



RAPPORT DE STAGE :

AGENTS EXPERTS EN ANALYSE DE VIABILITÉ

Laetitia ZALESKI

Encadrants :
Isabelle ALVAREZ
Jean-Pierre BRIOT

2017

Introduction

L'objectif est d'explorer l'utilisation de la viabilité dans le cadre d'un processus de gestion participatif collectif. Il s'agit de concevoir des outils d'assistance technique à la disposition des acteurs sociaux participants d'un processus participatif de gestion, tel que celui de ressources naturelles.

Plus précisément, nous comptons étudier les modes d'assistance (à l'aide d'agents assistants) de l'utilisation d'analyses de viabilité dans un contexte collectif. Ainsi permettre des analyses comparatives de différents points de vue (suivant les objectifs et contraintes posés par les différents acteurs sociaux), de manière à faciliter les bases objectives de négociation.

Pour ce faire, nous avons développé un premier scénario simplifié de dynamique de ressources naturelles dans une réserve de biodiversité (nous prenons l'exemple d'une réserve marine) et des actions possibles (prélèvements directs via la pêche et effets indirects, via le tourisme). Ce scénario est ensuite modélisé via la viabilité, d'abord pour chaque participant individuellement, puis en réunissant les données de chacun. Ainsi, des participants dont les intérêts divergent devrait être accompagnés au long de leur négociation.

On bénéficie de la nouvelle implémentation des calculs de viabilité à base de kd-trees (Alvarez et al., 2016) [1], grâce à laquelle les calculs de noyaux de viabilité seront effectués, ainsi que de la plateforme web Vino [2], utilisée pour les intersections. Ce projet s'appuie le récent projet SimParc (sur la simulation participative de parcs nationaux pour la conservation de la biodiversité) (Briot et al., 2016) [3] et de premières expérimentations dans un tel cadre d'analyse de viabilité (Wei et al., 2013) [4].

Nous exposerons d'abord deux scénarios (en deux et trois dimensions) et leurs dynamiques associées, en nous appuyant sur les travaux sur le tourisme et le développement durable de Cassagrandi et Rinaldi [5]. Grâce à ces dynamiques nous effectuerons des calculs de viabilité individuels pour différents profils. Enfin nous pourrons en venir aux analyses comparatives des différents profils.

1 Conception de modèles mathématiques pour la gestion d'espace protégé

Les modèles proposés se placent dans un scénario de gestion de réserve marine protégée ouverte à l'activité touristique. On considère dans les deux modèles la population animale (de tortues) et l'activité touristique. Des touristes se déplacent jusqu'à la réserve pour, entre autres, observer ces tortues.

Dans le modèle 3D, on considère que la réserve est ouverte à la pêche et on ajoute une variable : le capital des pêcheurs. Les pêcheurs proposent des services aux touristes (tours en bateau par exemple), leur pêche de poissons entraîne la prise accidentelle de tortues de mer.

On s'inspire des travaux sur le tourisme de Casagrandi et Rinaldi. [5],

1.1 Paramètres des modèles

Dans ces modèles, les animaux sont au sommet de la chaîne alimentaire et la dynamique de la population animale suit donc une loi de Verhulst avec un coefficient d'accroissement g et une capacité d'accueil K_A qui dépend de la qualité de l'environnement. La qualité de l'environnement dépend du nombre de touristes (les touristes influent négativement sur la qualité de l'environnement), ainsi que de l'effort de restauration de l'environnement ϵ . On a $K_A = 1 + M [1 + \eta T / (1 + \epsilon)]^{-1}$ avec η les dommages causés à l'environnement par un touriste. Les animaux sont négativement affectés par les touristes (en proportion du nombre de touristes). Les touristes viennent proportionnellement à l'attractivité du site (Casagrandi et Rinaldi, 2002 [5]), le paramètre d s'inspirant du paramètre a de Casagrandi et Rinaldi.

ζ représente la probabilité de voir une tortue, l le pourcentage de chance pour un touriste de gêner la reproduction de l'animal (les animaux sont stressés par le nombre de touristes) ou de le tuer. ζ et ϵ sont des paramètres de contrôle, c'est-à-dire que l'on considère qu'il est possible d'influer dessus, en fermant certaines zones aux touristes (pour ζ) ou en nettoyant les déchets laissés par ces derniers par exemple (pour ϵ). De plus, on rajoute une pêche possible, avec p le pourcentage d'animaux pris (p pouvant valoir 0 dans le cas où la pêche est interdite).

