format CSV, décrivant toutes les taches du projet et contenant comme attribut les valeurs de capacité définies au préalable par l'utilisateur. Les identifiants du tableau importé doivent être ceux qui ont été définis lors de la création initiale de l'habitat.

Comme pour l'option précédente, les valeurs de capacité du tableau importé remplacent les surfaces des tâches pour tous les calculs qui suivront cette importation, mais il est possible de revenir à l'état initial en cochant l'option **Surface de la tâche**.

2.2.4 Méta-tâches

À partir du menu Habitat, l'entrée **Créer des méta-tâches** permet de créer un nouvel habitat où les tâches correspondent à des ensembles de tâches connectées à une certaine distance. Cette fonction permet de mieux prendre en compte l'emboîtement des échelles dans les processus écologiques et notamment le fait que la définition fonctionnelle d'une tâche d'habitat varie selon l'échelle considérée (Theobald, 2006 ; Zetterberg et al., 2010).

Pour utiliser cette fonction, il est nécessaire de créer un graphe élagué à la distance souhaitée (par exemple à la distance quotidienne). Chaque composante de ce graphe élagué correspondra à une méta-tâche. Celle-ci est donc constituée de l'ensemble des tâches d'habitat connectées entre elles.

La capacité des méta-tâches est définie par défaut comme la somme de la capacité des tâches composant la méta-tâche. L'autre option "Distance pondérée" permet de tenir compte de l'éloignement des tâches entre elles. Pour cette option, il faut définir le paramètre α en donnant une distance d et une probabilité p associée (cf. [Paramétrage des métriques pondérées](#)). La capacité résultante (C) d'une méta-tâche s'écrit :

$$C = \frac{1}{n} \sum_i^n \sum_j^n c_j e^{-\alpha d_{ij}}$$

L'option **Capacité min** permet de ne garder pour le nouvel habitat que les méta-taches qui ont une capacité supérieure ou égale au seuil défini.

Le nouvel habitat est enregistré dans le projet et est affiché automatiquement à la fin du traitement.

Références : [Clauzel et al.(2015b)]

2.2.5 Fonctions complémentaires (menu contextuel)

Certaines fonctionnalités sont accessibles par le menu contextuel de chaque habitat dans l'arborescence du projet (clic droit sur le nom d'un habitat).

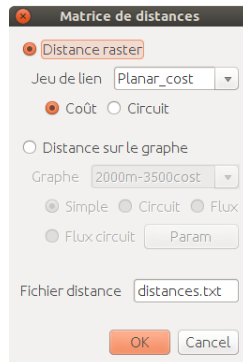
Suppression de taches

Après avoir défini une capacité différente de la surface de la tache, il peut être intéressant de supprimer les taches dont la capacité est considérée comme trop faible pour correspondre aux besoins l'espèce étudiée. L'entrée **Supprimer des taches** du menu contextuel, permet de créer un nouvel habitat identique à l'habitat sélectionné en supprimant les taches ayant une capacité inférieure à un seuil donné.

Après avoir renseigné la capacité minimale voulue, un nouvel habitat est créé dans le projet portant le même nom que l'habitat d'origine suffixé par le seuil de capacité minimale.

Matrice de distances inter-taches

L'entrée **Matrice de distance** du menu contextuel d'un habitat permet de calculer différentes distances entre toutes les taches de l'habitat. Cette fonction est par exemple utile pour comparer ces distances avec des distances génétiques.



Les distances peuvent être calculées de 2 manières : directement sur le raster ou bien à travers un graphe.

Distance raster En mode raster, les distances sont calculées en fonction d'un jeu de liens de référence. Si ce jeu de liens n'est pas en distance euclidienne, le calcul intègre les coûts associés aux catégories de la carte de paysage, tels qu'ils ont été définis dans le jeu de liens. Le résultat est une matrice de distance indépendante du graphe ; cette matrice correspond au calcul offert par les Systèmes d'Information Géographique.

L'option **Circuit** permet de calculer la "distance de résistance" entre chaque point, c'est-à-dire la résistance équivalente d'un réseau de résistances électriques. Cette méthode permet de tenir compte de l'ensemble des chemins possibles et non pas uniquement le plus court (cf. [McRae et al.(2008)]).

Distance sur le graphe En mode graphe, les distances sont calculées en fonction du graphe sélectionné.

- **Coût** : distance calculée suivant le plus court chemin sur le graphe de référence. Le type de distance correspond à celui qui a été utilisé pour la définition du jeu de liens qui a servi à définir le graphe

de référence. Suivant le choix qui a été fait au moment de la création du graphe, le calcul intègre éventuellement les distances intra-taches.

- **Circuit** : calcule la "distance de résistance" entre les noeuds du graphe. Le graphe est vu comme un réseau électrique où chaque lien du graphe représente une résistance. Cette option permet de tenir compte de l'ensemble des chemins possibles et non pas uniquement du plus court.
- **Flux** : fonctionnalité en test
- **Flux circuit** : fonctionnalité en test

Suppression d'un habitat

L'entrée **Supprimer** permet de supprimer l'habitat sélectionné. Le projet est enregistré automatiquement après cette suppression. L'ensemble des jeux de liens, graphes et métriques calculées à partir de cet habitat sont automatiquement supprimés !

Propriétés d'un habitat

L'entrée **Propriétés** rappellent les paramètres utilisés pour la construction de l'habitat ainsi que le nombre de taches.

2.3 Jeu de liens

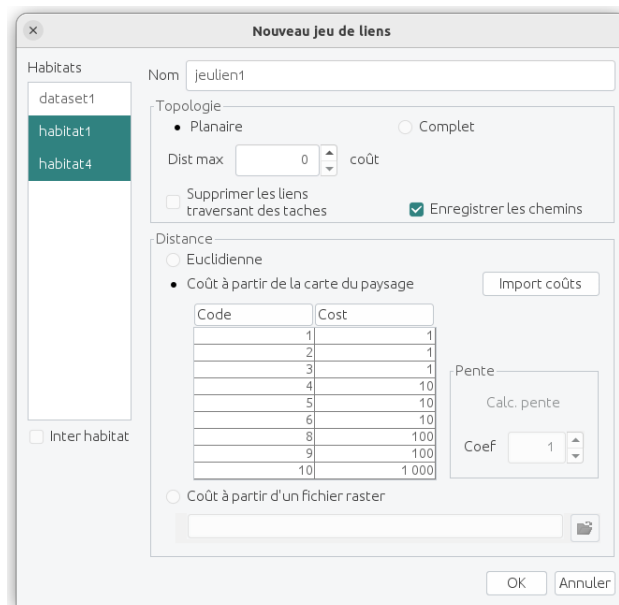
Un jeu de liens correspond à des chemins permettant de relier les taches d'habitat entre elles. Il dépend d'un ou plusieurs habitats.

2.3.1 Création d'un jeu de liens

La création d'un nouveau jeu de liens se fait à partir du menu **Graphe / Créer un jeu de liens**.

Le premier élément à définir est le ou les habitats à intégrer dans ce jeu de liens. Pour sélectionner plusieurs habitats, il faut utiliser la touche **Ctrl**.

Nom : nom du nouveau jeu de liens ; éviter les espaces et les caractères spéciaux. Il doit être unique dans le projet.



Topologie des liens

Deux topologies sont disponibles :

- **Planaire** : seuls les liens formant un graphe planaire minimal sont pris en compte. Cette topologie est mise en place par l'intermédiaire de polygones de Voronoï autour de chaque tache d'habitat. Ces polygones sont définis à partir des bords des taches, en distance euclidienne.
- **Complet** : tous les liens entre les taches sont potentiellement pris en compte.

Inter habitat : dans le cas d'un jeu de liens multi-habitat, la case à cocher **Inter habitat** permet de modifier la topologie en conservant uniquement les liens entre tache d'habitats différents.

Attention, cette option doit être utilisée normalement avec une topologie complète !

Dist max : cette option permet de préciser un seuil de distance au-delà duquel les liens ne sont plus créés. Elle permet ainsi de limiter le nombre de liens créés et d'accélérer la création des liens. L'unité de la distance dépend du type de distance : en mètre pour la distance euclidienne et en coût pour la distance de moindre coût.

Supprimer les liens traversant les taches : cette option, décochée par défaut, permet de ne pas créer de lien entre deux taches (A et C sur la figure suivante) passant par une tache intermédiaire (B). Ceci est recommandé pour le calcul de la métrique de centralité intermédiaire (BC) pour prendre en compte la fréquence avec laquelle chaque tache se trouve sur les plus courts chemins liant les paires de nœuds sur le graphe. A noter : cette option fournit une approximation de la distance intra-taches entre A et C. Si la case est décochée, un lien entre A et C passant par B sera créé et représentera la distance complète réelle entre les deux taches.

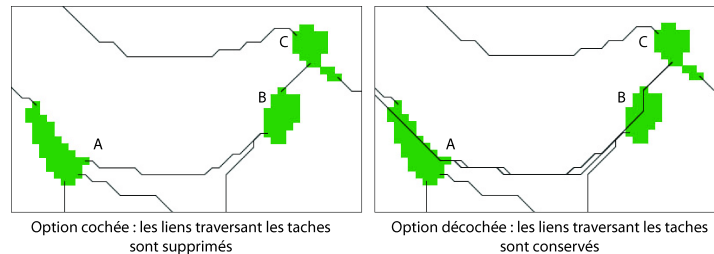


Figure 2.2: Illustration de l'option « Supprimer les liens traversant les taches »

Enregistrer les chemins :

- si la case est cochée : les liens sont enregistrés sous forme de chemins représentant le tracé réel du lien entre deux taches.
- si la case est décochée : les liens sont enregistrés sous forme topologique uniquement. Dans ce cas, la visualisation des liens en vue réaliste n'est plus possible. Ceci est recommandé lorsque le graphe contient de nombreux liens (dans le cas d'un graphe complet non seuillé par exemple) afin de limiter l'utilisation de la mémoire. A noter : si les chemins ne sont pas enregistrés, il n'est pas possible d'inclure les distances intra-taches dans le calcul des métriques.

Distance (ou impédance des liens)

Les distances sont calculées de bord à bord entre les taches. Deux principaux types de distance sont possibles : distance euclidienne et distance de moindre coût.

- **Distance euclidienne** : les liens sont calculés en distance euclidienne (parcours à vol d'oiseau entre les taches), ce qui revient à considérer la matrice comme uniforme.

- Distance de moindre coût : les liens sont calculés en distance coût. Cette distance correspond à la somme des coûts de tous les pixels du chemin parcouru. L'hétérogénéité de la matrice est prise en compte en assignant des valeurs de coût (ou résistance) aux classes paysagères. L'utilisateur peut activer cette possibilité de deux façons :

1. soit en indiquant les coûts correspondant aux catégories de la carte du paysage dans le tableau ;
2. soit à partir d'un fichier raster externe (format *.tif, *.asc ou *.rst) contenant pour chaque pixel une valeur de résistance.

Import coûts : en cliquant sur ce bouton, les coûts définis pour un autre jeu de liens de ce projet ou d'un autre projet peut être chargé automatiquement, évitant la saisie des valeurs de coûts.

Pour chaque lien créé, sa longueur métrique (**DistM**) et sa distance en unité de coût (**Dist**) sont enregistrées et consultables dans les propriétés (cf. [Propriétés des éléments sur la carte](#)).

Pente Les valeurs de résistance définies précédemment peuvent être modifiées en fonction de la pente [Clauzel et al.(2015b)]. Pour que la pente puisse être calculée, il faut tout d'abord charger un MNT à partir du menu Données | Importer un MNT.

Le paramètre coef (c) permet d'ajuster l'importance de la pondération par la pente (p). Pour un pixel donné, la résistance résultante (r_{final}) est calculée comme suit :

$$r_{final} = r * (1 + c.p)$$

avec $p = \frac{h}{l}$ la pente exprimée par le rapport hauteur(h) sur longueur (l).
 $p = 0$ pour une pente nulle et $p = 1$ pour une pente de 100% ($h = l$).

Pour $c = 1$ la valeur de résistance est doublée avec une pente de 100%.

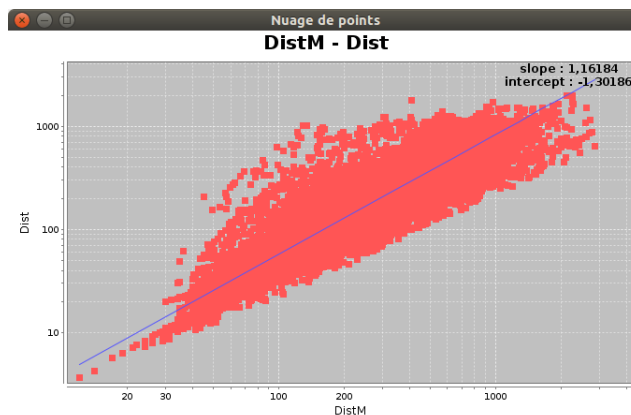
Pour $c = 10$ la valeur de résistance est doublée avec une pente de 10%.

2.3.2 Conversion des distances

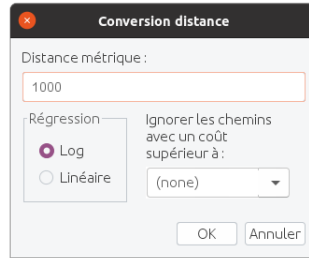
Dans le cas d'un jeu de liens en distance coût, il est souvent utile de convertir une distance métrique en un équivalent en distance coût, pour élaguer le graphe ou bien pour le paramétrage de distance d'une métrique.

Il est possible d'obtenir une approximation de la distance métrique (appelée **DistM**) exprimée en coûts cumulés (appelée **Dist**) en représentant tous les liens inclus dans le jeu de liens dans un nuage de points double logarithmique, et en utilisant l'équation ci-dessous pour effectuer cette conversion :

$$Dist = e^{slope.log(DistM)}$$



L'estimation peut être réalisée directement dans le logiciel à partir du menu **Conversion distance** présent dans le menu contextuel d'un jeu de liens (clic droit sur le jeu de liens).



Le paramètre "Ignorer les chemins avec un coût supérieur à" permet d'exclure de la régression les chemins traversant des pixels avec des coûts supérieurs ou égaux à la valeur renseignée. Cette option est utile pour exclure de la conversion de distance les liens traversant des éléments barrière ou quasi infranchissable. Ce paramètre ne fonctionne pas pour des jeux de liens ayant un raster de coût externe.

2.3.3 Fonctions complémentaires (menu contextuel)

Certaines fonctionnalités sont accessibles par le menu contextuel de chaque jeu de liens dans l'arborescence du projet (clic droit sur le nom du jeu de liens).

Conversion des distances

L'entrée **Conversion distance** permet de convertir une distance métrique en un équivalent en distance coût cf. [Conversion des distances](#).

Extraction des coûts

Fonctionnalité en test.

Suppression d'un jeu de liens

L'entrée **Supprimer** permet de supprimer le jeu de liens sélectionné. Le projet est enregistré automatiquement après cette suppression. L'ensemble des graphes et métriques calculées à partir de ce jeu de liens sont automatiquement supprimés !

Propriétés d'un jeu de liens

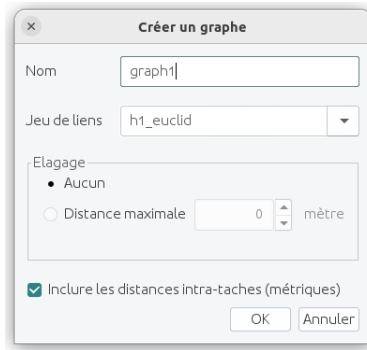
L'entrée **Propriétés** rappellent les paramètres utilisés pour la construction du jeu de liens ainsi que le nombre de liens qu'il contient.

2.4 Graphes

2.4.1 Création d'un graphe

Un projet Graphab peut donner lieu à la création de plusieurs graphes. Chaque graphe est construit à partir d'un jeu de liens.

La création d'un graphe s'effectue par le menu : **Graphe / Créer un graphe**.



Le graphe à créer doit tout d'abord être nommé. Il faut éviter les espaces, les caractères spéciaux et le nom doit être unique dans le projet.

L'utilisateur doit ensuite sélectionner un des jeux de liens créés précédemment (cf. [Jeu de liens](#)). Un graphe ne peut être basé que sur un seul jeu de liens. Pour créer un graphe basé sur plusieurs jeu de liens voir [Fusion de plusieurs graphes](#).

Elagage Il est possible de choisir d'élaguer ou non le graphe :

- **Aucun** : tous les liens du jeu de liens sont intégrés au graphe.
- **Distance maximale** : les liens retenus sont ceux avec une distance inférieure ou égale à la distance choisie.

Pour un graphe élagué par une distance maximale, l'unité de cette distance dépend du type de distance utilisé dans la création du jeu de liens. Si le jeu de liens a été créé en distance euclidienne, la distance maximale est exprimée en mètres. Si la création du jeu de lien a été réalisée en distance-coût, la distance maximale est exprimée en coûts cumulés. L'unité (mètre ou coût) du jeu de liens sélectionné est rappelé à côté du champ "Distance maximale".

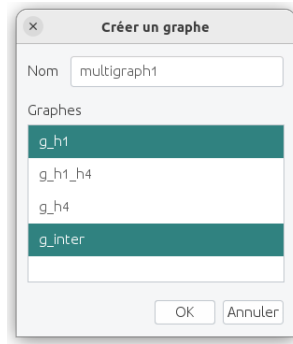
Il est possible d'obtenir une approximation de la distance métrique exprimée en coûts cumulés à partir du menu **Conversion distance** présent dans le menu contextuel d'un jeu de liens cf. [Conversion des distances](#).

Pour réaliser une analyse multiscalaire, il est souvent nécessaire de créer une série de graphes élagués dont la distance maximale est définie de façon croissante. L'utilisateur peut créer manuellement cette série de graphes, mais si l'objectif est d'analyser le comportement d'une métrique de connectivité en fonction de la distance maximale, il est préférable d'utiliser le menu : **Métriques / Graphes par lot** (cf. [2.5.6](#)).

Inclure les distances intra-taches : si la case est cochée, le calcul des métriques tient compte des distances intra-taches (recommandé) ; dans le cas contraire, seules les distances inter-taches sont prises en compte.

2.4.2 Fusion de plusieurs graphes

Graphab permet de fusionner plusieurs graphes en un seul. L'intérêt principal est de pouvoir regrouper plusieurs jeux de liens dans un même graphe. Cette fonction est accessible depuis le menu **Graphe / Fusionner des graphes**.



Il faut nommer le nouveau graphe et sélectionner les graphes à fusionner avec la touche **Ctrl**.

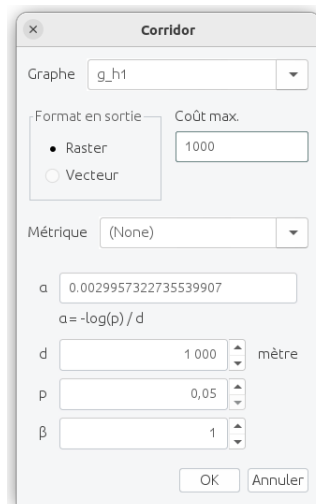
2.4.3 Corridors

Graphab permet de calculer les corridors représentant, pour une distance maximale donnée d_{max} , l'espace qui peut être traversé entre 2 taches d'habitat, *ie.* l'espace qui représente l'ensemble des chemins possibles reliant 2 taches et ayant une distance inférieure à d_{max} .

Cette fonctionnalité est accessible depuis le menu **Graphe / Corridor**.

Les corridors sont calculés à partir d'un graphe pour la définition des coûts ainsi que des liens.

La distance coût maximale d_{max} doit être renseignée avant de lancer le calcul. Dans le cas où le graphe est défini en distance euclidienne, tous les coûts sont définis à la résolution de la carte de paysage.



En fonction du type de sortie raster ou vecteur le paramétrage est différent.

Sortie raster

Dans le cas d'une sortie raster, plusieurs paramètres supplémentaires doivent être renseignés : le paramètre de décroissance de la distance α à partir d'une distance d et d'une probabilité p (cf. [Paramétrage des métriques pondérées](#)), et une métrique locale calculée sur les liens du graphe.

Quelque soit le paramétrage choisi, le principe général est le même. Un gradient de probabilité de déplacement est calculé entre chaque paire de taches d'habitat ayant un lien dans le graphe. La transformation utilisée pour créer ce gradient à partir des distances-coût est basé sur la fonction exponentielle négative, en considérant tous les chemins potentiels inférieurs à d_{max} et pas seulement le chemin de moindre coût. Cette transformation

tient compte de la déviation effectuée par l'espèce en passant par un point quelconque k n'appartenant pas au chemin de moindre coût (cf. figure 2.3). Ce gradient obtenu est multiplié par une valeur dépendant du paramétrage (métrique ou β). Enfin, pour obtenir un seul raster en sortie, l'ensemble des gradients calculés sont superposés, et en chaque point, la valeur maximale est conservée (cf. figure 2.4).