En ce qui concerne le modèle 3D, on ajoute la variable C représentant le capital des pêcheurs. Aux paramètres précédents on ajoute les charges h payées en moyenne par un pêcheur, la valeur m_p des poissons capturés pour chaque prise accidentelle de tortues et m_t l'argent qu'un touriste reverse à un pêcheur en moyenne.

Les dynamiques suivantes n'ont pas encore été confrontées à la réalité et devront être calées sur les données d'espaces protégés. Il faut aussi noter que l'étude asymptotique reste à faire (identification des attracteurs et calcul de stabilité).

1.2 Modèle 2D (Animaux, Touristes)

1.2.1 Dynamique

On obtient la dynamique suivante :

$$\frac{dA}{dt} = A(t)g\left(1 - \frac{A(t)}{1 + K_A}\right) - \zeta l T(t)A(t) - pA(t)$$

$$\frac{dT}{dt} = T(t)\left(-\frac{cT(t)}{T(t) + \phi} - d\right) + \zeta a A(t)$$

Voici le portrait de phase du modèle précédent, calculé en scilab, en prenant pour jeu de paramètre : $l = 0.01, g = 1, M = 5000, c = 0.01, p = 0.3, a = 100.0, e = 0.01, \eta = 0.0005, \phi = 1, d = 1$

Les différentes courbes correspondent à différentes conditions initiales. Elles convergent toutes vers un même attracteur.

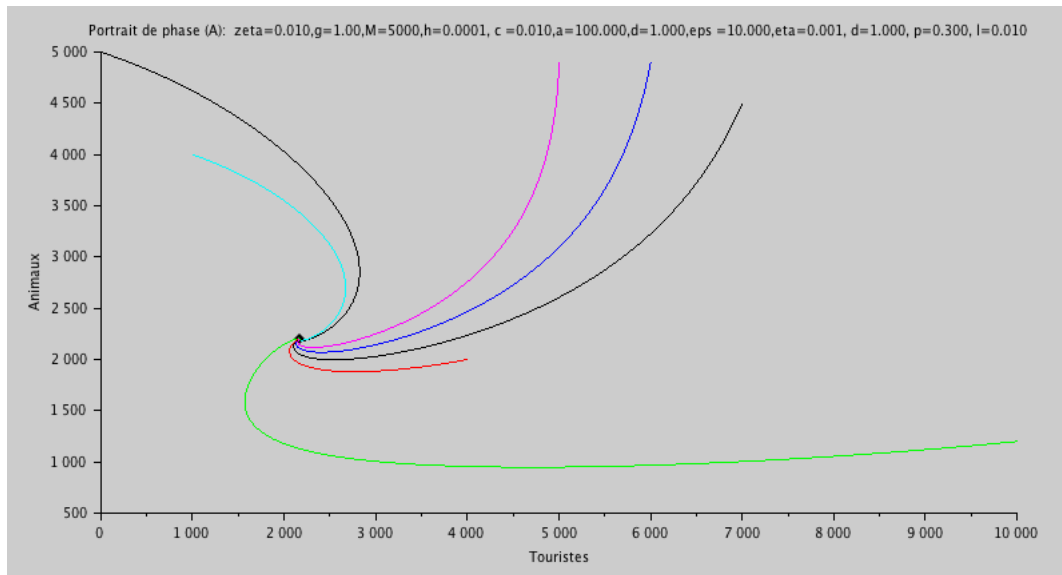


FIGURE 1 – Touristes en fonction des animaux

1.3 Modèle 3D (Capital, Animaux, Touristes)

1.3.1 Dynamique

Les interactions entre les trois variables du modèles peuvent se résumer comme suit :