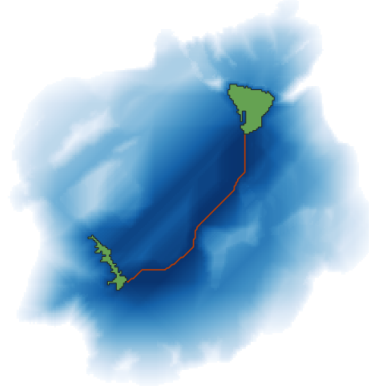


Figure 2.3: Gradient entre 2 taches d'habitat. Le chemin de moindre coût est affiché en orange

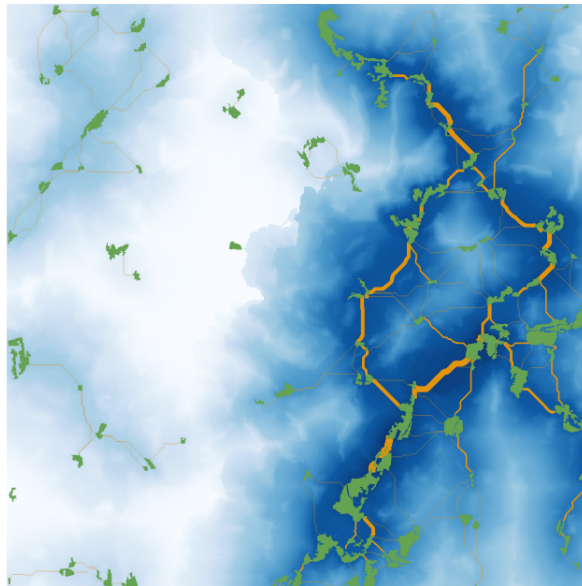


Figure 2.4: Résultat d'un calcul de corridor raster paramétré avec une métrique. Les chemins de moindre coût sont affichés en orange avec une taille proportionnelle à la valeur de la métrique

Le détail de la méthode de calcul selon le paramétrage est précisé dans les 2 paragraphes suivant.

Métrique locale Dans le cas où une métrique locale est sélectionnée le calcul du gradient G entre deux taches i et j au point quelconque k est défini tel que :

$$Gk_{ij} = M_{ij}e^{-\alpha(d_{ikj}-d_{ij})}$$

Avec M_{ij} : valeur de la métrique locale M sur le lien entre les taches i et j , d_{ij} : distance entre les taches i et j , d_{ikj} : distance entre la tache i et j en passant par le point k , α : paramètre de décroissance de la probabilité à mesure que la distance augmente (cf. [Paramétrage des métriques pondérées](#))

La matrice résultante représente un champ décroissant de valeurs (cf. figure 2.4). Les pixels situés sur le chemin de moindre coût ont une valeur égale à la métrique M_{ij} . Cette valeur diminue à 0 pour les pixels k ayant $d_{ikj} > d_{max}$.

Paramètre β Avec le paramètre β le calcul du gradient G entre deux taches i et j au point quelconque k est défini tel que :

$$Gk_{ij} = (a_i a_j)^\beta e^{-\alpha d_{ikj}}$$

Avec a_i : la capacité de la tache i , d_{ikj} : distance entre la tache i et j en passant par le point k , α : paramètre de décroissance de la probabilité à mesure que la distance augmente (cf. [Paramétrage des métriques pondérées](#))

Le gradient G résultant représente un champ décroissant de valeurs. Les pixels situés sur les chemins de moindre coût ont une valeur égale à $(a_i a_j)^\beta e^{-\alpha d_{ikj}}$. Cette valeur diminue à 0 pour les pixels k ayant $d_{ikj} > d_{max}$.

Résultat Quand chaque gradient Gk_{ij} est calculé, ils sont superposés et la valeur maximale est conservée :

$$Ck = \forall i, j \max Gk_{ij}$$

La couche corridor résultante correspond en chaque point k au gradient maximum en ce point.

Sortie vecteur

Dans le cas d'une sortie vecteur aucun autre paramètre n'est à renseigner.

Le résultat s'affiche dans une nouvelle couche vectorielle qui est automatiquement enregistrée dans le répertoire du projet. Cette couche contient pour chaque lien du graphe sélectionné un polygone représentant son corridor. Les liens ayant une distance supérieure à d_{max} n'auront pas de corridor.

L'affichage par défaut utilise la transparence pour faire apparaître les superpositions de corridors.

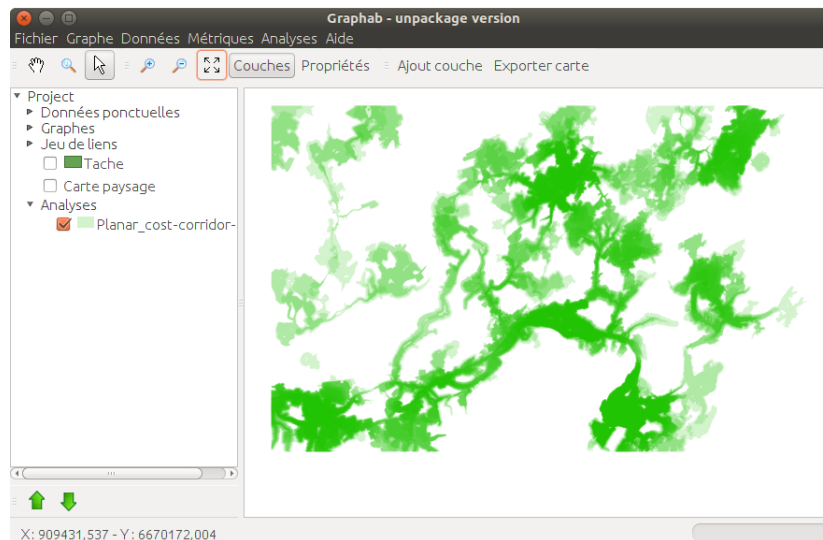


Figure 2.5: Résultat d'un calcul de corridor vecteur

2.4.4 Partitionnement de graphe

Graphab implémente un algorithme de partitionnement de graphe qui maximise l'indice de modularité [[Newman\(2006\)](#)]. La modularité est une mesure de la qualité d'un partitionnement des noeuds d'un graphe.

Le principe sous-jacent est qu'un bon partitionnement d'un graphe implique un nombre de liens à l'intérieur des groupes important et un nombre de liens inter-groupes faible. Le calcul utilisé est basé sur l'algorithme glouton, suivi d'une optimisation locale [Brandes et al.(2008)].

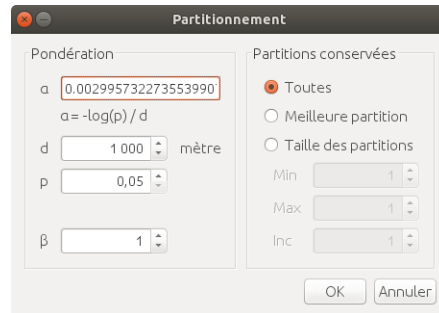
La modularité est calculée à partir d'un poids (w_{ij}) défini pour chaque lien du graphe :

$$w_{ij} = (a_i a_j)^\beta e^{-\alpha d_{ij}}$$

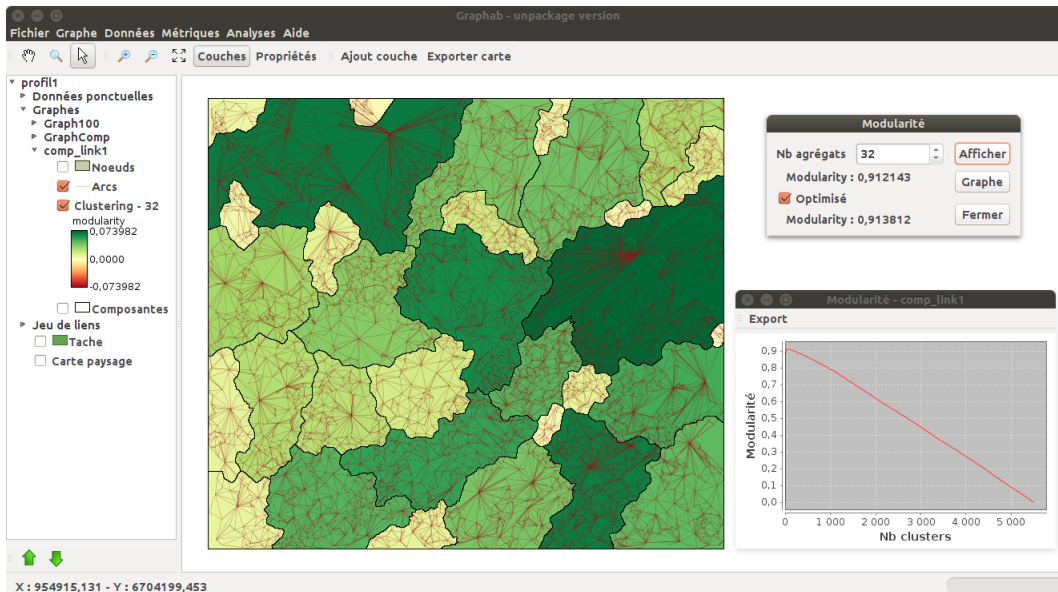
Les 2 paramètres β et α permettent de définir respectivement l'importance de la capacité des taches ($a_i a_j$) et l'importance de la distance (d_{ij}) pour le poids du lien w_{ij} . Si $\alpha = \beta = 0$ alors les poids sont tous identiques : $w_{ij} = 1$.

Le partitionnement est accessible depuis le menu contextuel d'un graphe (clic droit sur le nom d'un graphe). La première fenêtre demande de renseigner les paramètres α et β . Le paramètre α n'est pas renseigné directement, il est déterminé à partir de deux autres paramètres : une distance d et une probabilité p (cf. Paramétrage des métriques pondérées). Pour définir α à 0, il faut mettre p à 1.

La partie droite de la fenêtre permet de choisir quel(s) partitionnement(s) conservés. Par défaut tous les partitionnements calculés sont conservés en mémoire pour pouvoir les afficher par la suite. Ce paramétrage est utile pour éviter la saturation de la mémoire dans le cas de gros graphes (supérieur à 5000 noeuds). Dans ce cas, on peut conserver uniquement le meilleur ou bien donner un intervalle de tailles de partitions à conserver compris entre 1 et le nombre de noeuds du graphe.



Après validation de cette fenêtre, le calcul se lance jusqu'à obtenir un partitionnement maximisant la modularité. Le résultat s'affiche sur une nouvelle couche du graphe. Deux autres fenêtres viennent compléter ce résultat.



La nouvelle couche montre les groupes de la partition sous forme de polygones. Comme un groupe correspond à un ensemble de taches, il est représenté par un polygone agrégeant les polygones de Voronoï des taches affectées à celui-ci. La couleur d'un groupe correspond à sa valeur de modularité ; la modularité totale est la somme des modularités de tous les groupes.

Une première fenêtre contient un graphique montrant l'évolution de la modularité en fonction du nombre de groupes (clusters). Dans la figure précédente, on voit que la modularité maximale est atteinte pour un nombre groupes relativement faible. Il est possible de zoomer dans ce graphique pour visualiser plus précisément le pic de modularité.

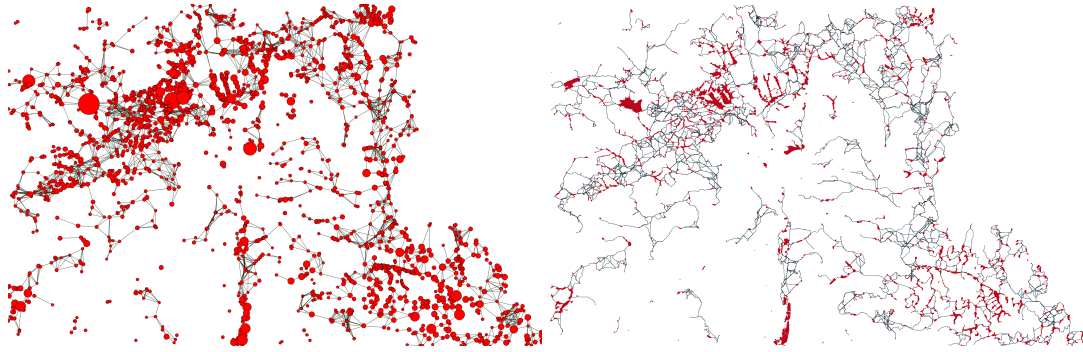
La deuxième fenêtre affiche le nombre de groupes (Nb agrégats) maximisant la modularité ainsi que la valeur de modularité. Plusieurs fonctionnalités sont disponibles sur cette fenêtre :

- le nombre de groupes peut être modifié pour visualiser des partitionnements sous optimaux. Le résultat peut être visualisé en cliquant sur le bouton **Afficher**. Une nouvelle couche de partitionnement est ajoutée au graphe.
- le bouton **Graphe** permet de créer un nouveau graphe ne contenant que les liens intra-groupes.
- la case à cocher **Optimisation** réalise une optimisation locale du partitionnement après l'algorithme glouton ce qui permet d'améliorer sensiblement la valeur de modularité. Cette option est active par défaut.

2.4.5 Visualisation des graphes

Les propriétés des graphes sont accessibles par un clic droit sur le nom du graphe. Deux modes de visualisation des graphes sont disponibles :

- La vue topologique affiche une vue simplifiée du graphe où les nœuds sont représentés par des cercles et les liens par des lignes droites de centroïde à centroïde.
- La vue réaliste affiche les taches d'habitat selon leurs limites réelles et les liens sont représentés par les chemins de moindre coût entre deux taches.



Gauche : vue topologique - Droite : vue réaliste

2.4.6 Fonctions complémentaires (menu contextuel)

Certaines fonctionnalités sont accessibles par le menu contextuel de chaque graphe dans l'arborescence du projet (clic droit sur le nom du graphe).

Vue topologique

Cette entrée permet de visualiser le graphe en vue topologique, voir [Visualisation des graphes](#).

Vue réaliste

Cette entrée permet de visualiser le graphe en vue réaliste, voir [Visualisation des graphes](#).

Partitionnement

Cette entrée permet de partitionner le graphe par un algorithme de modularité, voir [Partitionnement de graphe](#).

Matrice de distance

L'entrée **Matrice de distance** permet de créer un tableau indiquant la distance entre chaque paire de nœuds pour le graphe sélectionné. L'absence de connexion entre deux nœuds est notée NaN (Not a Number). Pour plus d'informations sur le paramétrage voir [Matrice de distances inter-taches](#).

Id composante

L'entrée **Id composante** permet de créer un attribut dans la couche habitat contenant pour chaque tache l'identifiant de la composante du graphe.

Suppression d'un graphe

L'entrée **Supprimer** permet de supprimer le graphe sélectionné. Le projet est enregistré automatiquement après cette suppression. L'ensemble des métriques calculées à partir de ce graphe sont automatiquement supprimées !

Propriétés d'un graphe

L'entrée **Propriétés** rappellent les paramètres utilisés pour la construction du graphe : le nom du graphe, le jeu de liens sélectionné, le type de graphe avec éventuellement la distance maximale utilisée et le nombre de liens.

2.5 Métriques de connectivité

2.5.1 Famille de métriques et niveaux de calcul

Chaque graphe créé dans un projet peut être utilisé pour calculer différentes métriques de connectivité, dont le détail de calcul et les références bibliographiques sont mentionnés dans le chapitre 7. Ces calculs s'effectuent à plusieurs niveaux correspondant aux grandes entrées du menu Métriques (tableau 7.1) :

- Métrique globale : ces métriques caractérisent le graphe entier.
- Métrique par composante : ces métriques caractérisent la connectivité interne de chaque composante (ou sous-graphe).
- Métrique locale : ces métriques caractérisent la connectivité de chaque élément du graphe, nœud ou lien.
- Delta-métrique : ces métriques caractérisent également chaque élément du graphe, mais à partir d'un mode de calcul spécifique. En utilisant la méthode dite de la « suppression » (suppression de nœud ou suppression de lien), l'importance relative de chaque élément du graphe est évaluée en calculant le taux de variation que sa suppression occasionne sur une métrique globale. Le résultat d'une delta-métrique est donc local, mais en référence au niveau global.

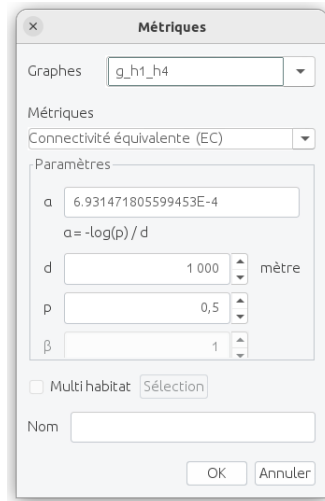
Dans la fenêtre de dialogue qui s'ouvre suite au choix d'un de ces quatre modes de calcul, trois familles de métriques sont disponibles :

- les métriques pondérées, qui sont fondées sur des critères de distance et de capacité des taches et qui nécessitent un paramétrage adapté à l'espèce de référence. Ces métriques nécessitent des calculs de chemin sur les graphes. En choisissant ces métriques, l'utilisateur doit indiquer le réglage souhaité.
- les métriques de surface, qui sont fondées en priorité sur un critère de surface. Dans le cas où la capacité ne correspond pas à la surface des taches mais à un autre critère, ces métriques peuvent être calculées et elles s'expriment dans l'unité du critère considéré.
- les métriques topologiques, qui sont issues de la théorie des graphes et qui ne nécessitent pas de paramétrage.

2.5.2 Paramétrage des métriques

Quel que soit le niveau choisi, l'utilisateur doit d'abord indiquer le graphe sur lequel le calcul va s'appliquer, puis choisir la métrique de connectivité souhaitée.

Selon la métrique sélectionnée des paramètres peuvent être à renseigner dans la zone **Paramètres** (cf. [Paramétrage des métriques pondérées](#)). Si le graphe contient plusieurs habitats, il est possible de sélectionner/filtrer le calcul de la métrique en fonction des habitats souhaités en cochant **Multi habitat** (cf. [Paramétrage multi-habitat](#)). Enfin un nom peut être donné au résultat du calcul de la métrique. Si aucun nom n'est donné, un nom sera généré automatiquement comprenant le nom court de la métrique, suivi de ses paramètres et du nom du graphe.



2.5.3 Paramétrage des métriques pondérées

Paramètre α

Plusieurs métriques intègrent dans leur calcul une pondération où les distances entre les taches sont converties en probabilité de déplacement. Ces métriques sont : F, IF, PC, EC, BC... La pondération est basée sur une fonction exponentielle de la forme :

$$p = e^{-\alpha d}$$

où p est la probabilité de déplacement entre deux taches, d la distance entre ces taches et α un paramètre contrôlant la vitesse avec laquelle la probabilité diminue quand la distance augmente. Comme la valeur du paramètre α n'est pas facile à déterminer, le logiciel propose de le calculer à partir des deux autres paramètres. L'utilisateur doit indiquer la distance correspondant à une certaine valeur de probabilité, par exemple :

- la distance de dispersion maximale de l'espèce correspondant à une faible valeur de p (0.05 ou 0.01).
- la distance de dispersion moyenne de l'espèce correspondant à une valeur médiane de p (0.5).

La valeur α est automatiquement obtenue à partir de la formule :

$$\alpha = -\log(p) / d$$

Paramètre β

Certaines de ces métriques F, IF, BC sont aussi contrôlées par le paramètre β . Ce paramètre est l'exposant appliqué à la capacité des taches. Il joue sur l'équilibre relatif entre le poids des distances et le poids des capacités des taches dans la pondération des métriques. En prenant l'exemple de la métrique F en calcul local, dont la forme générique est la suivante :

$$F = \sum a^\beta e^{-\alpha d}$$

- une valeur de $\beta = 0$ signifie que la capacité des taches ne joue pas de rôle dans la pondération.
- une valeur de $\beta = 1$ signifie que la capacité des taches joue de façon linéaire dans cette pondération.
- une valeur de $\beta = 2$ signifie que la capacité des taches est portée à la puissance 2.
- une valeur de $\beta = 0.5$ signifie que la racine carrée de la capacité des taches intervient dans la pondération ie. moyenne géométrique.
- une valeur de $\beta = -1$ signifie que la capacité des taches joue de façon inversement proportionnelle.

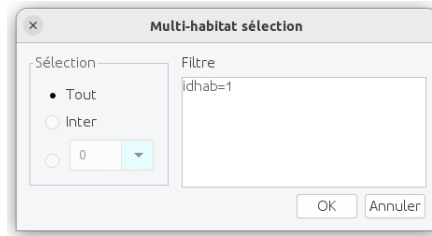
Au-delà de ces quelques exemples, toutes les valeurs de pondération sont possibles.

2.5.4 Paramétrage multi-habitat

Pour un graphe contenant plusieurs habitats et une métrique supportant la décomposition des habitats (cf. table 7.1), la case à cocher **Multi-habitat** peut être cochée. Quand elle est cochée, le résultat de la métrique sera décomposé pour chaque habitat.

Sélection

Le bouton **Sélection** permet de limiter le calcul à certains habitats ou à certains éléments du graphe.



La partie **Filtre** est activé uniquement pour les métriques locales ou delta, elle permet de filtrer les éléments du graphe pour lesquels un résultat sera calculé.

Exemples de filtre :

`idhab=0` : la métrique ne sera calculée que pour les noeuds de l'habitat 0

`id=12 || id=20` : la métrique ne sera calculée que pour les noeuds 12 et 20

`id="1-3"` : la métrique ne sera calculée que pour le lien 1-3

`node` : la métrique ne sera calculée que pour les noeuds du graphe

`edge` : la métrique ne sera calculée que pour les liens du graphe

2.5.5 Affichage des résultats

Après la fin d'un calcul de métrique, l'ensemble des résultats sont enregistrés automatiquement dans le projet. Après l'enregistrement, le résultat de la métrique s'affichera. Pour une métrique globale, une (ou quelques) valeurs seront affichées, pour une métrique par composante, les composantes du graphe s'afficheront avec un dégradé de couleurs représentant la valeur de la métrique, enfin pour une métrique locale ou delta métrique, les noeuds et les liens du graphe s'afficheront avec un dégradé de couleur sur les noeuds ou d'épaisseur sur les liens représentant la valeur de la métrique.

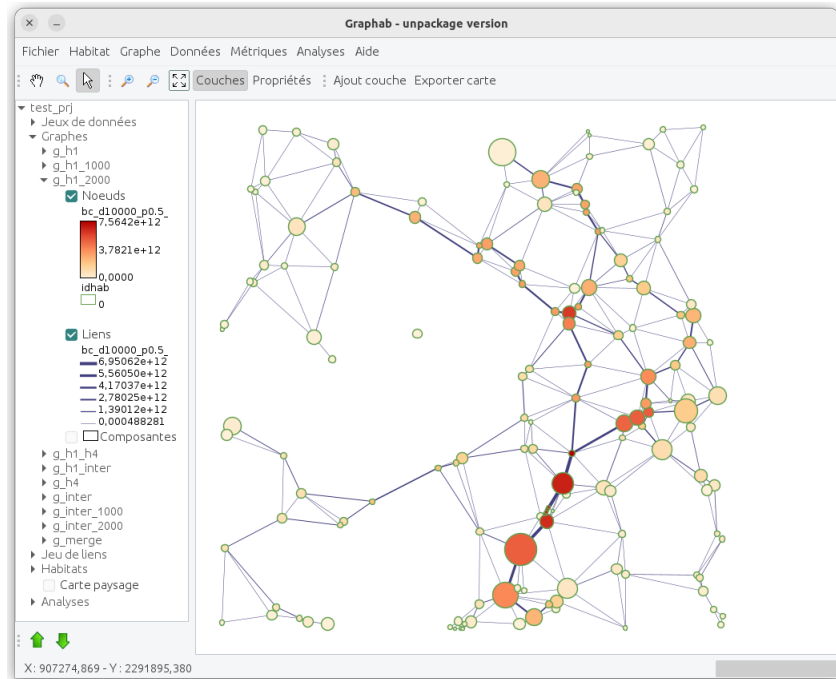


Figure 2.6: Affichage par défaut du résultat du calcul d'une métrique locale BC

L'ensemble des métriques calculées peuvent être listées à partir du menu **Métriques / Afficher les métriques**. Dans cette fenêtre, l'ensemble du paramétrage de chaque métrique calculée peut être retrouvé. Pour les métriques globales, le résultat est directement affiché, pour les autres métriques, il suffit de cliquer sur le bouton **Afficher** pour réafficher la métrique sur la carte avec le bon graphe.

Métrique	Type	Graphe	Résultat
f_d1000_p0.5_beta1	LOCAL	g_h1_inter	
bch_d1000_p0.5_beta1	LOCAL	g_h1_inter	
ec_d1000_p0.5	GLOBAL	g_h1	[2628795.5914445...
bch_d1000_p0.5_beta1	LOCAL	g_h1_h4	
fh_d1000_p0.5_beta1	LOCAL	g_inter	
delta_inter	DELTA	g_h1_h4	
f_d100_p0.5_beta1	LOCAL	g_h1_h4	
delta_ec	DELTA	g_h1_h4	
test	GLOBAL	g_h1	[2628795.5914445...
ec_d1000_p0.5	GLOBAL	g_h1_inter	[2835289.8525909...
f_d10_p0.5_beta1	LOCAL	g_h1_h4	
bch_d1000_p0.5_beta1	LOCAL	g_h1_h4	

F - LOCAL
Flux (F)
Paramètres : {d=100.0, p=0.5, beta=1.0}
Multi-habitat : 0
Attributs : [F_d100_p0.5_beta1]0_g_h1_h4
Filtre : idhab=1

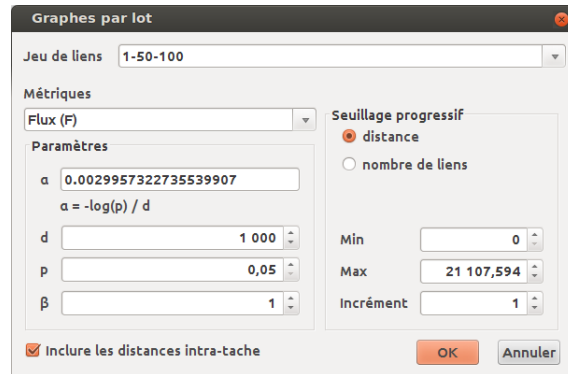
2.5.6 Calcul des métriques par lot

Chaque métrique compatible avec le niveau global peut être calculée en chaîne suivant la variation de l'échelle des distances. Cette variation peut concerner soit l'élagage du graphe (2.5.6), soit le paramétrage des métriques elles-mêmes (2.5.6). Le type de distance considéré pour l'élagage est celui qui a été défini pour le jeu de lien choisi, à savoir distance euclidienne, distance de moindre coût, chemin de moindre coût.

Graphes par lot

Le menu Métriques / Graphes par lot permet de créer une série de graphes élagués à partir d'un jeu de liens donné, et de calculer pour chacun une métrique de niveau global. Les distances maximales des graphes successifs sont définies de façon croissante, suivant deux modes :

- distance : un intervalle de distance fixe est déterminé entre les graphes successifs. Les valeurs de la métrique sont donc calculées à des intervalles réguliers de distance.
- nombre de liens : un nombre de liens fixe est défini entre les graphes successifs. Ce nombre de liens est automatiquement transformé en distance utilisée pour élaguer le graphe. Ces distances peuvent s'échelonner de façon irrégulière.



Les graphes sont définis suivant trois critères donnés par l'utilisateur :

- minimum : plus petite distance, utilisée pour le premier graphe de la série. Par défaut, ce minimum est à 0, ce qui correspond à l'absence totale de liens.
- maximum : distance maximale, utilisé pour le dernier graphe de la série. Par défaut, ce maximum correspond à la distance maximale, ou le nombre de lien, du jeu de lien sélectionné.
- incrément : valeur de distance ajoutée à chaque nouveau graphe.

Une fois le calcul terminé, le logiciel ouvre une nouvelle fenêtre dans laquelle la courbe de la métrique choisie est affichée en fonction de la distance. Les valeurs de cette courbe peuvent être enregistrées par le bouton export, en sélectionnant le format texte.

Paramètres par lot

Le menu Métriques / Paramètres par lot permet de calculer en série des métriques à partir d'un graphe donné. Cette possibilité s'applique exclusivement aux métriques pondérées. Elle se subdivise en deux entrées : métriques locales ou globales.

Paramètres par lot pour métriques locales

Une métrique pondérée de niveau local est calculée en série suivant la variation d'un de ses paramètres. L'utilisateur doit choisir le graphe, la métrique et le paramètre à faire varier. La variation du calcul est déterminée par :

- min : valeur minimale du paramètre
- max : valeur maximale du paramètre
- incrément : valeur d'intervalle entre deux calculs de la métrique

Une fois le calcul terminé, les taches (et dans certains cas les liens) du graphe sont caractérisées par une série de métriques supplémentaires.

Paramètres par lot pour métriques globales

Pour un graphe donné, une métrique pondérée de niveau global est calculée en série suivant la variation d'un de ses paramètres. Comme précédemment, cette variation est définie entre une valeur minimale (min), une valeur maximale (max) et un intervalle (incrément).

La procédure se termine par l'ouverture d'une nouvelle fenêtre dans laquelle la courbe de la métrique choisie est affichée en fonction du paramètre.

Le tableau 2.1 récapitule les possibilités de calcul des métriques.

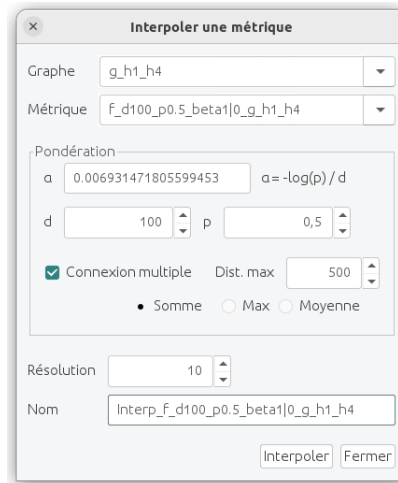
Type	Métrique de connectivité	Code	Capacité des taches	Distance intra-taches	Paramètres		Graphe par lot	Paramètre par lot
					α	β		
Métriques pondérées	Flux	F	×	×	×	×	×	×
	Connectivité équivalente	EC	×	×	×		×	×
	Probabilité de connectivité	PC	×	×	×		×	×
	Flux d'interaction	IF	×	×	×	×		×
	Fractions de delta Probabilité de connectivité	dPC	×	×	×			
	Centralité intermédiaire	BC	×	×	×	×		×
	Indice Intégral de Connectivité	IIC	×				×	
	Flux circuit	CF	×			×		×
Métriques de surface	Taille moyenne des composantes	MSC	×				×	
	Taille de la plus grande composante	SLC	×				×	
	Probabilité de coïncidence de classe	CCP	×				×	
	Taille de composante attendue	ECS	×				×	
Métriques topologiques	Degré	Dg						
	Coefficient de groupement	CC						
	Centralité de proximité	CCe		x				
	Excentricité	Ec		x				
	Corrélation de connectivité	CCor						
	Nombre de composantes	NC					×	
	Diamètre	GD			×		×	
	Indice d'Harary	H					×	
Lambda de Wilks	W					×		

Table 2.1: Possibilités de calcul des métriques de connectivité.

2.6 Analyses

2.6.1 Interpolation de métrique

Le menu **Analyses / Interpoler une métrique** permet de générer une couche raster à partir d'une métrique locale calculée au niveau des taches. Cette transformation est fondée sur une interpolation spatiale spécifique, qui permet d'attribuer les valeurs de connectivité des taches à chaque cellule d'une grille, en utilisant une fonction de pondération décroissante à partir de la bordure des taches (poids de 1). Globalement, plus les cellules sont situées à l'écart du graphe, plus elles obtiennent des valeurs faibles de connectivité.



La pondération est une fonction exponentielle négative de la forme $p = e^{-\alpha d}$ pour laquelle l'utilisateur doit choisir une distance (d) correspondant à une certaine probabilité (p) et le logiciel en déduit la valeur du paramètre α . En principe, ce réglage est à rendre cohérent avec le choix du graphe de référence ou le choix des éventuelles métriques pondérées qui sont à mobiliser, en utilisant la même valeur de d .

L'option **Connexion multiple** permet de faire intervenir plusieurs taches dans le calcul des métriques au niveau des points. Le calcul est fondé sur une somme, une moyenne pondérée ou le maximum des valeurs issues de toutes les taches comprises dans le voisinage des points, jusqu'à la Distance maximale indiquée.

Il est à noter que la distance considérée dans ces calculs dépend du graphe de référence. Si celui-ci est fondé sur une distance de moindre coût, l'interpolation spatiale utilise cette même distance et non une simple distance euclidienne.

2.6.2 Ajout de taches

La fonction **Ajout de taches** est accessible à partir du menu **Analyses**. Elle permet de tester un ensemble d'emplacements pour créer de nouvelles taches d'habitat en fonction du gain de connectivité qu'elles procurent. Le logiciel ajoute virtuellement une tache et les liens entre cette tache et les taches existantes (seulement si leur distance est inférieure à la distance maximale du graphe dans le cas d'un graphe élagué) et recalcule la métrique globale. Une fois tous les emplacements testés, le logiciel valide l'emplacement qui procure le gain maximal de connectivité. La procédure est répétée jusqu'à l'ajout du nombre souhaité de nouvelles taches.

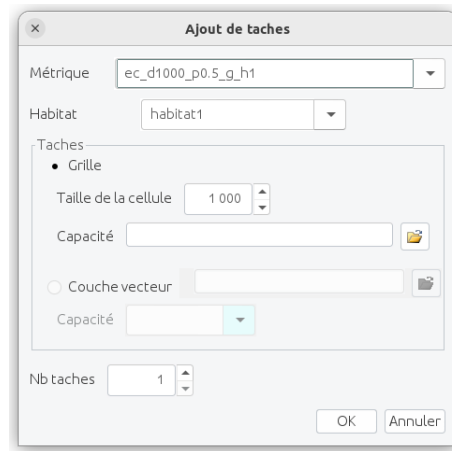
Plusieurs paramètres sont à définir :

- **Métrique** : la métrique globale que l'on cherche à maximiser. La métrique doit avoir été calculée en amont sur un graphe n'incluant pas les distances intra-taches et le jeu de liens sous-jacent doit être en topologie complète.
- **Habitat** : dans le cas où la métrique a été calculée sur plusieurs habitats, il faut sélectionner l'habitat qui contiendra les nouvelles taches.

- NB taches : nombre de taches (qui deviendront de nouveaux nœuds du graphe) que l'on souhaite ajouter.

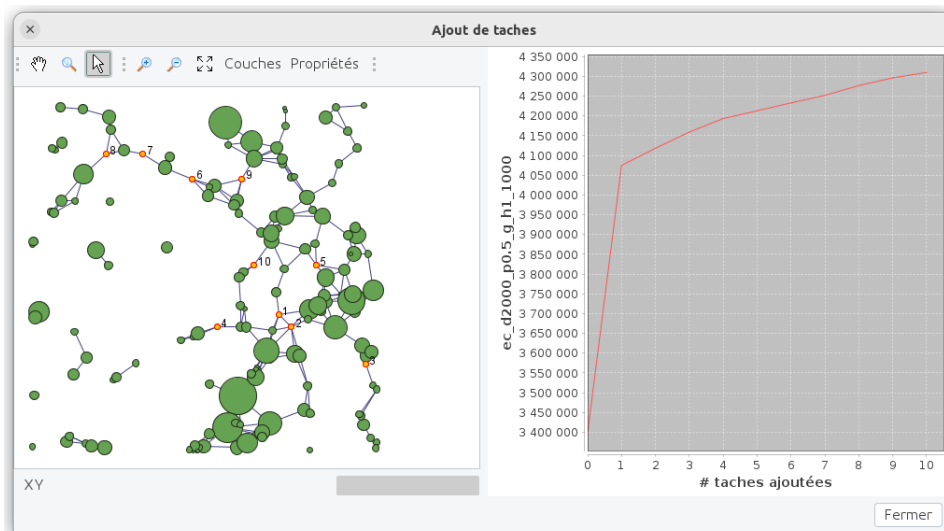
La définition du jeu de taches à tester peut se faire de deux manières :

- En appliquant automatiquement une grille d'une certaine résolution (champs "taille de la cellule" à renseigner). Dans ce cas, le logiciel teste chaque centroïde de cellule en ajoutant virtuellement une nouvelle tache. Par défaut, tous les centroïdes ont une valeur de capacité identique (=1). Il est néanmoins possible d'importer une couche raster où chaque pixel de la matrice paysagère a une valeur de capacité propre.
- En important une couche vecteur de points ou de polygones. Il faut dans ce cas indiquer la colonne de la table attributaire de cette couche correspondant à la capacité des taches.



À la fin du calcul une nouvelle fenêtre s'affiche avec 2 parties :

- à gauche le graphe avec les taches ajoutées numérotées de 1 à n.
- à droite l'évolution de la métrique globale après l'ajout de chaque nouvelle tache.



L'ensemble des résultats sont automatiquement enregistrés dans un répertoire nommé `addpatch_nX-metric_...` dans le répertoire du projet. Il contient :

- une couche contenant les nœuds ajoutés. La table attributaire indique l'étape à laquelle a été validé chacun des nœuds et la valeur de la métrique globale correspondante.
- deux couches contenant l'ensemble des liens du graphe, y compris ceux reliant les nouveaux nœuds, en version topologique `topo_links_metric.gpkg` ou en version réaliste `links_metric.gpkg`
- un répertoire `detail` contenant pour chaque étape la couche des emplacements testés et la valeur de la métrique globale correspondant.

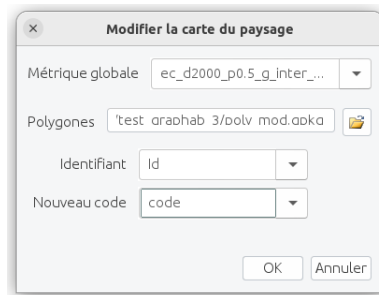
Références : [Clauzel et al.(2015a), Foltête et al.(2014)]

2.6.3 Modification de la carte de paysage

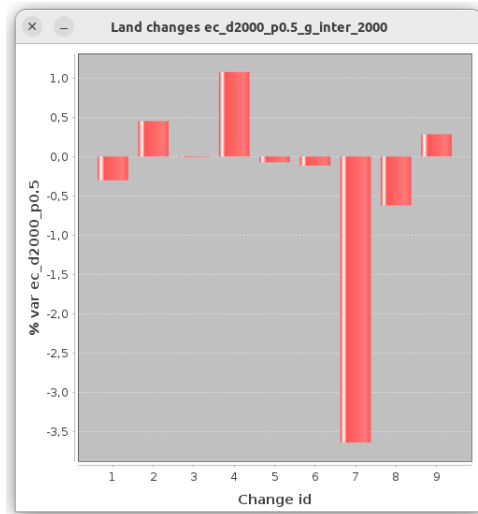
A partir du moment où des graphes et des métriques sont calculés dans Graphab, il peut être intéressant de tester des modifications de l'occupation du sol et mesurer son impact sur la connectivité du réseau écologique modélisé.

Graphab contient déjà plusieurs fonctionnalités spécifiques de modification du graphe (delta métrique, [Ajout de tâches](#) et plusieurs commandes en CLI [Ajout/Suppression d'éléments](#)). Cependant ces fonctions ne gèrent que la modification du graphe par l'ajout ou la suppression d'éléments, pour tester des scénarios plus fin il peut être plus intéressant de revenir à la carte de paysage elle-même pour définir précisément les changements sur l'occupation du sol. Ces changements pourront impacter les éléments du graphe de différentes manières : création/suppression d'éléments mais aussi la capacité des nœuds ou les distances des liens.

A partir du menu **Analyses/Modifier la carte du paysage**, il est possible de tester un ensemble de changements sur la carte du paysage et de mesurer leurs impacts sur une métrique de connectivité globale.



Il faut sélectionner une métrique globale déjà calculée sur un graphe, et une couche de polygones contenant les modifications à appliquer. Chaque modification à tester correspond à un ou plusieurs polygones avec le même identifiant et un code d'occupation du sol propre à chaque polygone qui sera remplacé sur la carte à l'emplacement du polygone. A partir de cette carte modifiée, un nouveau projet sera créé avec tous les éléments utiles pour recalculer la métrique globale sélectionnée (habitats, jeux de liens et graphes) dans un sous répertoire du projet nommé par l'identifiant. Cette procédure sera itérée pour chaque identifiant de la couche de polygones. En résultat, un graphique est affiché avec l'évolution de la métrique globale pour chaque identifiant de la couche de polygones. Un fichier `landmod-nom_métrique.csv` est automatiquement enregistré dans le dossier du projet, contenant la valeur de la métrique globale pour chaque identifiant.



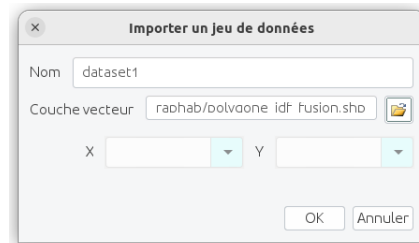
Comme cette méthode recrée un projet complet pour chaque test de modification, le traitement peut être très lourd et plus lourd que les méthodes d'ajout/suppression d'éléments sur le graphe.

2.7 Jeu de données externe

Le corps principal du logiciel est la mise en place des graphes et le calcul de métriques de connectivité, mais il est souvent très utile de relier ces éléments à des données externes. Le logiciel permet de faire interagir les données issues des graphes avec un jeu de données externe, ou même d'intégrer ce jeu de données directement dans un graphe.

2.7.1 Import de données externes

Des données externes peuvent être importées par le menu **Données / Importer un jeu de données**. Ces données doivent être sous forme d'une couche vectorielle, souvent ponctuelle.



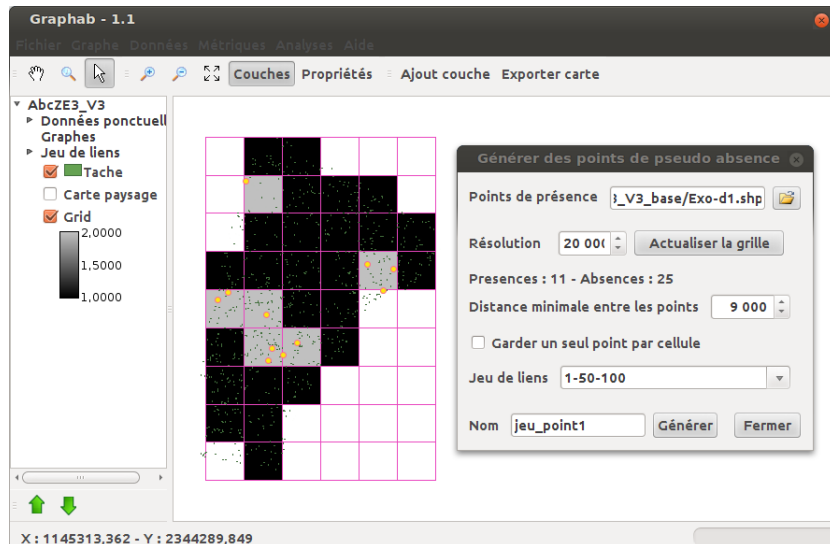
Le fichier importé peut être soit sous forme d'une couche vectorielle au format geopackage (*.gpkg) ou shapefile (*.shp), soit sous forme d'un tableau (*.csv). Dans le second cas, il faut indiquer les colonnes du tableau qui correspondent aux coordonnées X et Y des points.