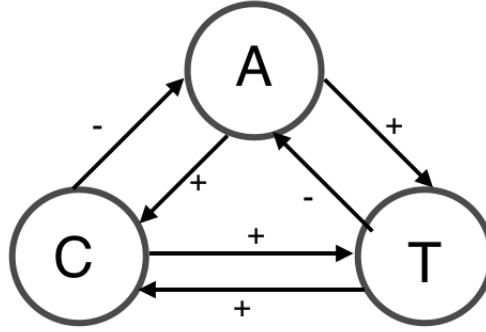


FIGURE 2 – Interactions entre les variables C, A et T

On obtient le modèle suivant :

$$\begin{aligned} \frac{dC}{dt} &= -\delta C + pA(t)m_p + T(t)m_t - h \\ \frac{dA}{dt} &= A(t)g\left(1 - \frac{A(t)}{1 + \frac{M}{K_e}}\right) - \zeta lT(t)A(t) - pA(t) \\ \frac{dT}{dt} &= T(t)\left(-\frac{cT(t)}{T(t) + \phi} + \frac{cT(t)}{C(t) + \phi_2} - d\right) + \zeta aA(t) \end{aligned}$$

1.3.2 Simulation

En testant pour le jeu de paramètre suivant, on obtient les portraits de phases de la figure 3. D'autres portraits de phases en 2D sont disponibles en Annexe.

$\zeta = 0.01, l = 0.01, g = 2, M = 5000, h = 0.0001, c = 0.01, p = 0.2, a = 100.0, e = 0.01, \eta = 0.0005, \epsilon = 10, \phi = 1, d = 1, \delta = 0.01, h = 12000, m_p = 15, m_t = 0.2$

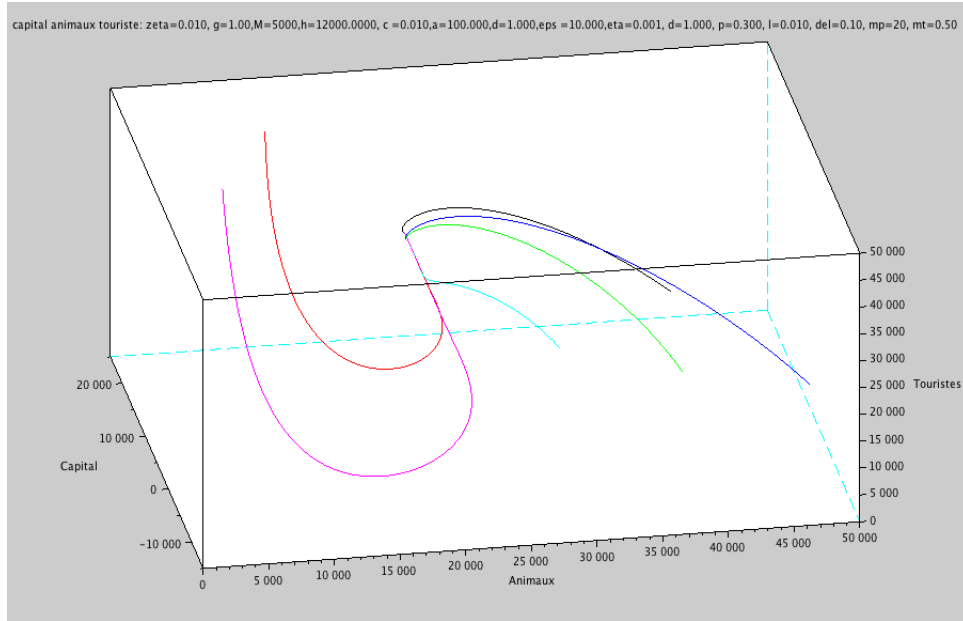


FIGURE 3 – Graphe représentant la dynamique Capital-Animaux-Touristes

2 Noyaux de viabilités

La théorie de la viabilité [Aubin, 1991] [6] considère un système dynamique pouvant être influencé par un ou plusieurs contrôles et étudie sa compatibilité avec des contraintes. Ici nos systèmes peuvent être influencés par deux contrôles : ζ et ϵ . On appelle *viable* un état initial pour lequel il existe au moins une évolution qui reste dans un ensemble de contraintes indéfiniment.

Le *noyau de viabilité* $\text{Viab}(K)$ rassemble les états à partir desquels on trouve toujours une fonction de contrôles qui permette au système de rester dans l'ensemble des contraintes K . La méthode principale pour résoudre un problème de viabilité est de déterminer le noyau de viabilité.