Dans le projet, le nouveau jeu de données apparaît dans l'arborescence sous **Jeux de données**. Il correspond à un habitat particulier dont la capacité des taches est nulle. Cette abstraction permet de créer des jeux de liens, des graphes et de calculer des métriques sur ces jeux de données externes.

Si les données représentent des points de présence d'une espèce sans point d'absence, l'utilisateur peut créer un jeu de données de pseudo-absence cf. [Générer des points aléatoires](#).

2.7.2 Générer des points aléatoires

Le menu Données / Générer des points aléatoires permet de créer un ensemble de points de pseudo-absence en s'appuyant sur un ensemble de points de présence.



L'utilisateur doit charger un fichier de points de présence et indiquer le nom du jeu de points de présence / absence à créer.

Plusieurs paramètres sont à définir pour effectuer un tirage aléatoire des points d'absence :

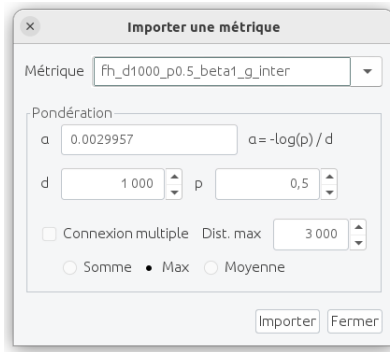
- Résolution de la grille (en mètres) pour définir la taille des cellules dans lesquelles seront éventuellement tirés des points d'absence. Le bouton « actualiser la grille » permet d'afficher la grille selon la résolution choisie
- Distance minimale entre les points (en mètres) : cette fonction permet d'atténuer les effets d'autocorrélation spatiale en restreignant la recherche des points d'absence générés aux zones situées à une distance minimale des points de présence.
- Type de distance : l'unité de la distance minimale entre les points dépend du type de distance utilisé dans le jeu de liens sélectionné.
- Garder un point par cellule : cette option permet de ne garder qu'un point de présence dans chaque cellule pour éviter les effets de grappe et atténuer l'autocorrélation spatiale.

2.7.3 Fonctions complémentaires (menu contextuel)

Certaines fonctionnalités sont accessibles par le menu contextuel de chaque jeu de données dans l'arborescence du projet (clic droit sur le nom du jeu de données).

Importer une métrique locale

Cette fonction permet de ramener une métrique locale calculée sur les noeuds d'un graphe aux éléments du jeu de données. Le calcul utilise le même principe que l'interpolation de métrique (cf. section 2.6.1). Mais au lieu d'interpoler une métrique en tout point de l'espace, la métrique sera interpolée uniquement pour les éléments du jeu de données, ce qui sera beaucoup plus rapide.



Matrice de distance

L'entrée **Matrice de distance** permet de créer un tableau indiquant la distance entre chaque paire d'éléments du jeu de données sélectionné. Pour plus d'informations sur le paramétrage : [Matrice de distances inter-taches](#).

Suppression d'un jeu de données

L'entrée **Supprimer** permet de supprimer le jeu de données sélectionné. Le projet est enregistré automatiquement après cette suppression. L'ensemble des jeux de liens, graphes et métriques calculées à partir de ce jeu de données sont automatiquement supprimés !

2.8 Affichage

2.8.1 Menu contextuel

Propriétés

Les propriétés de chaque habitat, jeu de liens, graphe sont accessibles par un clic droit sur chacun d'entre eux.

Style d'une couche

Le menu **Style** donne accès aux paramètres d'affichage des objets : couleur, épaisseur des contours, étiquettes, taille des symboles (pour les nœuds uniquement). Les objets peuvent être représentés de la même manière (symbole unique) ou en fonction d'un attribut. Il est possible d'appliquer une méthode de discrétisation pour classer les objets en fonction des valeurs de l'attribut sélectionné. Par défaut, la légende des objets est affichée dans la table des matières. Il est possible de la cacher en décochant le bouton **Légende**.

Export d'une couche

Le menu **Export** permet d'exporter les objets dans un fichier shapefile (*.shp), geopackage (*.gpkg) ou un fichier texte (*.txt).

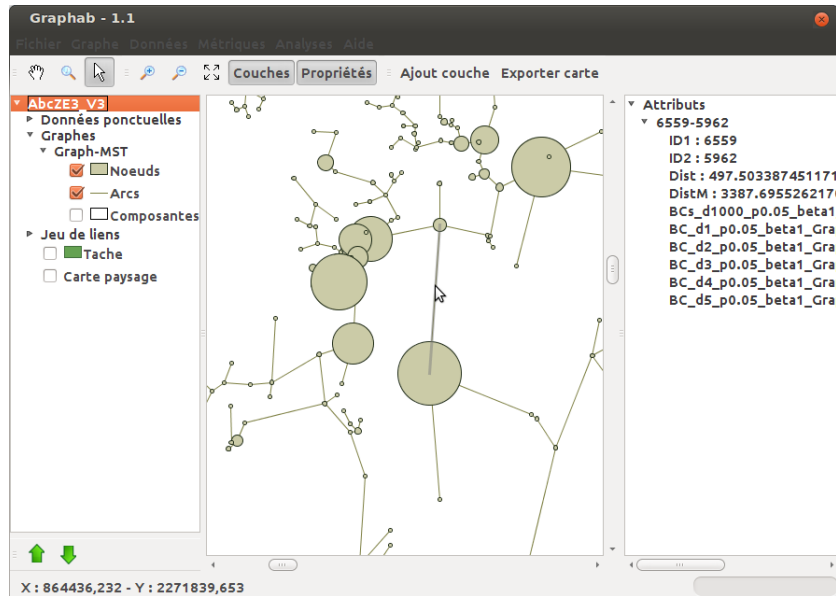
Statistiques

Le menu **Statistiques** permet d'afficher la distribution des valeurs d'un ou plusieurs attributs :

- nuage de point : affiche un nuage de point mettant en relation les valeurs de deux attributs
- histogramme : affiche l'histogramme des valeurs d'un attribut.

2.8.2 Propriétés des éléments sur la carte

Il est également possible d'afficher les valeurs d'un objet particulier en le sélectionnant à l'aide de la flèche. Après sélection, la valeur des attributs s'affiche dans une nouvelle colonne à droite. Il est possible de fermer cette colonne en cliquant sur le menu Propriétés dans la barre du haut.



Chapter 3

Interface en ligne de commande

3.1 Lancer Graphab en ligne de commande

Il faut tout d'abord ouvrir une fenêtre de terminal, puis aller dans le répertoire contenant Graphab avec la commande `cd`. Enfin, vous pouvez saisir la commande suivante pour afficher l'écran d'aide de Graphab :

```
java -jar graphab-3.0.jar --help
```

Résultat

```
Usage :  
java -jar graphab-3.0.jar --project prjfile.xml  
...  
...
```

Vous êtes prêt à utiliser Graphab en ligne de commande.

3.2 Syntaxe

3.2.1 Définition

Les commandes commencent par un double tiret : `--project`, `--linkset`, ...

Les options globales commencent par un tiret simple : `-proc`, `-nosave`, ...

Les paramètres n'ont pas de tiret : `name`, `complete`, `maxcost`, ...

3.2.2 Séparateur

Les espaces sont utilisés pour séparer les différents éléments d'une ligne de commande, par conséquent, vous ne pouvez pas avoir un nom qui contient des espaces.

Évitez de mettre des espaces dans les noms des éléments d'un projet.

3.2.3 Paramètre optionnel

Les paramètres entourés de crochets sont optionnels, et donc les paramètres qui ne sont pas entourés par des crochets sont obligatoires.

3.2.4 Intervalle et liste de valeurs

Définir un intervalle pour un paramètre (au lieu d'une valeur simple) exécutera la commande plusieurs fois pour chaque valeur définie dans l'intervalle. Un intervalle est défini par un minimum, un incrément et un maximum :

`min:inc:max`

Un intervalle de 0 à 10 avec un incrément de 2 créera 6 valeurs : 0, 2, 4, 6, 8, 10 :

`0:2:10`

La valeur minimale est toujours incluse, mais la valeur maximale est incluse seulement si la valeur incrémentée tombe exactement sur le maximum, sinon la dernière valeur sera la valeur incrémentée maximale inférieure au maximum. Un intervalle de 0 à 9 incrémenté de 2 donnera 5 valeurs : 0, 2, 4, 6, 8 :

`0:2:9`

Des valeurs décimales peuvent être utilisées, avec le point comme séparateur décimal :

`1.5:0.5:4`

Pour utiliser un ensemble de valeurs qui ne suit pas une progression arithmétique, vous pouvez utiliser une liste de valeurs séparées par des virgules :

`0,1,4,8,10`

La liste de valeurs ne doit pas contenir d'espace.

Une liste de valeurs ou une valeur simple peuvent toujours être utilisées à la place d'un intervalle. Si une commande contient plusieurs intervalles pour différents paramètres, la commande sera exécutée pour toutes les combinaisons possibles.

3.2.5 Séquençage des commandes

Graphab peut être lancé avec plusieurs commandes sur la même ligne, sauf pour les commandes `-help` et `-metrics`. Les commandes `-create` et `-project` peuvent être utilisées une seule fois et doivent être la première commande. Après une de ces deux commandes, toutes les autres commandes seront exécutées séquentiellement dans le même ordre que sur la ligne.

```
java -jar graphab-3.0.jar --project prj.xml --graph --gmetric NC
```

Cette ligne de commande charge le projet `prj.xml`, puis exécute la commande `-graph`, puis la commande `-gmetric`.

Chapter 4

Références des commandes

4.1 Commandes générales

4.1.1 `-help` : affichage de l'aide

Commande :

```
java -jar graphab-3.0.jar --help
```

Résultat :

```
Usage :
java -jar graphab.jar --metrics
java -jar graphab.jar [-proc n] --create prjname landrasterfile [nodata=val] [dir=path]
java -jar graphab.jar [-mpi | -proc n] [-nosave] [-distconv excost=val] --project prjfile.xml command1
Commands list :
--show
--show_habitats --show_linksets --show_graphs --show_metrics
--dem rasterfile
--habitat name=habitatname codes=code1,...,coden [minarea=val] [maxsize=val] [con8] [novoronoi]
--vhabitat name=habitatname file=vectorlayer|from=habitatname [mincapa=val] [capa=field] [novoronoi]
--usehabitat habitat1,...,habitatn
--removehabitat [habitat1,...,habitatn]
--mergehabitat [habitat1,...,habitatn] [novoronoi]
--capa [area [exp=value] [code1,..,coden=weight ...]] | [file=capacity.csv id=fieldname capa=...
--linkset distance=euclid|cost [name=linkname] [topo=planar|complete|planarcost] [inter] [maxcost=...
--uselinkset linkset1,...,linksetn
--removelinkset [linkset1,...,linksetn]
--graph [name=graphname] [nointra] [threshold=[{]min:inc:max[}]]
--usegraph graph1,...,graphn
--removegraph [graph1,...,graphn]
--mergegraph [name=graphname] [graph1,...,graphn]
--cluster d=val p=val [beta=val] [nb=val]
--metapatch [mincapa=value] [novoronoi]
--corridor maxcost=[{]min:inc:max[}] [format=raster|vector] [beta=exp|var=name d=val p=val]
--dataset name=dataname file=vectorlayer.gpkg
--removedataset [dataset1,...,datasetn]
--distance_matrix type=space|graph distance=euclidean|leastcost|circuit|flow|circuitflow [d=val p=val]
--gmetric global_metric_name [resfile=file.txt] [mh=all|inter|val] [param1=[{]min:inc:max[}] ...
--cmetric global_metric_name [mh=all|inter|val] [param1=[{]min:inc:max[}] [param2=...
```

```

--lmetric local_metric_name [mh=all|inter|val] [filter=condition] [param1=[{}min:inc:max[{}]] ...
--usemetric metric1,...,metricn
--removemetric [metric1,...,metricn]
--interp [var=patch_var_name d=val p=val] [name=rastername] [resol=val] [multi=dist_max ...
--model variable distW=[{}min:inc:max[{}]] [vars=var1,...,varn] [raster=r1,...,rn]
--delta [filter=condition | [obj=patch|link [sel=id1,id2,...,idn | fsel=file.txt]]
--addpatch npatch=val [hab=name] gridres=min:inc:max [capa=capa_file] | patchfile=file.gpkg ...
--remelem nstep global_metric_name [param1=val ...] obj=patch|link [sel=id1,id2,...,idn|fsel=file.txt]
--gttest nstep global_metric_name [param1=val ...] obj=patch|link sel=id1,id2,...,idn|fsel=file.txt
--gremove global_metric_name [param1=val ...] [patch=id1,id2,...,idn|fpatch=file.txt] [link=...
--landmod zone=filezones.gpkg id=fieldname code=fieldname [sel=id1,id2,...,idn]
--landmodgm zone=filezones.gpkg id=fieldname code=fieldname [sel=id1,id2,...,idn]

```

min:inc:max -> val1,val2,val3...

4.1.2 -metrics : affichage des métriques

Cette commande affiche l'ensemble des métriques disponibles avec leur abréviation, leur description et leurs paramètres.

Commande :

```
java -jar graphab-3.0.jar --metrics
```

Résultat :

```

===== Métriques globales=====
EC - Connectivité équivalente (EC)
params : [d, p]
PC - Probabilité de connectivité (PC)
params : [d, p]
IIC - Indice intégral de connectivité (IIC)
S#F - Sum Flux (F)
params : [d, p, beta]
CCP - Probabilité de coïncidence (CCP)
MSC - Taille moyenne des composantes (MSC)
SLC - Taille de la plus grande composante (SLC)
ECS - Taille de cluster attendue (ECS)
GD - Diamètre (GD)
H - Harary indice (H)
NC - Nombre de composantes (NC)
dPC - Delta PC décomposé (dPC)
params : [d, p]
W - Lambda de Wilks (W)
params : [attrs, npatch, warea]

===== Métriques locales =====
IF : Flux d'interaction (IF)
params : [d, p, beta]
F : Flux (F)
params : [d, p, beta]
BC : Centralité intermédiaire (BC)
params : [d, p, beta]
CF : Flux circuit (CF)
params : [beta]
Dg : Degré du noeud (Dg)

```

CC : Coefficient d'agrégation (CC)
CCe : Centralité de proximité (CCe)
CCor : Corrélation de Connectivité (CCor)
Ec : Excentricité (Ec)

4.2 Gestion du projet

4.2.1 `--create` : création d'un projet

```
java -jar graphab-3.0.jar --create prjname landrasterfile [nodata=val] [dir=path]
```

Paramètres obligatoires

- `prjname` : nom du nouveau projet
- `landrasterfile` : nom du fichier image contenant la carte du paysage (ou occupation du sol) au format Tiff ou Ascii grid

Paramètres optionnels

- `nodata=val` : valeur représentant l'absence de données dans l'image raster
- `dir=path` : chemin où enregistrer le projet, par défaut le projet est enregistré dans le répertoire courant.

Description

Cette commande permet de créer un nouveau projet et de le charger.

La commande `--create` ne peut être utilisée qu'une seule fois et doit être la première commande de la ligne. Les commandes suivantes seront exécutées sur le projet nouvellement créé.

4.2.2 `--project` : chargement d'un projet

Cette commande définit le chemin vers le fichier xml du projet à charger.

```
java -jar graphab-3.0.jar --project path2myproject/myproject.xml
```

Charge le projet *myproject* contenu dans le répertoire *path2myproject*.

La commande `--project` ne peut être utilisée qu'une seule fois et doit être la première commande de la ligne. Toutes les commandes qui suivent dans ce manuel ont besoin d'un projet chargé.

4.2.3 `--show` : affichage des éléments d'un projet

Affiche l'ensemble des habitats, jeux de liens, graphes, métriques et jeux de données contenu dans le projet chargé. Cette commande est utile pour récupérer les noms exacts des éléments pour les utiliser par la suite dans une ligne de commande.

Commande :

```
java -jar graphab-3.0.jar --project path2myproject/myproject.xml --show
```

Résultat :

```
==== Habitats ====  
2 - dataset1  
0 - habitat1  
1 - habitat4
```

```

===== Jeux de liens =====
jl1_cost
jl_h1_h4
jl_inter

===== Graphes =====
g_h1_cost
g_h1_h4
g_h1_inter

===== Métriques =====
ec_d2000_p0.5_g_h1_cost
bc_d1000_p0.5_beta1_g_h1_h4
delta_inter_g_h1_h4
f_d100_p0.5_beta1_g_h1_h4
ec_d1000_p0.5_g_h1_inter

```

Attention aux noms des éléments : la ligne de commande est sensible à la casse (différentiation minuscule/majuscule) et ne gère pas les espaces dans les noms.

4.2.4 `-dem` : chargement d'un MNT

Cette commande définit le chemin vers un MNT (Modèle Numérique de Terrain) qui pourra être utilisé dans la création des liens pour tenir compte de la pente.

```
--dem rasterfile
```

Paramètre obligatoire

- `rasterfile` : fichier raster au format Tiff ou AsciiGrid représentant les altitudes. La géométrie du raster doit correspondre exactement à la carte du paysage.

4.3 Habitat

4.3.1 `-habitat` : création d'un habitat à partir de la carte de paysage

```
--habitat name=habitatname codes=code1,...,coden [minarea=val]
[maxsize=val] [con8] [novoronoi]
```

Paramètres obligatoires

- `habitatname` : nom du nouvel habitat
- `codes=code1,...,coden` : le ou les codes d'occupation du sol représentant les taches d'habitat.

Paramètres optionnels

- `minarea=val` : taille minimale d'une tache d'habitat en hectare
- `maxsize=val` : découpe les taches dont la longueur ou largeur en mètre dépasse `maxsize` sur une grille de taille `maxsize`
- `con8` : connexité aux 8 pixels voisins pour la définition des taches, par défaut la connexité est seulement de 4

- **novoronoi** : ne calcule pas le diagramme de voronoi pour la topologie planaire. Cette option est utile pour gagner du temps si la topologie planaire n'est pas utilisée à la création du jeu de liens.

Description

Cette commande permet de créer un nouvel habitat.

Les commandes suivantes seront exécutées sur l'habitat nouvellement créé.

4.3.2 **-vhabitat** : création d'un habitat à partir d'une couche vecteur

```
--vhabitat name=habitatname file=vectorlayer|from=habitatname [mincapa=val]
[capa=field] [novoronoi]
```

Paramètres obligatoires

- **habitatname** : nom du nouvel habitat
- **file=vectorlayer|from=habitatname** : la source de données de cette habitat. A partir d'une couche vecteur au format `.gpkg` ou `.shp` (**file**) ou bien d'un habitat existant du projet (**from**). Cette seconde option est utile pour filtrer les taches avec une capacité trop faible avec le paramètre **mincapa**

Paramètres optionnels

- **mincapa=val** : capacité minimale d'une tache pour qu'elle soit conservée
- **capa=field** : le nom du champ de la couche vecteur contenant la capacité. Utile uniquement pour le paramètre **file**
- **novoronoi** : ne calcule pas le diagramme de voronoi pour la topologie planaire. Cette option est utile pour gagner du temps si la topologie planaire n'est pas utilisée à la création du jeu de liens.

Description

Cette commande permet de créer un nouvel habitat à partir d'une couche vectorielle indépendamment de la carte de paysage. L'étendue de cette couche doit être comprise dans l'étendue de la carte de paysage.

Cette commande permet aussi de créer un habitat à partir d'un habitat existant pour filtrer les taches ayant une capacité trop faible avec le paramètre **from** utilisé conjointement avec le paramètre **mincapa**.

Les commandes suivantes seront exécutées sur l'habitat nouvellement créé.

4.3.3 **-usehabitat** : sélection des habitats

```
--usehabitat ALL|habitat1,...,habitatn
```

Sélectionne les habitats qui seront utilisés dans les commandes suivantes. Les noms d'habitat sont sensibles à la casse et ne peuvent pas contenir d'espace. Le mot clé **ALL** peut être utilisé pour sélectionner tous les habitats.

Par défaut, tous les habitats d'un projet sont sélectionnés.

4.3.4 **-removehabitat** : suppression d'habitat

```
--removehabitat ALL|habitat1,...,habitatn
```

Supprime les habitats sélectionnés. Tous les jeux de liens, les graphes ainsi que les métriques dépendant de ces habitats seront aussi supprimés. Le mot clé **ALL** peut être utilisé pour supprimer tous les habitats.

L'option **-nosave** est sans effet sur cette commande.

4.3.5 `--mergehabitat` : fusion d'habitat

```
--mergehabitat [ALL|habitat1,...,habitatn] [novoronoi]
```

Paramètres optionnels

- `habitat1,...,habitatn` : liste des habitats à fusionner. Si cette liste n'est pas donnée, les habitats actuellement sélectionnés seront fusionnés.
- `novoronoi` : ne calcule pas le diagramme de voronoi pour la topologie planaire. Cette option est utile pour gagner du temps si la topologie planaire n'est pas utilisée à la création du jeu de liens.

Fusionne les habitats sélectionnés en un seul habitat. Cette commande est utile pour calculer un jeu de liens reliant l'ensemble de ces habitats.

Les commandes suivantes seront exécutées sur l'habitat fusionné.

4.3.6 `--capa` : capacité des taches

```
--capa [area [exp=value] [code1,...,coden=weight ...]]  
| [file=capacity.csv id=fieldname capa=fieldname]  
| [maxcost=[{valcost}] codes=code1,code2,...,coden [weight]]
```

Paramètres définition de surface

- `area`: définit la capacité comme la superficie de la tache (défaut)
- `exp=value`: modifie la surface des taches par un exposant.
- `code1,...,coden=weight ...`: pondération de la superficie de la tache par les codes d'occupation du sol la composant. Ce paramètre est utile seulement quand les taches sont définies par plusieurs codes d'occupation du sol.

Paramètres fichier externe

- `file=capacity.csv`: définit la capacité à partir du fichier csv donné
- `id=fieldname`: nom de la colonne du fichier contenant l'identifiant de la tache
- `capa=fieldname`: nom de la colonne du fichier contenant la nouvelle valeur de capacité.

Paramètres définition de surface de voisinage

- `maxcost=[{valcost}]` : coût cumulé maximum pour le voisinage
- `codes=code1,code2,...,coden` : codes d'occupation du sol à prendre en compte dans la surface de voisinage
- `weight` : introduit une pondération en fonction de l'éloignement à la tache, par le biais d'une fonction exponentielle négative. De cette façon, les surfaces sélectionnées compteront plus si elles sont proches de la tache et inversement.

Description

La commande `--capa` calcule la capacité des taches des habitats sélectionnés et enregistre le projet sauf si l'option `-nosave` a été spécifiée. Sans paramètre ou avec le paramètre `area`, la capacité est définie comme la surface des taches en m^2 .

Avec le paramètre `file=`, la capacité est définie à partir d'un fichier externe au format csv.

Avec le paramètre `maxcost`, la capacité est calculée comme une surface d'éléments autour de la tache, jusqu'à la distance `valcost`. Les distances sont calculées à partir des coûts définis dans le jeu de lien sélectionné. Si c'est un jeu de lien euclidien, les coûts sont mis à la résolution pour l'ensemble des classes d'occupation du sol. Cette commande supporte la sélection d'un seul jeu de liens (cf. [-uselinkset : sélection des jeux de liens](#)).

Exemples

```
--capa
--capa area
```

Les 2 exemples précédents calculent la capacité comme la surface de la tache (en m^2).

```
--capa area 1=0.5 5=2
```

Définit la capacité de la tache comme la surface de la tache (en m^2) pondérée par 0.5 pour le code d'occupation du sol 1 et pondérée par 2 pour le code d'occupation du sol 5.

```
--capa maxcost=100 codes=1,3
```

Définit la capacité de chaque tache comme la surface (en pixel) d'éléments de code d'occupation du sol 1 et 3 présent autour de la tache jusqu'à une distance de 100 (en coût).

4.3.7 `--metapatch` : création d'un habitat méta-tache

```
--metapatch [mincapa=value] [novoronoi]
```

Paramètre optionnel

- `mincapa=value` : capacité minimale d'une méta-tache pour être conservée dans le nouvel habitat
- `novoronoi` : ne calcule pas le diagramme de voronoi pour la topologie planaire. Cette option est utile pour gagner du temps si la topologie planaire n'est pas utilisée à la création du jeu de liens.

Description

La commande `--metapatch` crée un nouvel habitat de méta taches pour chaque graphe sélectionné. Chaque composante du graphe sélectionné correspondra à une méta-tache. Celle-ci est donc constituée de l'ensemble des taches d'habitat connectées entre elles. La capacité des méta-taches est définie comme la somme de la capacité des taches composant la méta-tache.

Ce nouvel habitat devient l'habitat courant pour les commandes suivantes.

Exemple

```
--usegraph 2000m_euclid --metapatch mincapa=1000
```

Cette commande créera un nouvel habitat basé sur le graphe `2000m_euclid` et supprimera toutes les méta-taches dont la capacité est inférieure à 1000.

Références

[[Clauzel et al.\(2015b\)](#)], [[Foltête et al.\(2016\)](#)]

4.4 Jeu de liens

4.4.1 `--linkset` : création d'un jeu de liens

```
--linkset distance=euclid|cost [name=linkname] [topo=planar|complete|planarcost] [inter]
[maxcost=valcost] [slope=coef] [remcrosspath|nopathsaved] [[code1,..,coden=cost1 ...]
codei,..,codej=min:inc:max | extcost=rasterfile]
```

Paramètre obligatoire

- `distance=euclid|cost` : type de distance pour les liens, euclidien (`distance=euclid`) ou coût cumulé (`distance=cost`). Si `distance=cost`, il faut définir les coûts par classe d'occupation du sol ou un raster contenant les coûts.

Paramètres optionnels généraux

- `name=linkname` : nom du jeu de liens qui sera créé
- `topo=planar|complete` : définit la topologie du jeu de liens. La topologie par défaut est planaire (`planar`)
- `inter` : crée un jeu de liens inter-habitat ie. ne conserve que les liens entre des habitats différents
- `maxcost=valcost` : le paramètre `maxcost` permet de limiter le calcul des liens à une distance maximale.

Paramètres spécifiques au cas `distance=cost`

- `topo=planarcost` : cette option permet de déterminer la topologie planaire à partir des distances coûts au lieu de la distance euclidienne.
- `slope=coef` : pondère les coûts par la pente. Pour utiliser cette option, il faut que le projet contienne un MNT.
- `remcrosspath` : supprime les liens traversant une tache
- `nopathsaved` : n'enregistre pas la géométrie des chemins. Utile pour réduire l'utilisation mémoire en topologie complète
- `[code1,..,coden=cost1 ...] codei,..,codej=min:inc:max` : définition des coûts à partir des codes de l'occupation du sol
- `extcost=rasterfile` : définition des coûts à partir d'un raster externe au format tiff, ascii grid ou rst.

Description

Crée un nouveau jeu de liens pour chaque habitat sélectionné et enregistre le projet sauf si l'option globale `-nosave` est utilisée.

Si le paramètre `name` n'est pas défini, le nom du jeu de lien sera déterminé par la définition des coûts.

La définition des coûts ne supporte pas plusieurs intervalles ou liste de valeurs.

Après la commande `--linkset`, les jeux de liens sélectionnés correspondent aux jeux de liens créés.

Exemples

Cette commande créera un jeu de lien en topologie planaire nommé `cost_1_2_3_4_5_6_7-1.0` avec tous les coûts à 1 :

```
--linkset distance=cost 1,2,3,4,5,6,7=1
```

Cette commande créera un jeu de lien *cost_1_2_3-1.0* avec un coût égal à 1 pour les valeurs d'occupation du sol 1, 2 et 3 et un coût égal à 2 pour les valeurs d'occupation du sol 4, 5, 6 et 7 :

```
--linkset distance=cost 1,2,3=1 4,5,6,7=2
```

Par défaut la topologie est planaire ; pour créer une topologie complète, utilisez l'option **complete** :

```
--linkset distance=cost topo=complete 1,2,3,4,5,6,7=1
```

On peut préciser un seuil de distance pour éviter de créer trop de liens. Exemple avec un seuil max à 100 :

```
--linkset distance=cost topo=complete maxcost=100 1,2,3,4,5,6,7=1
```

On peut créer plusieurs jeux de liens avec des coûts différents en utilisant un intervalle ou une liste de valeurs :

```
--linkset distance=cost 4,5,6,7=10 1,2,3=100:50:200
```

ou

```
--linkset distance=cost 4,5,6,7=10 1,2,3=100,150,200
```

Résultat : la commande ci-dessus créera 3 jeux de liens *cost_1_2_3-100.0 cost_1_2_3-150.0 cost_1_2_3-200.0* avec les valeurs d'occupation du sol 1, 2 et 3 égales à 100, 150 ou 200.

4.4.2 **-uselinkset** : sélection des jeux de liens

```
--uselinkset ALL|[linkset1,..,linksetn]
```

Sélectionne les jeux de liens qui seront utilisés dans les commandes suivantes. Les noms de jeu de liens sont sensibles à la casse et ne peuvent pas contenir d'espace.

Par défaut, tous les jeux de liens sont sélectionnés.

4.4.3 **-removelinkset** : suppression de jeux de liens

```
--removelinkset ALL|[linkset1,..,linksetn]
```

Supprime les jeux de liens sélectionnés. Tous les graphes et métriques dépendant de ces jeux de liens sont aussi supprimés.

L'option **-nosave** est sans effet sur cette commande.

4.5 Graphe

4.5.1 **-graph** : création d'un graphe

```
--graph [name=graphname] [nointra] [threshold=[]min:inc:max[]]
```

Paramètres optionnels

- **name=graphname** : nom du graphe qui sera créé. Ce paramètre peut être utilisé uniquement si la commande ne crée qu'un seul graphe.
- **nointra** : désactive les distances intra-tache pour le calcul des métriques
- **threshold=[]min:inc:max[]** : définit la distance maximale des liens inclus dans chaque graphe. Sans ce paramètre, le graphe contient tous les liens du jeu de liens. Entouré d'accolades, les valeurs de distance sont converties en valeur de coût automatiquement.

Description

Crée un graphe pour chaque jeu de liens sélectionné et enregistre le projet sauf si l'option globale `-nosave` est utilisée.

Après la commande `--graph`, les graphes sélectionnés correspondent aux graphes venant d'être créés.

Exemples

Sans paramètre, la commande créera un graphe sans élagage pour chaque jeu de liens sélectionné :

```
--graph
```

Le nom du graphe sera la concaténation de `comp_` et du nom du jeu de lien.

Le paramètre `threshold` permet de spécifier un seuil (100 dans l'exemple) pour le graphe à créer :

```
--graph threshold=100
```

Le nom du graphe sera la concaténation de `thresh_100.0_` et du nom du jeu de lien.

Si le paramètre `threshold` est défini avec un intervalle ou une liste de valeurs, un graphe élagué pour chaque valeur de seuil et chaque jeu de liens sera créé :

```
--graph threshold=1000:100:1500
```

ou

```
--graph threshold=1000,1100,1200,1300,1400,1500
```

Résultat : cette commande créera 6 graphes élagués (1000, 1100, 1200, 1300, 1400, 1500) pour chaque jeu de liens sélectionné.

4.5.2 `--usegraph` : sélection des graphes

```
--usegraph ALL|graph1,...,graphn
```

Sélectionne les graphes qui seront utilisés dans les commandes suivantes. Les noms de graphe sont sensibles à la casse et ne peuvent pas contenir d'espace.

Par défaut, tous les graphes sont sélectionnés.

4.5.3 `--removegraph` : suppression de graphes

```
--removegraph ALL|[graph1,...,graphn]
```

Supprime les graphes sélectionnés. Les métriques dépendant de ces graphes seront aussi supprimées.

L'option `-nosave` est sans effet sur cette commande.

4.5.4 `--mergegraph` : fusion de graphes

```
--mergegraph [name=graphname] ALL|[graph1,...,graphn]
```

Paramètres optionnels

- `name=graphname` : nom du graphe qui sera créé.
- `ALL|graph1,...,graphn` : liste des graphes à fusionner ou le mot clé `ALL` pour fusionner tous les graphes.

Fusionne les graphes sélectionnés en un seul graphe. La sélection peut être faite de 2 manières : soit une sélection précédent cette commande, soit par une liste de graphes donnée à cette commande

Les commandes suivantes seront exécutées sur le graphe nouvellement créé.

Exemples

Sans paramètre, la commande créera un graphe fusionnant les graphes sélectionnés précédemment

```
--mergegraph
```

La commande ci-dessous crée un graphe nommé gm fusionnant les graphes g1 et g2

```
--mergegraph name=gm g1,g2
```

4.5.5 --corridor : calcul de corridors

```
--corridor maxcost=[{min:inc:max}] [format=raster|vector] [beta=exp|var=name d=[{val}] p=v
```

Paramètre obligatoire

- `maxcost=[{min:inc:max}]` : coût cumulé maximum des chemins formant un corridor. Entouré d'accolades, la distance donnée en mètre est convertie en unité de coût automatiquement.

Paramètre optionnel

- `format=raster|vector` : format en sortie raster (.tif) ou vectoriel (.gpkg). Le format par défaut est vectoriel.

Paramètres optionnels pour le format raster

Si le paramètre `beta` ou `var` est défini le pixel ne correspond plus au nombre de corridors mais à la valeur maximale provenant des corridors. Au niveau de chaque corridor un gradient décroissant est calculé depuis les chemins les plus courts. Ce gradient prend la valeur 1 au niveau des chemins les plus courts et décroît en suivant la loi exponentielle inverse : $e^{-\alpha d}$. Le résultat de ce gradient est multiplié par le produit des capacités des taches $(a_i a_j)^\beta$ si le paramètre `beta` est défini ou par une métrique locale si le paramètre `var` est défini. Les paramètres `beta` et `var` sont exclusifs et les paramètres `d` et `p` doivent être données pour définir l'exposant α de décroissance de la distance.

- `beta=exp` : exposant du produit des capacités des taches $(a_i a_j)^\beta$. Les valeurs habituelles sont 0 : pas d'effet des capacités, 0.5 : moyenne géométrique des capacités ou 1 : produit des capacités.
- `var=name` : `name` correspond à une métrique locale calculée sur les liens
- `[{d=val}]` : distance pour définir l'exposant α de décroissance de la distance
- `p=val` : probabilité pour définir l'exposant α de décroissance de la distance

Description

La commande `--corridor` calcule le corridor associé à chaque lien d'un graphe. Le résultat est stocké dans une couche vecteur (ou un raster au format tif) différent pour chaque graphe sélectionné. Chaque couche contient, pour chaque lien du graphe, un polygone représentant l'ensemble des chemins possibles reliant les deux taches et ayant une distance en coût cumulé inférieure ou égale à `valcost`. Au format raster, par défaut, chaque pixel dénombre le nombre de corridors passant par ce pixel. Si le paramètre `beta` ou `var` est défini le calcul est différent (cf. [Sortie raster](#)).

Exemple

```
--corridor maxcost=500
```

4.5.6 --cluster : partitionnement de graphe

```
--cluster d=val p=val [beta=val] [nb=val]
```

Paramètres obligatoires

- **d=val** : distance permettant de définir l'exposant α .
- **p=val** : probabilité permettant de définir l'exposant α . Pour définir α à 0 et ne pas tenir compte de l'impédance des liens, mettre **p** à 1.

Paramètres optionnels

- **beta=val** : exposant des capacités, par défaut à 1. Pour ne pas tenir compte de la capacité des tâches, mettre **beta** à 0.
- **nb=val** : nombre de compartiments, par défaut le nombre de compartiments sélectionné correspond à la modularité maximale.

Description

Crée un graphe basé sur son partitionnement qui maximise la modularité [Newman(2006)] pour chaque graphe sélectionné et enregistre le projet sauf si l'option globale **-nosave** est utilisée. Les nouveaux graphes conservent uniquement les liens intra-groupes.

La modularité est calculée à partir d'un poids (w_{ij}) défini pour chaque lien du graphe :

$$w_{ij} = (a_i a_j)^\beta e^{-\alpha d_{ij}}$$

Les 2 paramètres β et α permettent de définir respectivement l'importance de la capacité des tâches ($a_i a_j$) et l'importance de la distance (d_{ij}) pour le poids du lien w_{ij} . Si $\alpha = \beta = 0$ alors les poids sont tous identiques : $w_{ij} = 1$. Pour plus de détails voir [Partitionnement de graphe](#).

Après la commande **--cluster**, les graphes sélectionnés correspondent aux graphes venant d'être créés.

Référence

[Foltête and Vuidel(2017)]

4.6 Métrique

4.6.1 **-gmetric** : calcul d'une métrique globale

```
--gmetric global_metric_name [resfile=file.txt] [mh=all|inter|val]
[param1=min:inc:max [param2=min:inc:max ...]]
```

Paramètre obligatoire

- **global_metric_name** : acronyme de la métrique globale à calculer (EC, PC, ...)

Paramètres optionnels

- **resfile=file.txt** : fichier contenant le résultat du calcul de la métrique
- **mh=all|inter|val** : décomposition de la métrique en fonction des habitats : **all** crée autant de résultats que de combinaison d'habitats possible, **inter** crée un seul résultat entre habitats différents, **val** spécifie la combinaison d'habitat voulu (ex: 0/0)
- **param1=[{]min:inc:max[}]** : paramètre(s) de la métrique, si elle en a. Entouré d'accolades, les valeurs de distance sont converties en valeur de coût automatiquement.
- ...

Description

Calcule la métrique globale données sur chaque graphe sélectionné. Le nom de la métrique est son nom court comme affiché par la commande `--metrics`. Si la métrique a besoin de paramètres, ils peuvent être spécifiés dans n'importe quel ordre.

Les résultats sont enregistrés dans un fichier texte dans le répertoire du projet. Par défaut, le nom du fichier correspond au nom court de la métrique. Un autre nom peut être donné par le paramètre `resfile`.

Exemples

Calcule la métrique NC sur tous les graphes sélectionnés :

```
--gmetric NC
```

Le résultat est enregistré dans le fichier NC.txt dans le répertoire du projet :

Graph	NC
2000m-3500cost	25.0
2000m-3500cost_comp	24.0
2000m_euclid	9.0
2000m_euclid_comp	9.0

Pour les métriques ayant des paramètres, on peut tester plusieurs valeurs de paramètre en une seule commande. La commande suivante calcule la métrique PC avec 6 jeux de paramètres différents pour chaque graphe sélectionné :

```
--gmetric PC d=1000:500:2000 p=0.05,0.5
```

Le résultat est stocké dans le fichier PC.txt :

Graph	d	p	PC
2000m-3500cost	1000.0	0.05	2.108166945899072E-15
2000m-3500cost	1500.0	0.05	2.4839095790785042E-15
2000m-3500cost	2000.0	0.05	2.866220685546806E-15
2000m-3500cost	1000.0	0.5	1.317091007462398E-6
2000m-3500cost	1500.0	0.5	1.4758311225154786E-6
2000m-3500cost	2000.0	0.5	1.5884005111333579E-6
2000m-3500cost_comp	1000.0	0.05	2.1106833149811817E-15
2000m-3500cost_comp	1500.0	0.05	2.493027588606987E-15
2000m-3500cost_comp	2000.0	0.05	2.887027144684246E-15
2000m-3500cost_comp	1000.0	0.5	1.3171878007185563E-6
2000m-3500cost_comp	1500.0	0.5	1.476224502635306E-6
2000m-3500cost_comp	2000.0	0.5	1.5892024206564504E-6
2000m_euclid	1000.0	0.05	2.8238137481213476E-15
2000m_euclid	1500.0	0.05	3.516516320195261E-15
2000m_euclid	2000.0	0.05	4.285009196943927E-15
2000m_euclid	1000.0	0.5	1.7079340030649911E-6
2000m_euclid	1500.0	0.5	1.8176869551880345E-6
2000m_euclid	2000.0	0.5	1.8976284261240914E-6
2000m_euclid_comp	1000.0	0.05	2.867215798466552E-15
2000m_euclid_comp	1500.0	0.05	3.581685718161269E-15
2000m_euclid_comp	2000.0	0.05	4.374805768414666E-15
2000m_euclid_comp	1000.0	0.5	1.7172345329380555E-6
2000m_euclid_comp	1500.0	0.5	1.8277346595840518E-6
2000m_euclid_comp	2000.0	0.5	1.9080569002930505E-6

4.6.2 `--cmetric` : calcul d'une métrique par composante

```
--cmetric global_metric_name [mh=all|inter|val]  
[param1=min:inc:max [param2=min:inc:max ...]]
```

Paramètre obligatoire

- `global_metric_name` : acronyme de la métrique globale à calculer (PC, IIC, ...)

Paramètres optionnels

- `mh=all|inter|val` : décomposition de la métrique en fonction des habitats : `all` crée autant de résultats que de combinaison d'habitats possible, `inter` crée un seul résultat entre habitats différents, `val` spécifie la combinaison d'habitat voulu (ex: 0/0)
- `param1=[{[]min:inc:max[]}]` : paramètre(s) de la métrique, si elle en a. Entouré d'accolades, les valeurs de distance sont converties en valeur de coût automatiquement.
- ...

Description

Calcule la métrique globale donnée sur chaque composante de chaque graphe sélectionné. Le nom de la métrique est son nom court comme affiché par la commande `--metrics`. Le résultat est enregistré dans un attribut des composantes de chaque graphe. Le projet est enregistré sauf si l'option globale `-nosave` est utilisée.

Exemple

Avec une métrique ayant des paramètres, on peut tester plusieurs jeux de paramètres en une commande :

```
--cmetric PC d=1000:500:2000 p=0.05,0.5
```

6 PC seront calculés pour chaque composante de chaque graphe sélectionné :

- `PC_d1000_p0.05`
- `PC_d1500_p0.05`
- `PC_d2000_p0.05`
- `PC_d1000_p0.5`
- `PC_d1500_p0.5`
- `PC_d2000_p0.5`

4.6.3 `--lmetric` : calcul d'une métrique locale

```
--lmetric local_metric_name [name=resname] [mh=all|inter|val] [filter=condition]  
[param1=min:inc:max [param2=min:inc:max ...]]
```

Paramètre obligatoire

- `local_metric_name` : acronyme de la métrique locale à calculer (F, IF, BC, CF, ...)

Paramètres optionnels

- **name=resname**: définit le nom du résultat du calcul de la métrique ie. nom des attributs
- **mh=all|inter|val** : décomposition de la métrique en fonction des habitats : **all** crée autant de résultats que d'habitats, **inter** crée un seul résultat entre habitats différents, **val** spécifie l'habitat voulu par son identifiant
- **filter=condition** : permet de déterminer pour quels éléments du graphe un calcul sera réalisé et stocké cf. [Paramètre filter=condition](#)
- **param1=[{]min:inc:max[}]** : paramètre(s) de la métrique, si elle en a. Entouré d'accolades, les valeurs de distance sont converties en valeur de coût automatiquement.
- ...

Description

Calcule la métrique locale donnée sur chaque élément (noeud et/ou lien) de chaque graphe sélectionné. Le nom de la métrique est son nom court comme affiché par la commande `--metrics`. Le résultat est enregistré dans un attribut des taches et/ou des liens. Le projet est enregistré sauf si l'option globale `-nosave` est utilisée.