2.1 Calcul à l'aide des kdTrees

Le calcul de noyaux de viabilité se fait à l'aide du logiciel Viabilitree (Alvarez et Al) [1], qui effectue ses calculs en s'appuyant sur les kd-tree.

Un kdtree est une structure de donnée permettant de stocker une partition de l'espace dans un arbre binaire, rendant les recherches plus rapide. Chaque noeud non terminal divise l'espace en deux demi-espaces.

2.2 Noyaux 2D

Pour une population animale comprise entre 1750 et 5000 individus, et entre 3500 et 4000 touristes. Avec ζ pouvant prendre les valeurs 0.02 ou 0.04 et epsilon pouvant prendre les valeurs 0 ou 10, et $p = 0.3$. On obtient le noyau suivant :

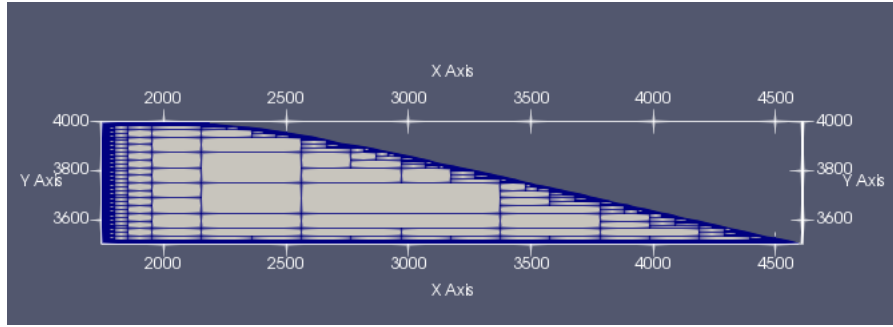


FIGURE 4 – Exemple de noyau de viabilité (modèle 2D)

En changeant les paramètres $g = 1$ et $p = 0.02$ et en posant les bornes 1200-5000 pour les Animaux et 0-3500 pour les Touristes, on obtient le noyau suivant :

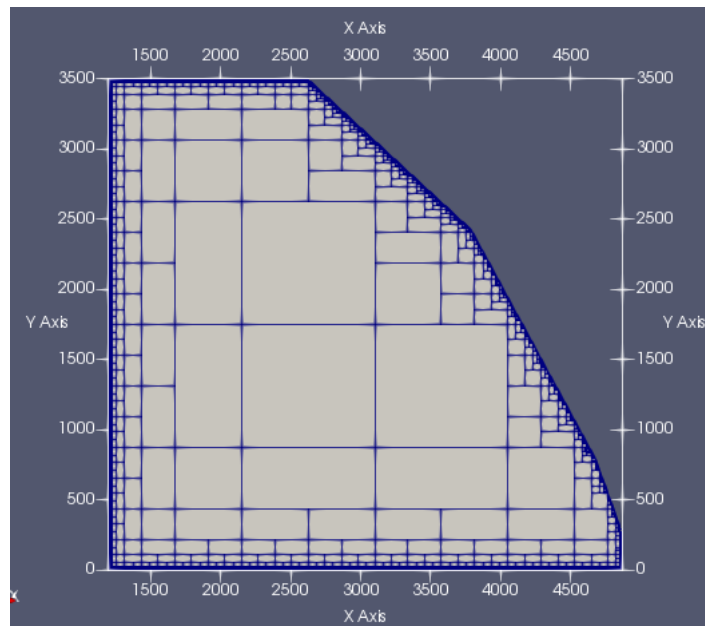


FIGURE 5 – Exemple de noyau de viabilité (modèle 2D)

2.3 Noyaux 3D

On effectue les tests avec un capital compris entre 0 et 25000, une population animale comprise entre 2000 et 5000, et une activité touristique comprise entre 1500 et 2500. Avec ζ pouvant prendre les valeurs 0.02 ou 0.04 et epsilon pouvant prendre des valeurs de 0 à 100 par pas de 10. On obtient les résultats suivants :