Exemple

Avec une métrique ayant des paramètres, on peut tester plusieurs jeux de paramètres en une commande :

```
--lmetric F d=1000:500:2000 p=0.05 beta=0,1
```

La métrique F sera calculée avec 6 jeux de paramètres différents pour chaque élément de chaque graphe sélectionné :

- F_d1000_p0.05_beta0
- F_d1500_p0.05_beta0
- F_d2000_p0.05_beta0
- F_d1000_p0.05_beta1
- F_d1500_p0.05_beta1
- F_d2000_p0.05_beta1

4.6.4 Paramétrage des métriques

Paramètre filter=condition

Les commandes `--lmetric` et `--delta` ont un paramètre permettant de sélectionner les éléments sur lesquels le calcul sera réalisé.

Appliquer le calcul uniquement sur les noeuds du graphe :

```
filter=node
```

Appliquer le calcul uniquement sur les noeuds du graphe et dont l'identifiant d'habitat est 1 :

```
"filter=node && idhab=1"
```

A noter que si vous mettez des espaces dans la condition il faut entourer l'ensemble de double cote.

Appliquer le calcul uniquement sur les noeuds du graphe qui ont une surface supérieure à $1000m^2$:

```
"filter=node && area > 1000"
```


Appliquer le calcul uniquement sur les liens du graphe :

```
filter=edge
```

Si la métrique contient les paramètres `d` et `p` alors 2 autres paramètres optionnels sont utilisables : `maxd` ou `minp`

Paramètre `maxd=valcost`

Le paramètre `maxd` limite le calcul de chemin à `valcost` (ie. les chemins supérieurs à `valcost` ne seront pas calculés). Ce paramètre peut réduire significativement le temps de calcul d'une métrique basée sur le calcul de chemins, mais le résultat sera moins précis.

Paramètre `minp=valproba`

Le paramètre `minp` limite le calcul de chemin à une probabilité supérieure à `valproba`. Ce paramètre peut réduire significativement le temps de calcul d'une métrique basée sur le calcul de chemins, mais le résultat sera moins précis. Ce paramètre est équivalent au paramètre `maxd`, à la différence qu'il s'applique à la probabilité au lieu de la distance. Ce paramètre est utile quand on fait varier le paramètre distance.

4.6.5 `--interp` : interpolation de métriques

```
--interp [var=patch_var_name d=val p=val] [name=rastername] [resol=val]
[multi[=dist_max] [ag=sum|max|avg]]
```

Paramètres optionnels

- `var=patch_var_name` : nom de la variable à interpoler. Ce nom doit correspondre à un attribut des taches ; habituellement le résultat du calcul d'une métrique locale. Le nom court de la métrique sans les paramètres peut être utilisé.
- `d=val` : distance permettant, avec le paramètre `p`, de définir le coefficient α de décroissance de la distance
- `p=val` : probabilité permettant, avec le paramètre `d`, de définir le coefficient α de décroissance de la distance
- `name=rastername` : nom du fichier raster stockant le résultat de l'interpolation
- `resol=val` : résolution de l'interpolation. Dans certains cas, ce paramètre est ignoré et la résolution de la carte d'occupation du sol est utilisée à la place.
- `multi[=dist_max]` : interpolation à partir de toutes les taches à une distance inférieure ou égale à `dist_max`, au lieu de la tache la plus proche. Le paramètre `dist_max` est optionnel si la commande s'exécute sur une métrique pondérée sélectionnée.
- `ag=sum|max|avg` : agrégation par une somme (`sum`), le maximum (`max`) ou bien une moyenne pondérée (`avg`). Utilisé uniquement avec le paramètre `multi`.

Description

La commande `--interp` permet d'interpoler sur l'ensemble de la zone une donnée disponible au niveau des taches, habituellement une métrique locale. L'interpolation est basée sur une fonction décroissante de la distance. La valeur interpolée à un point p_j de l'espace sera :

$$p_j = var_i * e^{-\alpha d_{ij}}$$

var_i représente la valeur de la variable pour la tache la plus proche du point p_j . Le coefficient α est défini par les 2 paramètres `d` et `p`, tel que $p = e^{-\alpha d}$. Le calcul de la distance entre le point et la tache la plus proche d_{ij} dépend du jeu de liens utilisé pour le calcul de la métrique.

Si le paramètre `multi` est activé, l'interpolation ne se calcule plus uniquement avec la tache la plus proche, mais aussi avec toutes les taches dont la distance est inférieure ou égale à `dist_max`. Ces différentes valeurs seront agrégées, par défaut, par la somme. Si le paramètre `ag=avg` est ajouté, l'agrégation utilisée sera une moyenne pondérée. Dans le cas de la somme la formule d'interpolation d'un point p_j devient :

$$p_j = \sum_i var_i * e^{-\alpha d_{ij}}$$

Exemple

La commande `--interp` peut être utilisée sans paramètre si une ou plusieurs métriques locales sont calculées ou sélectionnées précédemment

```
--lmetric F d=1000 p=0.5 beta=1 --interp multi=5000 ag=sum
```

Le paramètre α normalement défini par les paramètres `d` et `p` est défini à partir du paramétrage de la métrique. Même exemple en donnant le nom de la métrique :

```
--interp var=f_d1000_p0.5_beta1_2000m_euclid d=1000 p=0.5 multi=5000 ag=sum
```

Même commande que précédemment, le nom de l'attribut a été raccourci pour plus de simplicité. Dans ce cas toutes les métriques `F` calculées seront interpolées.

```
--interp var=f_ d=1000 p=0.5 name=test multi=5000 ag=sum
```

4.7 Jeu de données et modèle

Un jeu de donnée est un habitat vectoriel dont les capacités sont nulles. Cette abstraction permet de calculer des jeux de liens, des graphes et des métriques incluant un jeu de donnée en plus des habitats habituels. De ce fait, dans la plupart des commandes un habitat peut être remplacé par un jeu de données.

4.7.1 `--dataset` : import de jeu de données

```
--dataset name=dataname file=vectorlayer.gpkg
```

Importe un nouveau jeu de données à partir d'une couche vecteur donnée et enregistre le projet sauf si l'option globale `-nosave` est utilisée.

Paramètres obligatoires

- `name=dataname` : définit le nom du nouveau jeu de données
- `file=vectorlayer` : fichier contenant le couche vecteur

Après la commande `--dataset`, les jeux de données sélectionnés correspondent à celui nouvellement créé.

4.7.2 `--distance_matrix` : matrice de distances d'un jeu de données ou habitat

```
--distance_matrix type=space|graph distance=leastcost|circuit|flow|circuitflow [dist=val proba=
```

Paramètres obligatoires

- `type`: le calcul de distance est basé sur l'espace continu `space` (euclidien ou raster) ou sur le graphe `graph`
- `distance`: à partir d'un raster, la distance peut être moindre coût (`leastcost`) ou `circuit`, à partir d'un graphe, la distance peut être `leastcost`, `circuit`, `flow` or `circuitflow`

Paramètres optionnels

Pour les distances `flow` et `circuitflow` les paramètres suivants doivent être renseignés :

- `dist` : la distance entre 2 tâches
- `proba` : la probabilité de mouvement entre 2 tâches pour la distance `dist`

Description

Cette commande calcule la matrice de distances pour chaque jeu de données (ou habitat) sélectionnés et enregistre chaque matrice dans un fichier texte commençant par `distance_`.

Pour un calcul basé sur un raster de coût, une matrice de distance sera calculé pour chaque jeu de données (ou habitat) et chaque jeu de liens sélectionnés, pour un calcul basé sur un graphe, une matrice de distance sera calculée pour chaque jeu de données (ou habitat) et chaque graphe sélectionné.

Exemple

Dans l'exemple ci-dessous, les distances de moindre coût entre les points sera calculées à partir des raster de coûts de chaque jeu de liens sélectionné et ce, pour chaque jeu de données (ou habitat) sélectionné.

```
--distance_matrix type=space distance=leastcost
```

4.7.3 `--model` : calcul d'un modèle

```
--model variable distW=min:inc:maxcost [vars=var1,...,varn] [raster=r1,...,rn]
```

Paramètres obligatoires

- `variable` : nom de la variable binaire contenu dans le(s) jeux(x) de points
- `distW=[{min:inc:maxcost}]` : pondération de la distance entre la tâche et le point avec une probabilité de 0.05

Paramètres optionnels

- `vars=var1,...,varn` : ajoute d'autres variables de la couche des tâches non liées au graphe (Capacity par exemple)
- `raster=r1,...,rn` : ajoute des variables provenant de raster externe

Description

Calcule un modèle logistique avec chaque métrique locale existant dans chaque graphe sélectionné pour la variable binaire donnée sur chaque jeu de données sélectionné. La `variable` doit exister dans chaque jeu de données sélectionné.

Le paramètre `distW` définit la pondération par la distance entre la tâche et le point avec une probabilité de 0.05.

Exemple

```
--model PRESENCE distW=1000,2000
```

Le résultat est enregistré dans le fichier `model-PRESENCE-dW1000,2000.txt` dans le répertoire du projet :

Graph	Pointset	Metric	DistW	R2	p-value	Coef
2000m-3500cost	point-3500cost	BC_d3500_p0.05_beta1	1000.00	0.229898	0.00268	3.63230e-07
2000m-3500cost	point-3500cost	BC_d3500_p0.05_beta1	2000.00	0.134672	0.12585	4.89398e-08

2000m-3500cost	point-3500cost	F_d3500_p0.05_beta1	1000.00	0.522162	1.29e-5	0.00155784
2000m-3500cost	point-3500cost	F_d3500_p0.05_beta1	2000.00	0.266793	0.00074	0.000145906

Références

[Foltête et al.(2012b), Foltête et al.(2012a), Girardet et al.(2013), Clauzel et al.(2013)]

4.8 Ajout/Suppression d'éléments

4.8.1 `--delta` : suppression d'un élément

```
--delta [filter=condition | [obj=patch|link [sel=id1,id2,...,idn | fsel=file.txt]]
```

Paramètres optionnels

- `filter=condition` : permet de sélectionner les éléments du graphe à tester cf. [Paramètre filter=condition](#)
- `obj=patch|link` : teste la suppression des taches (`obj=patch`) ou des liens (`obj=link`)
- `sel=id1,id2,...,idn` : limite le calcul aux éléments (taches ou liens) listés par leur identifiant
- `fsel=file.txt` : limite le calcul aux éléments (taches ou liens) listés dans le fichier `file.txt`. Le fichier doit contenir un identifiant par ligne.

Description

Calcule une métrique globale en mode delta sur les noeuds et/ou les liens en fonction du paramètre `obj` pour chaque métrique sélectionnée.

Si le paramètre `sel` ou `fsel` est défini, le calcul est effectué seulement sur les éléments listés. Le résultat est enregistré dans un attribut des taches et/ou des liens. Le projet est enregistré sauf si l'option globale `-nosave` est utilisée.

Exemples

Calcule la métrique NC en delta pour toutes les taches :

```
--gmetric NC --delta obj=patch
```

Calcule la métrique EC en delta pour les taches ayant l'identifiant 2 et 3 :

```
--gmetric EC d=1000 p=0.05 --delta obj=patch sel=2,3
```

4.8.2 `--gremove` : suppression de plusieurs éléments

```
--gremove global_metric_name [param1=val ...]
[patch=id1,id2,...,idn|fpatch=file.txt] [link=id1,id2,...,idm|flink=file.txt]
```

Paramètre obligatoire

- `global_metric_name` : acronyme de la métrique globale à calculer (EC, ...)

Paramètres optionnels

- `param1=val` : paramètre(s) de la métrique, si elle en a. Les intervalles ne sont pas permis pour cette commande.
- `patch=id1,id2,...,idn` : enlève les taches listées par leur identifiant

- `fpatch=file.txt` : enlève les taches listées dans le fichier *file.txt*. Le fichier doit contenir un identifiant par ligne.
- `link=id1,id2,...,idn` : enlève les liens listés par leur identifiant
- `flink=file.txt` : enlève les liens listés dans le fichier *file.txt*. Le fichier doit contenir un identifiant par ligne.

Description

Pour chaque graphe sélectionné, enlève l'ensemble des noeuds et/ou liens listés et calcule la métrique globale donnée. La liste des identifiants peut être donnée directement sur la ligne de commande ou bien dans un fichier séparé.

Le résultat est seulement affiché à l'écran. Pour chaque graphe, la commande affiche le nombre de taches et de liens réellement enlevés du graphe. Le nombre de taches ne varient pas, mais le nombre de liens peut varier car les liens connectés à une tache enlevée sont aussi enlevés. Si un identifiant n'existe pas, il sera ignoré.

Exemple

```
--gremove NC patch=2,3
```

Cette commande calcule la métrique NC sur chaque graphe sélectionné après avoir enlevé les taches 2 et 3.

Résultat affiché :

```
Global indice NC
Graph 2000m-3500cost
Remove 2 patches and 5 links
NC : 25.0

Graph 2000m-3500cost_comp
Remove 2 patches and 8 links
NC : 24.0

Graph 2000m_euclid
Remove 2 patches and 7 links
NC : 9.0

Graph 2000m_euclid_comp
Remove 2 patches and 9 links
NC : 9.0
```

La même commande peut être écrite comme suit :

```
--gremove NC fpatch=patch.txt
```

avec un fichier `patch.txt` dans le répertoire courant, contenant un identifiant par ligne :

```
2
3
```

4.8.3 `-gttest` : suppression et ajout itératif d'éléments

```
--gttest nstep global_metric_name [param1=val ...] obj=patch|link
sel=id1,id2,...,idn | fsel=file.txt
```

Paramètres obligatoires

- `nstep` : nombre d'éléments successifs à ajouter
- `global_metric_name` : acronyme de la métrique globale (PC, ...)
- `obj=patch|link` : type d'éléments à tester, taches (`obj=patch`) ou liens (`obj=link`)
- `sel=id1,id2,...,idn` : liste des identifiants des éléments (taches ou liens) à tester
- `fsel=file.txt` : fichier contenant les identifiants des éléments (taches ou liens) à tester. Le fichier *file.txt* doit contenir un identifiant par ligne.

Paramètres optionnels

- `param=val` : paramètre(s) de la métrique, si elle en a. Les intervalles ne sont pas permis pour cette commande.

Description

Cette commande commence par supprimer les éléments (taches ou liens) sélectionnés par le paramètre `sel` ou `fsel`. Puis elle remet l'élément supprimé qui maximise la métrique *global_metric_name*. La remise d'un élément est répétée autant de fois que le paramètre *nstep*. L'ensemble du traitement est effectué pour chaque graphe sélectionné. Pour chacun de ces graphes, les résultats sont enregistrés dans 2 fichiers textes. Le premier liste les éléments ajoutés avec la valeur de la métrique correspondante. Le second, plus détaillé, liste à chaque étape l'ajout de chaque élément.

Exemple

```
--gtest 3 IIC obj=patch sel=1,2,3,5,6,10,15,16
```

Cette commande enlève les 8 taches sélectionnées par leur identifiant et remet successivement les 3 taches qui maximisent la métrique globale IIC.

Pour le graphe *2000m_euclid*, le premier fichier se nomme *gtest-2000m_euclid-IIC.txt* et contient :

Step	Id	IIC
0	init	1.7825075712618167E-6
1	1	1.753327449219068E-6
2	15	1.7793775301389374E-6
3	16	1.780788239231567E-6

La première ligne (Step 0), correspond à la valeur de la métrique après suppression des 8 taches. Les lignes suivantes donnent, à chaque étape, quel élément a été sélectionné ainsi que la nouvelle valeur de la métrique après l'ajout de l'élément.

Pour le même graphe, le fichier détaillé se nomme *gtest-2000m_euclid-IIC-detail.txt* et contient le détail de chaque étape :

Step	Id	IIC
1	init	1.6852479916976776E-6
1	16	1.6866587007903073E-6
1	1	1.753327449219068E-6
1	2	1.685330641324248E-6
1	3	1.6852916014495828E-6
1	5	1.6858693558888445E-6
1	6	1.6852486177082585E-6
1	10	1.6854627861279089E-6
1	15	1.6994856702980223E-6
2	init	1.7533274492190677E-6

2	16	1.7547381583116972E-6
2	2	1.753410098845639E-6
2	3	1.7538298458957464E-6
2	5	1.753948813410235E-6
2	6	1.7533280752296487E-6
2	10	1.753542243649299E-6
2	15	1.7793775301389374E-6
3	<code>init</code>	1.7793775301389372E-6
3	16	1.780788239231567E-6
3	2	1.7794601797655085E-6
3	3	1.7799512085537507E-6
3	5	1.7799988943301041E-6
3	6	1.7795590940743919E-6
3	10	1.7795923245691686E-6

4.8.4 `--addpatch` : ajout itératif de taches

```
--addpatch npatch=val [hab=name] gridres=min:inc:max [capa=capa_file]
| patchfile=file.gpkg [capa=capa_field]
```

Paramètres généraux

- `npatch=val` : nombre de taches à ajouter
- `hab=name` : nom de l'habitat dans lequel les taches seront ajoutées dans le cas où la métrique a été calculée sur un graphe contenant plusieurs habitats

Description

Cette commande teste successivement l'ajout d'une nouvelle tache parmi un ensemble prédéfini et conserve la tache qui maximise la métrique globale sélectionnée. Ce processus est répété autant de fois que le paramètre `npatch` et ce, pour chaque métrique globale sélectionnée. En sortie, un sous répertoire du projet est créé pour chaque métrique, contenant le détail des résultats.

Pour chaque test d'une nouvelle tache, le graphe est recalculé incluant cette nouvelle tache ainsi que les liens reliant cette tache aux autres déjà existantes, ensuite la métrique globale est recalculée sur ce nouveau graphe. Quand l'ensemble des taches possibles a été testé, celle maximisant la métrique donnée est ajoutée au projet. Ce processus est itéré jusqu'à obtenir `npatch` nouvelles taches dans le projet.

Les taches créées ont par défaut une emprise au sol d'un pixel, si le pixel d'occupation du sol correspondant à la tache à tester est déjà dans la catégorie habitat ou bien est en dehors de la zone d'étude, la tache est ignorée. Les taches à tester peuvent être donnée par l'intermédiaire d'une grille régulière ou bien d'un ensemble de points ou polygones provenant d'une couche vecteur.

Test sur un jeu de données vectoriel

```
--addpatch npatch=val [hab=name] patchfile=file.gpkg [capa=capa_field]
```

Pour chaque élément de la couche vecteur (point ou polygone), le programme teste l'ajout d'une tache sur le(s) pixel(s) couvrant l'objet vectoriel, si le(s) pixel(s) ne sont pas en NoData ou bien déjà dans la classe habitat.

Le paramètre `capa` permet de définir un attribut de la couche contenant une valeur de capacité pour chaque tache testée. Si le paramètre `capa` n'est pas renseigné, la capacité des nouvelles taches sera de 1.

Test sur une grille régulière

```
--addpatch npatch=val [hab=name] gridres=min:inc:max [capa=capa_file]
```

Teste l'ajout d'une tache sur une grille régulière dont la résolution est définie par `gridres`.

Le paramètre `capa` permet de spécifier un fichier raster (au format TIFF ou AsciiGrid) donnant une valeur de capacité potentielle en tout point de l'espace. Si la capacité est nulle, aucune tache ne sera testée à cette position. Le fichier raster peut avoir une autre résolution que la grille ou que la carte de paysage. Si le paramètre `capa` n'est pas renseigné, la capacité des nouvelles taches sera de 1.

Exemples

```
--gmetric IIC --addpatch npatch=5 gridres=100
```

Ajoute 5 taches maximisant la métrique IIC en testant l'ajout d'une tache tous les 100 mètres. Le sous répertoire `addpatch_n5_IIC_graph_res100` contient les résultats :

- `addpatch_IIC_graph.gpkg` : contient les taches ajoutées avec la valeur de la métrique au moment de l'ajout
- `addpatch_IIC_graph.txt` : contient le nombre de taches ajoutées avec la valeur de la métrique au moment de l'ajout
- `links_IIC_graph.shp` : contient l'ensemble des liens du graphe
- `topo-links_IIC_graph.gpkg` : contient l'ensemble des liens du graphe en version topologique
- `detail/` : sous répertoire contenant le détail des tests d'ajout de tache à chaque étape
- `detail/detail_i.gpkg` : ensemble des points testés pour l'ajout de la i-ème tache avec la valeur de la métrique en attribut.

La ligne de commande ci-dessous est équivalente à la précédente mais s'exécutera à 3 résolutions différentes (100,200,500). Les résultats seront stockés dans 3 répertoires, un pour chaque résolution.

```
--usemetric IIC_graph --addpatch npatch=5 gridres=100,200,500
```

Un autre exemple avec un jeu de points :

```
--usemetric IIC_graph --addpatch npatch=5 patchfile=testpoint.gpkg
```

Les résultats seront stockés dans le répertoire `addpatch_n5_IIC_graph_vecttestpoint.gpkg`.

Limitations

La commande `--addpatch` ne fonctionne qu'avec des graphes sans distances intra-taches et de jeux de liens en topologie complète.

Attention, cette commande modifie le nombre de tache du projet, l'exécution de commandes après celle-ci peuvent générer des incohérences. Il est fortement déconseillé d'ajouter des commandes à la suite de la commande `--addpatch`.