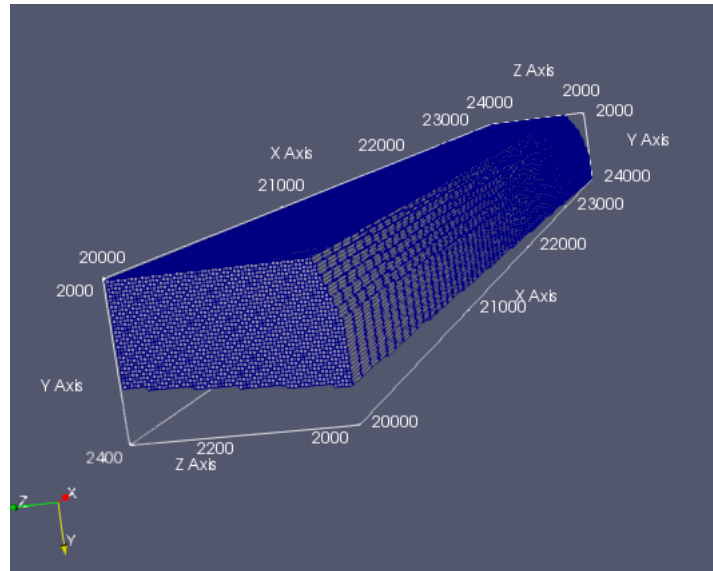


FIGURE 6 – Noyau de viabilité (modèle 2D)

3 L'assistant viabilité

L'objectif est que chaque utilisateur ait le moyen de calculer son propre noyau de viabilité, pour cela il doit entrer son ensemble de contraintes. Nous voulons aussi offrir la possibilité aux utilisateurs de modifier les paramètres de la dynamique ainsi que les moyens de contrôles. L'utilisateur devra répéter cette étape jusqu'à obtenir un noyau non vide. En effet un noyau vide indique que les contraintes qu'il a choisies ne sont pas compatibles avec la dynamique et les contrôles qu'il se donne. Dans cet objectif j'ai intégré les modèles 2D et 3D à Viabilitree (en scala) [1] et j'ai testé leur fonctionnement.

Afin de savoir s'il est possible de trouver un ensemble de contrôles qui satisfasse les contraintes de tous les utilisateurs, nous procéderons en plusieurs étapes, dont les premières ont été testées pendant le stage.

Tout d'abord l'intersection des ensembles de contraintes des utilisateurs. Si cette intersection est vide, tout ou partie des utilisateurs devront changer leurs contraintes pour que la session puisse continuer. En cas d'intersection de contraintes non vide, on calcule l'intersection des noyaux de chacun des utilisateurs. Si cette intersection est vide, tout ou partie des utilisateurs devront modifier leur problème de viabilité et recalculer leur noyau. Si cette intersection est non-vide, il est possible de relancer un calcul de noyau à l'intérieur de cette intersection. Il est nécessaire de faire un calcul pour chaque dynamique et contrôles des utilisateurs, si ceux-ci utilisent des dynamiques et contrôles différents. La prochaine étape est de calculer l'intersection de ces noyaux. Si les utilisateurs ont la même dynamique et les mêmes contrôles, il s'agit du noyau que nous avons calculé. Si cette intersection est vide, tout ou partie des utilisateurs devront reformuler leur problème de viabilité. Sinon, un ensemble d'états communs a été trouvé. Si les utilisateurs appliquent des politiques de contrôles différentes, ils n'auront peut être pas les moyens pour rester dans cette dynamique : il est nécessaire de calculer, pour l'union de tous les contrôles des utilisateurs, quels contrôles peuvent être utilisé pour rester dans cet ensemble.

Ainsi, la discussion porte d'abord sur les contraintes, puis sur les moyens d'action à considérer et enfin sur la solution à appliquer, (comme le font les travaux du collectif Commod [7])

Ces étapes sont exprimées dans le schéma suivant :

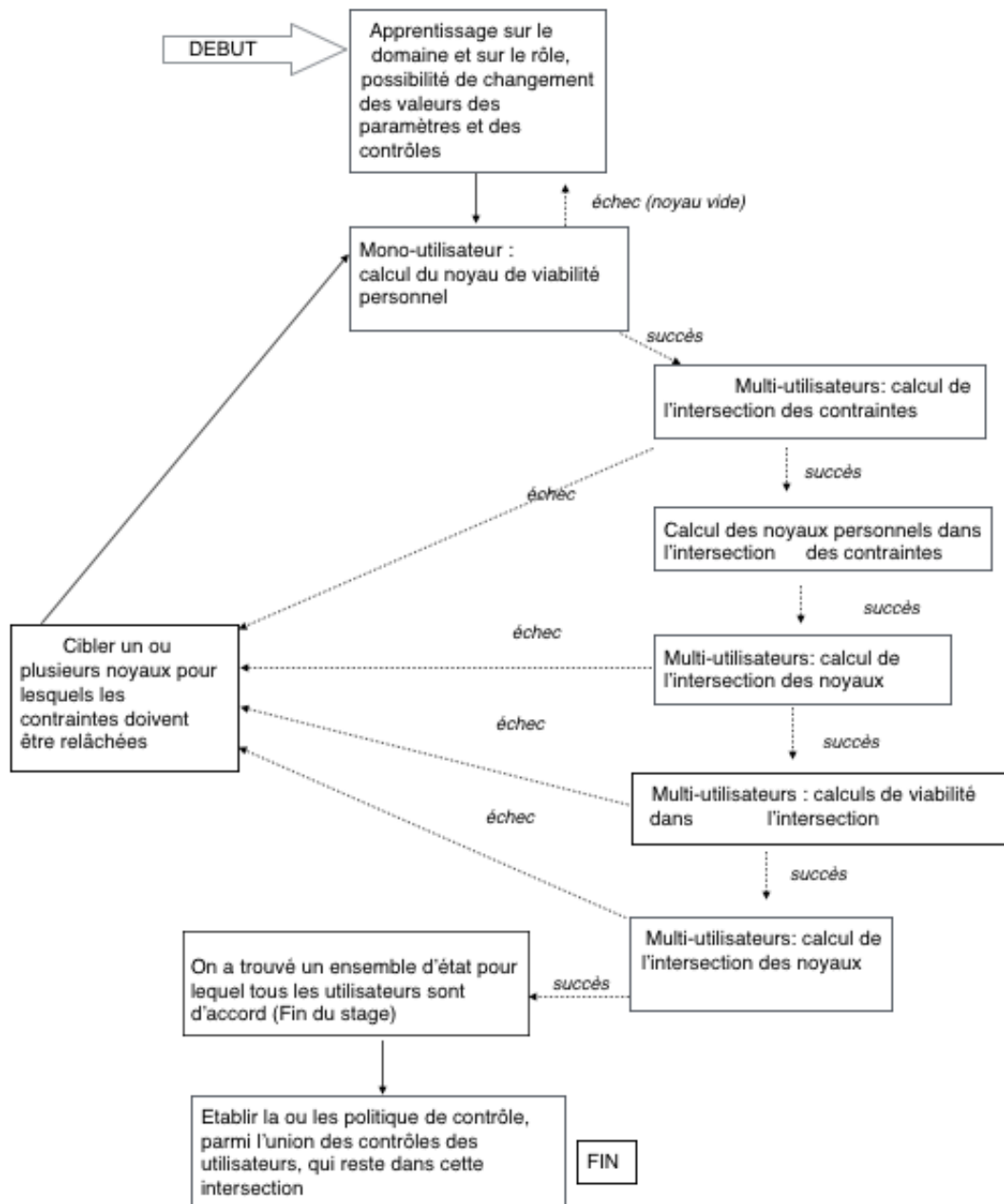


FIGURE 7 – Déroulement d’une session

3.1 Etat de l'art

3.1.1 Assistants dans les jeux sérieux et la gestion participative

Posons d'abord les bases de la collaboration, qui sont, d'après Martinez-Maldonado (2013) [8], une compréhension mutuelle, un dialogue, un regroupement d'informations et enfin un consensus.

Selon Katz (2013)[9] si l'étudiant répond de manière incorrecte à une question il est préférable de lui montrer pourquoi cette réponse n'est pas correcte en lui expliquant sous quelles conditions elle le serait.

Dans notre assistant nous pourrions, en cas de noyau vide, montrer où la dynamique sort du noyau (en superposant le portrait de phase et les bornes de l'utilisateur, lui dire au bout de combien de temps on sort) .

Il convient aussi de formuler les conseils et aides de l'assistant afin de guider l'utilisateur. Tai (2013) [10] enjoint à construire une relation entre l'assistant et l'utilisateur, notamment grâce à un avatar, et une formulation bien choisie afin de minimiser la frustration de l'utilisateur en cas d'échecs répétés. Le premier point est soutenu par les articles de Yannier [11] et Blessing [12] dans lesquels la présence de personnages et d'un scénario est décrite comme un point important pour la compréhension et l'adhésion de l'utilisateur.