Références

[[Clauzel et al.\(2015a\)](#), [Foltête et al.\(2014\)](#)]

4.8.5 `--remelem` : suppression itérative d'éléments

```
--remelem nstep global_metric_name [param1=val ...] obj=patch|link  
[sel=id1,id2,...,idn | fsel=file.txt]
```


Paramètres obligatoires

- `nstep` : nombre d'éléments successifs à supprimer
- `global_metric_name` : acronyme de la métrique globale (EC, ...)
- `obj=patch|link` : type d'éléments à supprimer, taches (`obj=patch`) ou liens (`obj=link`)

Paramètres optionnels

- `sel=id1,id2,...,idn` : sélection des identifiants des éléments (taches ou liens) à tester
- `fsel=file.txt` : fichier contenant les identifiants des éléments (taches ou liens) à tester. Le fichier *file.txt* doit contenir un identifiant par ligne.
- `param=val` : paramètre(s) de la métrique, si elle en a. Les intervalles ne sont pas permis pour cette commande.

Description

La commande `--remelem` teste la suppression des éléments du graphe sélectionné (taches ou liens selon le paramètre `obj`) comme la commande `--delta`. Puis elle supprime l'élément qui minimise la métrique `global_metric_name`. La procédure est répétée autant de fois que le paramètre `nstep`. Si le paramètre `sel` ou `fsel` est utilisé, le test de suppression ne se fait que sur les éléments sélectionnés. L'ensemble du traitement est effectué pour chaque graphe sélectionné. Pour chacun de ces graphes, le résultat est enregistré dans un fichier texte qui liste les éléments supprimés avec la valeur de la métrique correspondante.

Exemple

```
--remelem 3 IIC obj=patch
```

Cette commande supprimera successivement les 3 taches qui minimisent la métrique globale IIC, pour chaque graphe.

Pour le graphe *2000m_euclid*, le fichier se nomme `rempatch-IIC-2000m_euclid.txt` et contient :

Step	Id	IIC
0	init	1.7825075712618167E-6
1	120	1.070922766241333E-6
2	145	8.001250544646545E-7
3	118	6.05326129866607E-7

La première ligne (Step 0), correspond à la valeur initiale de la métrique. Les lignes suivantes donnent, à chaque étape, quel élément a été supprimé ainsi que la nouvelle valeur de la métrique après la suppression de celui-ci.

Référence

[Foltête et al.(2016)]

4.9 Changements d'occupation du sol

4.9.1 `--landmod` : changements d'occupation du sol

```
--landmod zone=filezones.gpkg id=fieldname code=fieldname [sel=id1,...,idn]
```

Paramètres obligatoires

- `zone=filezones.gpkg` : couche de polygones contenant les changements d'occupation du sol
- `id=fieldname` : nom d'un attribut de la couche servant à identifier les polygones. Si les valeurs ne sont pas uniques, les polygones ayant le même identifiant seront appliqués en un seul changement.
- `code=fieldname` : nom d'un attribut de la couche stockant la nouvelle catégorie d'occupation du sol du polygone.

Paramètre optionnel

- `sel=id1,...,idn` : liste des identifiants de polygones à traiter. Si ce paramètre n'est pas renseigné tous les polygones de la couche seront traités.

Description

Cette commande doit être placée avant les autres commandes de calcul. Elle va dupliquer le projet pour chaque polygone de la couche `zone` et modifier l'occupation du sol couvert par le(s) polygone(s). Les commandes suivant celle-ci seront exécutées sur chaque projet modifié. Les projets créés ne contiennent pas les jeux de liens ni les graphes du projet initial.

Les projets créés seront nommés par l'identifiant du (ou des) polygones et stockés dans le répertoire du projet.

Exemple

```
--landmod zone=zones.gpkg id=id code=code --linkset distance=euclid --graph
--gmetric IIC
```

Pour chaque identifiant de polygones contenus dans la couche `zones.gpkg`, un nouveau projet nommés par la valeur de `id` sera créé dans un sous répertoire du projet actuel. Sur chacun des ces projets, un jeu de liens en distance euclidienne, suivi d'un graphe seront créés et la métrique IIC sera calculée sur ce graphe. Au final, chaque sous répertoire du projet actuel contiendra un fichier `IIC.txt` contenant la valeur de la métrique tenant compte de chaque changement d'occupation du sol.

Si l'objectif est de calculer l'impact des changements au moyen d'une métrique globale, il est plus simple d'utiliser la commande suivante `--landmodgm`.

Référence

[[Sahraoui et al.\(2017\)](#)]

4.9.2 `--landmodgm` : changements d'occupation du sol sur métrique globale

```
--landmodgm zone=filezones.gpkg id=fieldname code=fieldname [sel=id1,...,idn]
```

Paramètres obligatoires

- `zone=filezones.gpkg` : couche de polygones contenant les changements d'occupation du sol
- `id=fieldname` : nom d'un attribut de la couche servant à identifier les polygones. Si les valeurs ne sont pas uniques, les polygones ayant le même identifiant seront appliqués en un seul changement.
- `code=fieldname` : nom d'un attribut de la couche stockant la nouvelle catégorie d'occupation du sol du polygone

Paramètre optionnel

- `sel=id1,...,idn` : liste des identifiants de polygones à traiter. Si ce paramètre n'est pas renseigné tous les polygones de la couche seront traités.

Description

Cette commande sera exécutée sur chaque métrique globale calculée ou sélectionnée précédemment. Elle va dupliquer le projet pour chaque polygone de la couche zone et modifier l'occupation du sol couvert par le(s) polygone(s). Le jeu de liens et le graphe de la métrique sélectionnée seront créés dans chaque nouveau projet ainsi que la métrique globale.

Les projets créés seront nommés par l'identifiant du (ou des) polygones et stockés dans le répertoire du projet.

Un fichier `.csv` sera créé stockant les résultats pour chaque métrique globale.

Exemple

```
--linkset distance=euclid --graph --gmetric IIC
--landmodgm zone=zones.gpkg id=id code=code
```

Pour chaque identifiant de polygones contenus dans la couche `zones.gpkg`, un nouveau projet nommé par la valeur de `id` sera créé dans un sous répertoire du projet actuel. Sur chacun des ces projets, le jeu de liens et le graphe de la métrique IIC seront créés et la métrique IIC sera calculée. Au final, un fichier `landmod-iic.csv` sera créé contenant la valeur de la métrique pour chaque changement d'occupation du sol.

4.10 Options

4.10.1 -nosave

Cette option empêche l'enregistrement du projet par une commande. Elle est utile quand vous ne voulez pas qu'une commande modifie le projet.

4.10.2 -distconv

Cette option permet de définir des paramètres optionnels à la conversion de distance (ie. quand une distance est entourée d'accolades). Pour l'instant, il y a un seul paramètre : `excost=val`, pour exclure les chemins contenant des coûts supérieurs ou égaux à `val` de la régression. Cette option est utile pour exclure les chemins traversant des éléments barrière ou quasi infranchissable de la conversion de distance.

4.10.3 -proc

Défini le nombre de processeurs (ou coeurs) utilisés par Graphab. Par défaut, la ligne de commande utilise la valeur définie dans la fenêtre des préférences. Pour plus de détails, voir la section [Parallélisme](#).

4.10.4 -mpi

Cette option indique que Graphab est utilisé dans un environnement MPI. Pour plus de détails, voir la section [Parallélisme](#).

Chapter 5

Exemples de lignes de commandes

Tous les exemples qui suivent peuvent être testés avec les données d'exemple fourni sur le site.

5.1 Création d'un projet jusqu'au graphe

Crée un nouveau projet nommé `sample` dans le répertoire courant à partir de la carte `os.tif`

```
java -jar graphab-3.0.jar --create sample os.tif
```

Ajoute un habitat au projet `sample` nommé `fields`, les taches d'habitats correspondent au code 8 de la carte.

```
java -jar graphab-3.0.jar --project sample/sample.xml --habitat name=fields codes=8
```

Crée un jeu de liens `fields_euclid` en distance euclidienne et en topologie planaire basé sur l'habitat `fields`

```
java -jar graphab-3.0.jar --project sample/sample.xml --linkset distance=euclid  
name=fields_euclid
```

Crée un graphe `g_fields_euclid_500` élagué à 500m

```
java -jar graphab-3.0.jar --project sample/sample.xml --graph name=g_fields_euclid_500  
threshold=500
```

L'ensemble des commandes précédentes peuvent être exécutées en une seule fois :

```
java -jar graphab-3.0.jar --create sample os.tif --habitat name=fields codes=8  
--linkset distance=euclid name=fields_euclid  
--graph name=g_fields_euclid_500 threshold=500
```

5.2 Habitat

Crée un nouvel habitat `fields1a` dont les taches doivent avoir une superficie d'au moins 1 are (0.01 hectare).

```
java -jar graphab-3.0.jar --project sample/sample.xml --habitat name=fields1a codes=8  
minarea=0.01
```

Le même traitement peut être réalisé en partant de l'habitat `fields` en éliminant les taches ayant une capacité $< 100m^2$

```
java -jar graphab-3.0.jar --project sample/sample.xml --vhabitat name=fields1a  
from=fields mincapa=100
```

Crée un autre habitat `forest` à partir des codes 3 et 4 de l'occupation du sol. Les taches dépassant les 500 mètres en longueur ou largeur sont redécoupées.

```
java -jar graphab-3.0.jar --project sample/sample.xml --habitat name=forest codes=3,4
maxsize=500
```

5.3 Jeux de liens et graphes

5.3.1 Jeu de liens

Crée un jeu de liens `fields_cost` à partir de l'habitat `fields` en distance coût et topologie planaire.

```
java -jar graphab-3.0.jar --project sample/sample.xml --usehabitat fields
--linkset distance=cost name=fields_cost 8=1 2=3 3,4=10 0,6=100 1,5=1000
```

Crée un jeu de liens `fields_comp_cost` à partir de l'habitat `fields` en distance coût et topologie complète. En topologie complète, il est important d'indiquer une distance max, ici 1000 unités de coût.

```
java -jar graphab-3.0.jar --project sample/sample.xml --usehabitat fields
--linkset distance=cost topo=complete maxcost=1000 name=fields_comp_cost
8=1 2=3 3,4=10 0,6=100 1,5=1000
```

5.3.2 Graphes

Crée un graphe `g_fields_comp_cost` basé sur le jeu de liens `fields_comp_cost`

```
java -jar graphab-3.0.jar --project sample/sample.xml --uselinkset fields_comp_cost
--graph name=g_fields_comp_cost
```

Crée un graphe `g_fields_comp_cost_500` basé sur le jeu de liens `fields_comp_cost` et élagué à une distance coût équivalente à 500 mètres. La conversion de distance est effectuée automatiquement : 500 mètres est converti dans ce cas en 261.5 unités de coût.

```
java -jar graphab-3.0.jar --project sample/sample.xml --uselinkset fields_comp_cost
--graph name=g_fields_comp_cost_500 threshold={500}
```

5.4 Métriques

5.4.1 Métrique globale

Métrique globale à plusieurs distances

Calcule la métrique EC sur le graphe non élagué `g_fields_comp_cost` en faisant varier le paramètre `d` de 100 à 900 par pas de 200 :

```
java -jar graphab-3.0.jar --project sample/sample.xml
--usegraph g_fields_comp_cost --gmetric EC d=100:200:900 p=0.05
```

Le résultat est stocké dans le fichier `EC.txt` dans le répertoire du projet :

Graph	d	p	EC
<code>g_fields_comp_cost</code>	100.0	0.05	301923.01540184487
<code>g_fields_comp_cost</code>	300.0	0.05	359617.35658238636
<code>g_fields_comp_cost</code>	500.0	0.05	409521.1200767072
<code>g_fields_comp_cost</code>	700.0	0.05	451094.0363294105
<code>g_fields_comp_cost</code>	900.0	0.05	486124.7711331516

Calcule la métrique EC sur le graphe non élagué `g_fields_comp_cost` en faisant varier le paramètre `d` en distance métrique (de 500m à 2000m) avec conversion automatique en distance coût :

```
java -jar graphab-3.0.jar --project sample/sample.xml
--usegraph g_fields_comp_cost --gmetric EC d={500:500:2000} p=0.05
```

Le résultat est stocké dans le fichier EC.txt dans le répertoire du projet :

Graph	d	p	EC
g_fields_comp_cost	261.5279516552625	0.05	349052.7726285711
g_fields_comp_cost	486.5823280879073	0.05	406448.22457832936
g_fields_comp_cost	699.6560264253405	0.05	451028.7186860784
g_fields_comp_cost	905.3042342469763	0.05	486981.8241196714

5.5 Multi-habitat

Crée un jeu de liens `fields_forest_euclid` entre l'habitat `fields` et l'habitat `forest` en distance euclidienne et topologie complète seuillée à 1000m. Avec le mot clé `inter`, ne sont conservés que les liens entre deux taches d'habitat différent.

```
java -jar graphab-3.0.jar --project sample/sample.xml --mergehabitat fields,forest
--linkset distance=euclid topo=complete maxcost=1000 inter name=fields_forest_euclid
```

5.6 Afficher le projet

Affiche les éléments du projet :

```
java -jar graphab-3.0.jar --project sample/sample.xml --show
```

Résultat :

```
==== Habitats ====
0 - fields
2 - fields1a
1 - forest

==== Jeux de liens ====
fields_comp_cost
fields_cost
fields_euclid
fields_forest_euclid

==== Graphes ====
g_fields_comp_cost
g_fields_comp_cost_500
g_fields_euclid_500

==== Métriques ====
ec_d100_p0.05_g_fields_comp_cost
ec_d261.52795165526254_p0.05_g_fields_comp_cost
ec_d300_p0.05_g_fields_comp_cost
....
```

Affiche le détail des habitats :

```
java -jar graphab-3.0.jar --project sample/sample.xml --show_habitats
```

Résultat :

```
==== Habitats ====
0 - fields
```

Nom de l'habitat : fields
taches : 270

Connexité des taches : 4-connexité
Codes de l'habitat : [8]
Surface min. de l'habitat : 0.0 m2
Taille max. de l'habitat : 0.0 m

==== Habitats ====

2 - fields1a
Nom de l'habitat : fields1a
taches : 68

Habitat à partir d'une couche vecteur

==== Habitats ====

1 - forest
Nom de l'habitat : forest
taches : 1334

Connexité des taches : 4-connexité
Codes de l'habitat : [3, 4]
Surface min. de l'habitat : 0.0 m2
Taille max. de l'habitat : 500.0 m

Chapter 6

Capacités de traitement et limitations

Les méthodes basées sur les graphes sont connues pour fournir un cadre de modélisation efficace, mais peuvent poser une question de capacité de calcul. Deux points spécifiques ont reçu une attention particulière dans Graphab : (1) le calcul des jeux de liens, (2) le calcul des métriques de connectivité. Tous ces calculs ont été optimisés par parallélisation. Ce mode de développement améliore l'efficacité de calcul en utilisant une architecture multiprocesseur, un processeur quatre-cœur permettant théoriquement un calcul quatre fois plus rapide.

Dans [Foltête et al.(2012a)], plusieurs tests ont été effectués afin de mesurer la capacité de calcul de Graphab dans différentes configurations. Trois configurations ont été comparées : 1) un cœur (3Go RAM) correspondant à un ordinateur standard ; 2) quatre cœurs (6 Go RAM) correspondant à une station de travail ; 3) 20 cœurs (15 Go RAM) correspondant à un serveur. La carte d'occupation du sol était un fichier raster de 14000*18000 pixels (252 millions de pixels) représentant l'occupation du sol de la région Franche-Comté (France) à une résolution spatiale de 10 mètres. La carte contenait 22 634 taches d'habitats. Le tableau 6.1 donne quelques temps de calcul pour la création de liens.

Topologie	Distance	Ordinateur standard	Station de travail	Serveur
Complète	Euclidienne	1927s (32 min)	516s (8 min)	133s (2 min)
	Moindre coût	19252s (5h 21 min)	4301s (1h 11 min)	1037s (17 min)
Planaire	Euclidienne	43s	12s	2.6s
	Moindre coût	1080s (18 min)	295s (5 min)	82s (1 min)

Table 6.1: Temps de calcul (en seconde) pour la création de différents jeux de liens.

La mémoire allouée au programme joue aussi un rôle important. Si il n'y a pas assez de mémoire vive, le calcul sera plus lent voire même ne pourra pas aboutir (message `OutOfMemoryError` ou `GC Overhead`). Le menu Fichier / Préférences / Mémoire permet de régler la mémoire allouée pour Graphab. Si vous avez une version 32 bits de Java, Graphab sera limité à environ 2 Go (2000 Mo) de mémoire. Si votre ordinateur a plus que 2 Go de mémoire vive il est fortement conseillé d'installer une version 64 bits de Java pour utiliser la mémoire disponible au-delà des 2 Go.

6.1 Parallélisme

6.1.1 Un ordinateur : threads

Si votre ordinateur a plus qu'un cœur (la plupart), vous pouvez tirer partie de la parallélisation pour accélérer vos calculs avec Graphab. Il faut définir le nombre de cœurs (ou processeurs) que Graphab peut utiliser avec

l'option `-proc` avant la commande `--project` :

```
java -jar graphab-3.0.jar -proc 8 --project path2myproject/myproject.xml ...
```

Par défaut, en ligne de commande, le nombre de coeurs utilisés correspond au nombre défini dans la fenêtre Préférences de l'interface graphique.

En augmentant le nombre de coeurs utilisés par Graphab, vous augmentez, par la même occasion, la taille de la mémoire utilisée par Graphab.

6.1.2 Cluster : MPI

Graphab peut aussi être utilisé sur des clusters de calcul supportant Java avec OpenMPI.

```
mpirun java -jar graphab-3.0.jar -mpi --project path2myproject/myproject.xml ...
```

Seulement certaines commandes peuvent être utilisées dans l'environnement MPI : `--gmetric`, `--cmetric`, `--lmetric`, `--delta`, `--addpatch`, `--gtest`, `--remelem`, `--landmod`

6.2 Gestion mémoire

Par défaut, la taille de la mémoire utilisable par Graphab est dépendante du système et peut varier de 128 MB à plusieurs GB. Si vous avez un gros projets, certaines commandes seront lentes voir même stopperont le programme à cause d'un manque de mémoire. Si l'exécution se termine par une erreur `OutOfMemoryError` ou `GC overhead`, vous devez augmenter la mémoire disponible pour Graphab.

Pour définir manuellement la mémoire maximale utilisable par Graphab, il faut utiliser l'option Java `-Xmx` :

```
java -Xmx4g -jar graphab-3.0.jar ... # 4Gb allocated
java -Xmx1500m -jar graphab-3.0.jar ... # 1500 Mb -> 1.5Gb allocated
```

Si vous ne pouvez pas allouer plus de 1 GB ou 1.5 GB alors que votre ordinateur a plus de mémoire vive, vous utilisez sûrement une version 32-bit de Java qui est limitée à moins de 2 GB. Pour tester votre version de Java :

```
java -version
```

Si vous avez une version 32-bit, installez une version 64-bit de Java pour allouer plus de mémoire à Graphab.

Chapter 7

Métriques

Famille	Métrique de connectivité	Code	Niveau de calcul			Delta métriques	Multi habitat
			Global	Comp.	Local		
Métriques pondérées	Flux	F	×	×	×		×
	Connectivité équivalente	EC	×	×		×	×
	Probabilité de Connectivité	PC	×	×		×	
	Flux d'interaction	IF			×		×
	Fractions de delta Probabilité de Connectivité	dPC				×	
	Indice de centralité intermédiaire	BC			×		×
	Indice Intégral de Connectivité	IIC	×	×		×	
	Flux circuit	CF			×		
Métriques de surface	Taille moyenne des composantes	MSC	×				
	Taille de la plus grande composante	SLC	×				
	Probabilité de coïncidence de classe	CCP	×				
	Taille de composante attendue	ECS	×				
Métriques topologiques	Degré du noeud	Dg			×		
	Coefficient de groupement	CC			×		
	Centralité de proximité	CCe			×		
	Excentricité	Ec			×		
	Corrélation de connectivité	CCor			×		
	Nombre de composantes	NC	×				
	Diamètre	GD	×	×		×	
	Indice d'Harary	H	×	×		×	
	Lambda de Wilks	W	×				

Table 7.1: Liste des métriques de connectivité et de leur niveau de calcul. Les codes sont indiqués suivant le nom anglais des métriques.

Terme	Signification
n	Nombre de taches
nc	Nombre de composantes
n_k	Nombre de taches de la composante k
N_i	Ensemble des taches voisines de la tache i
a_i	Capacité de la tache i (en général sa surface)
ac_k	Capacité de la composante k (somme des capacités des taches qui la composent)
A	Surface de la zone d'étude
d_{ij}	distance entre les taches i et j (en général la distance de moindre coût qui les sépare)
$e^{-\alpha d_{ij}}$	Probabilité de déplacement entre les taches i et j
α	Frein de la distance vis-à-vis des déplacements
β	Exposant permettant de pondérer plus ou moins la capacité

Table 7.2: Termes mathématiques utilisés

7.1 Métriques pondérées

7.1.1 Flux

Flux (F)	Formule	Signification
Niveau global	$S\#F = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n a_j^\beta e^{-\alpha d_{ij}}$	Pour le graphe entier : somme des dispersions potentielles à partir de toutes les taches.
Niveau local	$F_i = \sum_{\substack{j=1 \\ j \neq i}}^n a_j^\beta e^{-\alpha d_{ij}}$	Pour la tache focale i : somme des capacités des taches différentes de i et pondérées en fonction de leur distance minimale à la tache focale en parcourant le graphe. Cette somme est un indicateur de la dispersion potentielle à partir de la tache i , ou inversement à destination de la tache i .
Valeurs	<p>Les valeurs dépendent de la définition de a. Si a représente une surface, F exprime une surface.</p> <p>Valeur minimale : 0</p> <p>Valeur maximale : surface totale de l'habitat si $\beta = 1$</p>	
Commentaire	<p>Le chemin du graphe utilisé est celui qui maximise $e^{-\alpha d}$, c'est-à-dire celui qui minimise la distance d (ou le coût) entre les taches i et j.</p> <p>Cette métrique est appelée Area Weighed Flux (AWF) dans certaines références. Dans Graphab cependant, a est plus général car il représente la capacité des taches, qui peut être leur surface ou un autre critère choisi par l'utilisateur. De même la pondération est variable en fonction du paramètre β.</p> <p>Dans CS22, AWF est calculé seulement à partir des taches reliées directement à la tache focale, alors que Graphab tient compte des taches indirectement connectées.</p>	
Références	[Urban and Keitt(2001)] [Saura and Torné(2009)] [Foltête et al.(2012b)]	

7.1.2 Connectivité équivalente

Connectivité équivalente (<i>EC</i>)	Formule	Signification
Niveau global Niveau composante Delta	$EC = \sqrt{\sum_{i=1}^n \sum_{j=1}^n a_i a_j e^{-\alpha d_{ij}}}$	Racine carrée de la somme des produits de la capacité de tous les couples de tâches pondérées par leur probabilité d'interaction.
Valeurs	L'unité correspond à l'unité des capacités des tâches. Valeur minimale : 0 Valeur maximale : capacité totale des tâches	
Commentaire	Pour chaque couple de tâches, le chemin du graphe utilisé est celui qui maximise $e^{-\alpha d}$, c'est-à-dire celui qui minimise la distance d (ou le coût) entre les tâches i et j .	
Référence	[Saura et al.(2011)]	

7.1.3 Probabilité de Connectivité

Probabilité de Connectivité (<i>PC</i>)	Formule	Signification
Niveau global Niveau composante Delta	$PC = \frac{1}{A^2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j e^{-\alpha d_{ij}}$	Somme des produits de la capacité de tous les couples de tâches pondérées par leur probabilité d'interaction, divisée par le carré de la zone d'étude. Ce rapport équivaut à la probabilité que deux individus tirés au hasard dans la zone d'étude parviennent à entrer en contact.
Valeurs	Les valeurs correspondant à une probabilité. Valeur minimale : 0 Valeur maximale : 1	
Commentaire	Pour chaque couple de tâches, le chemin du graphe utilisé est celui qui maximise $e^{-\alpha d}$, c'est-à-dire celui qui minimise la distance d (ou le coût) entre les tâches i et j . Les métriques <i>PC</i> et <i>EC</i> sont reliées par cette égalité : $PC = \left(\frac{EC}{A}\right)^2$ Cette métrique est désactivée si a ne représente pas la surface des tâches.	
Référence	[Saura and Pascual-Hortal(2007)]	

7.1.4 Flux d'interaction

Flux d'interaction (IF)		
	Formule	Signification
Niveau local	$IF_i = \sum_{j=1}^n a_i^\beta a_j^\beta e^{-\alpha d_{ij}}$	Somme des produits de la capacité de la tâche focale avec toutes les autres tâches, pondérées par leur probabilité d'interaction. Cette métrique remplace l'ancienne métrique FPC nommée PC_{flux} dans les articles.
Valeurs	Valeur minimale : 0 Valeur maximale : somme des capacités	
Commentaire	<p>Pour chaque couple de tâches, le chemin du graphe utilisé est celui qui maximise $e^{-\alpha d}$, c'est-à-dire celui qui minimise la distance d (ou le coût) entre les tâches i et j.</p> <p>Cette métrique correspond à la contribution locale d'une tâche à l'indice PC : $PC = \frac{1}{A^2} \sum_i IF_i$ avec $\beta = 1$, ainsi que pour EC : $EC^2 = \sum_i IF_i$.</p> <p>Elle équivaut à $dPC_{i,flux}/2 + dPC_{i,area}$ à un facteur près. Cependant la métrique IF est obtenue beaucoup plus rapidement car son calcul n'est pas fondé sur la suppression itérative des tâches (mode delta).</p> $IF_i = EC^2 \left(\frac{1}{2} dPC_{i,flux} + dPC_{i,area} \right)$ <p>On peut aussi faire le lien entre IF et la métrique locale F :</p> $IF_i = a_i^2 + a_i F_i$	
Références	[Foltête et al.(2014)],[Sahraoui et al.(2017)]	

7.1.5 Fractions de delta Probabilité de Connectivité

Fractions de delta Probabilité de Connectivité (dPC_i , $dPC_{i,area}$, $dPC_{i,flux}$, $dPC_{connector}$)		
	Formules	Signification
Delta	$dPC_i = \frac{(PC - PC'_i)}{PC} =$ $dPC_{i,area} + dPC_{i,flux} + dPC_{i,connector}$ $dPC_{i,area} = \frac{a_i^2}{EC^2}$ $dPC_{i,flux} = 2 \frac{IF_i - a_i^2}{EC^2}$	<p>Taux de variation entre la valeur de l'indice PC et la valeur PC' représentant la suppression de la tâche i. La valeur de dPC_i se décompose en trois parties :</p> <ul style="list-style-type: none"> - $dPC_{i,area}$ correspond à la variation due à la perte de la surface suite à la suppression de la tâche i ; - $dPC_{i,flux}$ correspond à la variation due à la perte des interactions entre la tâche i et les autres tâches ; - $dPC_{i,connector}$ correspond à la variation due à la modification des chemins reliant les autres tâches et passant initialement par i.
Valeurs	Valeur minimale : 0 Valeur maximale : 1	
Commentaire	Si a ne représente pas la surface des tâches, dPC_{area} n'exprime pas une perte de surface mais de capacité.	
Référence	[Saura and Rubio(2010)]	

7.1.6 Indice de centralité intermédiaire

Indice de centralité intermédiaire (BC)	Formule	Signification
Niveau local	$BC_i = \sum_j \sum_k a_j^\beta a_k^\beta e^{-\alpha d_{jk}}$ $j, k \in \{1..n\}, k < j, i \in P_{jk}$	<p>Somme des plus courts chemins passant par la tâche focale i, chaque chemin étant pondéré par le produit des capacités des tâches reliées et de leur probabilité d'interaction.</p> <p>P_{jk} représente l'ensemble des tâches traversées par le plus court chemin entre les tâches j et k.</p>
Valeurs	<p>Les valeurs dépendent du paramétrage. Elles correspondent à un poids de transit potentiel.</p> <p>Valeur minimale : 0</p> <p>Valeur maximale : surface totale de l'habitat au carré.</p>	
Commentaire	<p>Avec un paramétrage de $\alpha = 0$ et $\beta = 0$ (pondération uniforme des chemins), on retrouve l'indice BC tel qu'il est utilisé dans d'autres types de graphes.</p> <p>Un paramétrage de $\alpha = 0$ et $\beta = 1$ donne aux chemins un poids égal au produit des capacités des tâches qu'ils relient, quelle que soit leur distance.</p> <p>Dans [Foltête et al.(2012a), Foltête et al.(2012b)], l'indice BC_l a été proposé pour pondérer plus fortement les chemins supérieurs au critère choisi (distance de dispersion par exemple). Mais des tests ont ensuite montré que cet indice était très fortement corrélé au BC pondéré avec $\alpha = 0$.</p> <p>Dans [Bodin and Saura(2010)], le BC_{pc} correspond au BC pondéré avec d égal à la distance de dispersion, α tel que $e^{-\alpha d} = 0.05$ et $\beta = 1$.</p>	
Références	[Bodin and Saura(2010)] [Foltête et al.(2012a)]	

7.1.7 Indice Intégral de Connectivité

Indice Intégral de Connectivité (IIC)	Formule	Signification
Niveau global Niveau composante Delta	$IIC = \frac{1}{A^2} \sum_{i=1}^n \sum_{j=1}^n \frac{a_i a_j}{1 + nl_{ij}}$	<p>Produit des capacités des tâches divisé par le nombre de liens qui les sépare, la somme étant divisée par le carré de la surface de la zone d'étude. IIC est construit comme l'indice PC, mais en utilisant l'inverse d'une distance topologique plutôt qu'une fonction exponentielle négative de la distance basée sur l'impédance des liens.</p>
Valeurs	<p>Valeur minimale : 0</p> <p>Valeur maximale : 1</p>	
Référence	[Pascual-Hortal and Saura(2006)]	

7.1.8 Flux circuit

Flux circuit (CF)	Formule	Signification
Niveau local	$CF_i = \sum_j^n c_i^j$	Somme des courants traversant la tache focale i . c_i^j représente le courant à travers la tache i lorsque un courant est émis depuis toutes les taches vers la tache j (la tache j est reliée à la terre)
Valeurs	Valeur minimale : 0 Valeur maximale : $(n-1)(n-2)$ si $\beta = 0$ $(n-2) \sum_i^{n-1} a_i$ si $\beta = 1$	
Commentaire	La métrique CF utilise la théorie des circuits électriques. Chaque lien du graphe correspond à une résistance électrique, les sources de courants et la terre sont quant à elles attachées aux taches. Si $\beta = 0$ chaque tache émet un courant unitaire, si $\beta = 1$ chaque tache émet un courant égal à sa capacité. Cette métrique peut être vue comme un équivalent de la métrique BC (avec $\alpha = 0$ et $\beta = 0$) qui tient compte de l'ensemble des chemins possibles et non pas uniquement du chemin le plus court.	
Références	[Girardet et al.(2015)]	

7.2 Métriques de surface

7.2.1 Taille moyenne des composantes

Taille moyenne des composantes (MSC)	Formule	Signification
Niveau global	$MSC = \frac{1}{nc} \sum_{k=1}^{nc} ac_k$	Moyenne des capacités des composantes.
Valeurs	Valeur minimale : capacité minimale Valeur maximale : SLC	

7.2.2 Taille de la plus grande composante

Taille de la plus grande composante (SLC)		
	Formule	Signification
Niveau global	$SLC = \max\{ac_k\}$	Plus grande capacité des composantes.
Valeurs	Valeur minimale : capacité minimale Valeur maximale : capacité maximale	

7.2.3 Probabilité de coïncidence de classe

Probabilité de coïncidence de classe (<i>CCP</i>)		
	Formule	Signification
Niveau global	$CCP = \sum_{k=1}^{nc} \left(\frac{ac_k}{\sum_l ac_l} \right)^2$	Probabilité que deux points choisis au hasard sur le graphe appartiennent à la même composante.
Valeurs	Valeur minimale : $\rightarrow 0$ (autant de composantes que de tâches et capacités uniformes) Valeur maximale : 1 (une seule composante)	
Référence	[Pascual-Hortal and Saura(2006)]	

7.2.4 Taille de composante attendue

Taille de composante attendue (<i>ECS</i>)		
	Formule	Signification
Niveau global	$ECS = \frac{1}{\sum_k ac_k} \sum_{k=1}^{nc} ac_k^2$	Taille attendue d'une composante
Valeurs	Valeur minimale : capacité minimale (autant de composantes que de tâches et capacités uniformes) Valeur maximale : somme des capacités (1 seule composante)	
Référence	[O'Brien et al.(2006)]	

7.3 Métriques topologiques

7.3.1 Lambda de Wilks

Lambda de Wilks (<i>W</i>)		
	Formule	Signification
Niveau global	$W_{Lambda} = \frac{ W }{ T }$	Rapport entre la (co)variance intra-classe (intra composante) $ W $ et la (co)variance totale $ T $.
Values	Valeur minimale : 0 les tâches appartenant à la même composante sont identiques pour toutes les variables sélectionnées. Valeur maximale : 1 les tâches appartenant à la même composante ont leurs variables qui diffèrent autant entre elles qu'avec les autres tâches.	
Reference	[Everitt and Dunn(2001)], [Foltête and Vuidel(2017)]	

7.3.2 Indice d'Harary

Indice d'Harary (H)	Formule	Signification
Niveau global Niveau composante Delta	$H = \frac{1}{2} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{nl_{ij}}$	Somme de l'inverse du nombre de liens séparant tous les couples de tâches.
Valeurs	Valeur minimale : 0 pour un graphe sans lien (autant de composantes que de tâches) Valeur maximale : $\frac{n(n-1)}{2}$ pour un graphe complet	
Commentaire	Pour les couples de tâches non connectées par un chemin on a : $nl_{ij} = +\infty$	
Référence	[Ricotta et al.(2000)]	

7.3.3 Diamètre

Diamètre (GD)	Formule	Signification
Niveau global Niveau composante Delta	$GD = \max_{ij} d_{ij}$ $GD = \max_i Ec_i$	Distance la plus grande entre deux tâches du graphe
Valeurs	Valeur minimale : 0 Valeur maximale : $+\infty$	
Commentaire	Quand les nœuds i et j ne sont pas connexes $d_{ij} = 0$ Cette métrique est la version globale de la métrique Ec_i	

7.3.4 Nombre de composantes

Nombre de composantes (NC)	Formule	Signification
Niveau global	$NC = nc$	Nombre de composantes du graphe
Valeurs	Valeur minimale : 1 Valeur maximale : n	

7.3.5 Degré du noeud

Degré du noeud (Dg)	Formule	Signification
Niveau local	$Dg_i = N_i $	Nombre de liens connectés au noeud i ie. nombre de taches voisines de la tache i
Valeurs	Valeur minimale : 0 Valeur maximale : n	
Commentaire	Il y a équivalence entre le degré d'un nœud et le nombre de nœuds voisins car les graphes ne sont pas orientés et ne contiennent pas de boucle.	

7.3.6 Coefficient de groupement

Coefficient de groupement (CC)	Formule	Signification
Niveau local	$CC_i = \frac{1}{ N_i (N_i - 1)} \sum_{j \in N_i} N_i \cap N_j $	Rapport du nombre de nœuds voisins de i voisins entre eux sur le total possible.
Valeurs	Valeur minimale : 0 Valeur maximale : 1	
Commentaire	Si $ N_i \leq 1 \rightarrow CC_i = 0$	

7.3.7 Centralité de proximité

Centralité de proximité (CCe)	Formule	Signification
Niveau local	$CCe_i = \frac{1}{n_k - 1} \sum_{\substack{j=1 \\ j \neq i}}^{n_k} d_{ij}$	Distance moyenne de la tache i vers toutes les autres taches de sa composante k .
Valeurs	Valeur minimale : 0 Valeur maximale : $+\infty$	
Commentaire	Si $n_k = 1 \rightarrow CCe_i = 0$	

7.3.8 Excentricité

Excentricité (Ec)	Formule	Signification
Niveau local	$Ec_i = \max_j d_{ij}$	Distance maximale de la tâche i vers une autre tâche de sa composante.
Valeurs	Valeur minimale : 0 Valeur maximale : $+\infty$	

7.3.9 Corrélation de connectivité

Corrélation de connectivité ($CCor$)	Formule	Signification
Niveau local	$CCor_i = \frac{ N_i ^2}{\sum_{j \in N_i} N_j }$	Rapport entre le degré du nœud i et le degré de ses nœuds voisins j
Valeurs	Valeur minimale : 0 Valeur maximale : $ N_i $	
Commentaire	Si $ N_i = 0 \rightarrow CCor_i = 0$	
Référence	[Minor and Urban(2008)]	

Bibliography

- [Bodin and Saura(2010)] Orjan Bodin and Santiago Saura, 2010. Ranking individual habitat patches as connectivity providers: Integrating network analysis and patch removal experiments. *Ecological Modelling*, 221(19) 2393 – 2405.
- [Brandes et al.(2008)] U. Brandes, D. Dellinger, M. Gaertler, R. Gorke, M. Hofer, Z. Nikoloski, and D. Wagner, 2008. On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20(2) 172–188.
- [Clauzel et al.(2015a)] Celine Clauzel, Cyrielle Bannwarth, and Jean-Christophe Foltete, 2015a. Integrating regional-scale connectivity in habitat restoration: An application for amphibian conservation in eastern france. *Journal for Nature Conservation*, 23 98 – 107.
- [Clauzel et al.(2015b)] Celine Clauzel, Deng Xiqing, Wu Gongsheng, Patrick Giraudoux, and Li Li, 2015b. Assessing the impact of road developments on connectivity across multiple scales: Application to yunnan snub-nosed monkey conservation. *Biological Conservation*, 192 207 – 217.
- [Clauzel et al.(2013)] Céline Clauzel, Xavier Girardet, and Jean-Christophe Foltête, 2013. Impact assessment of a high-speed railway line on species distribution: Application to the european tree frog (*Hyla arborea*) in franche-comté. *Journal of Environmental Management*, 127 125 – 134.
- [Everitt and Dunn(2001)] Brian S Everitt and Graham Dunn, 2001. *Applied multivariate data analysis*, volume 2. Wiley Online Library.
- [Foltête and Vuidel(2017)] Jean-Christophe Foltête and Gilles Vuidel, 2017. Using landscape graphs to delineate ecologically functional areas. *Landscape Ecology*, 32(2) 249–263.
- [Foltête et al.(2021)] Jean-Christophe Foltête, Gilles Vuidel, Paul Savary, Céline Clauzel, Yohan Sahraoui, Xavier Girardet, and Marc Bourgeois, 2021. Graphab: an application for modeling and managing ecological habitat networks. *Software Impacts*, 8 100065.
- [Foltête et al.(2012a)] Jean-Christophe Foltête, Céline Clauzel, and Gilles Vuidel, 2012a. A software tool dedicated to the modelling of landscape networks. *Environmental Modelling and Software*, 38 316 – 327.
- [Foltête et al.(2012b)] Jean-Christophe Foltête, Céline Clauzel, Gilles Vuidel, and Pierline Tournant, 2012b. Integrating graph-based connectivity metrics into species distribution models. *Landscape Ecology*, 27(4) 557–569.
- [Foltête et al.(2016)] Jean-Christophe Foltête, Geoffroy Couval, Marilyne Fontanier, Gilles Vuidel, and Patrick Giraudoux, 2016. A graph-based approach to defend agro-ecological systems against water vole outbreaks. *Ecological Indicators*, 71 87 – 98.
- [Foltête et al.(2014)] Jean-Christophe Foltête, Xavier Girardet, and Céline Clauzel, 2014. A methodological framework for the use of landscape graphs in land-use planning. *Landscape and Urban Planning*, 124 140 – 150.
- [Girardet et al.(2015)] Xavier Girardet, Géraldine Conruyt-Rogéon, and Jean-Christophe Foltête, 2015. Does regional landscape connectivity influence the location of roe deer roadkill hotspots? *European Journal of Wildlife Research*, 61(5) 731–742.

- [Girardet et al.(2013)] Xavier Girardet, Jean-Christophe Foltête, and Céline Clauzel, 2013. Designing a graph-based approach to landscape ecological assessment of linear infrastructures. *Environmental Impact Assessment Review*, 42 10 – 17.
- [McRae et al.(2008)] Brad H. McRae, Brett G. Dickson, Timothy H. Keitt, and Viral B. Shah, 2008. Using circuit theory to model connectivity in ecology, evolution, and conservation. *Ecology*, 89(10) 2712 – 2724.
- [Minor and Urban(2008)] Emily Minor and Dean Urban, 2008. A graph-theory framework for evaluating landscape connectivity and conservation planning. *Conservation biology*, 22 297–307.
- [Newman(2006)] M. E. J. Newman, 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23) 8577–8582.
- [O’Brien et al.(2006)] Dan O’Brien, Micheline Manseau, Andrew Fall, and Marie-Josée Fortin, 2006. Testing the importance of spatial configuration of winter habitat for woodland caribou: An application of graph theory. *Biological Conservation*, 130(1) 70 – 83.
- [Pascual-Hortal and Saura(2006)] Lucía Pascual-Hortal and Santiago Saura, 2006. Comparison and development of new graph-based landscape connectivity indices: towards the prioritization of habitat patches and corridors for conservation. *Landscape Ecology*, 21(7) 959–967.
- [Ricotta et al.(2000)] C. Ricotta, A. Stanisci, G.C. Avena, and C. Blasi, 2000. Quantifying the network connectivity of landscape mosaics: a graph-theoretical approach. *Community Ecology*, 1(1) 89–94.
- [Sahraoui et al.(2017)] Yohan Sahraoui, Jean-Christophe Foltête, and Céline Clauzel, 2017. A multi-species approach for assessing the impact of land-cover changes on landscape connectivity. *Landscape Ecology*, 32(9) 1819–1835.
- [Saura et al.(2011)] Santiago Saura, Christine Estreguil, Coralie Mouton, and Mónica Rodríguez-Freire, 2011. Network analysis to assess landscape connectivity trends: Application to european forests (1990–2000). *Ecological Indicators*, 11(2) 407 – 416.
- [Saura and Pascual-Hortal(2007)] Santiago Saura and Lucía Pascual-Hortal, 2007. A new habitat availability index to integrate connectivity in landscape conservation planning: Comparison with existing indices and application to a case study. *Landscape and Urban Planning*, 83(2–3) 91 – 103.
- [Saura and Rubio(2010)] Santiago Saura and Lidón Rubio, 2010. A common currency for the different ways in which patches and links can contribute to habitat availability and connectivity in the landscape. *Ecography*, 33(3) 523–537.
- [Saura and Torné(2009)] Santiago Saura and Josep Torné, 2009. Conefor sensinode 2.2: A software package for quantifying the importance of habitat patches for landscape connectivity. *Environmental Modelling and Software*, 24(1) 135 – 139.
- [Urban and Keitt(2001)] Dean Urban and Timothy Keitt, 2001. Landscape connectivity: A graph-theoretic perspective. *Ecology*, 82(5) 1205–1218.