Pour éviter la frustration de l'utilisateur, l'assistant peut proposer un noyau proche des contraintes de l'utilisateur et qui serait non-vide, au bout d'un certain nombre d'échecs (par exemple en étendant les bornes de l'utilisateur jusqu'à un point d'équilibre)

3.1.2 Assistants dans les outils mathématiques ou de designs

Chandrasekaran [13] propose le framework *propose-critique-modify* pour l'analyse de modèles, dont des problèmes de modélisation où une partie du processus est une recherche de relaxation. Cette méthode consiste à proposer une solution, la vérifier, en cas d'échec de vérification, critiquer cette proposition en identifiant la source de l'échec puis de boucler en reproposant une solution en tenant compte des critiques. De plus, l'article suggère de proposer des modifications de manières ordonnées en commençant par plus significantes. Par exemple : Modification 1 : Passer de $A > 1000$ à $A > 750$; Modification 2 : Passer de $T < 10000$ à $T < 12000$, avec la Modification 1 élargissant plus le noyau que la modification 2. Il faut bien sûr rester raisonnable et ne pas proposer des modifications trop éloignées des propositions de l'utilisateur (par exemple $\pm 20\%$ des bornes initiales). De manière générale, même si le noyau n'est pas vide, cette proposition de relaxation peut être faite à l'utilisateur.

Finn utilise le formalisme précédent dans son article sur un assistant pour l'analyse préliminaire de modélisation (1992) [14] . Son assistant propose plusieurs fonctionnalités qui peuvent être directement applicables pour l'assistant viabilité, telle que la suppression de dimension ou l'ajustement de l'équation de départ en ajoutant/supprimant/ajustant des phénomènes. Par exemple, l'utilisateur peut supprimer le facteur de gêne par les touristes et ajuster le paramètre de reproduction des poissons. Le but étant de sélectionner, appliquer et évaluer plusieurs options de modélisation. En mode pluri-utilisateurs, si tous les agents ne sont pas d'accord entre eux sur le modèle, les noyaux devront être non vides pour chaque modèle.

3.2 Intersection

Si l'intersection de contraintes est non vide, on peut calculer l'intersection des noyaux.

Un grand domaine de contraintes ne veut pas nécessairement dire un grand noyau. Par exemple, pour notre modèle, la capacité d'accueil des animaux est de 5000, ainsi, si un utilisateur relâche sa borne supérieure sur A de 5000 à 10 000, cela n'aura pas d'influence sur son noyau.

On utilise la plate-forme Vino (Viable Numerical Object) [2] afin de calculer et visualiser l'intersection de deux noyaux de viabilité. Cette plate-forme permet de calculer et d'afficher des intersections de noyaux produits par des logiciels différents. Nous pouvons alors cette intersection et l'importer sous Viabilitree [1] Sur la figure 8 on a par exemple l'intersection du noyau ayant pour bornes $0 \leq A \leq 5000$; $0 \leq T \leq 3000$ (1) et celui ayant pour bornes $1750 \leq A \leq 4000$; $3500 \leq T \leq 4000$ (2) dont on a fait l'intersection sur Vino [2] puis importé sur Viabilitree (Figure 9).

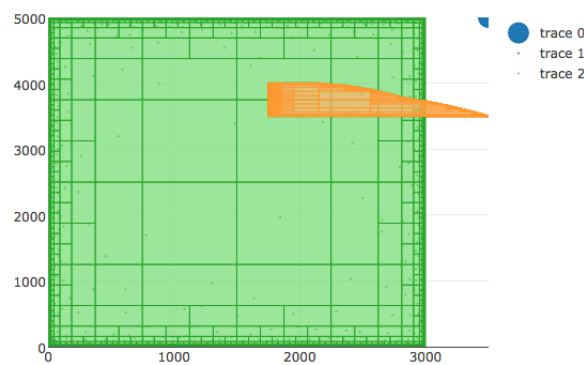


FIGURE 8 – Visualisation de l'intersection de deux Noyaux sur la plateforme Vino

On effectue cette intersection en convertissant d'abord les noyaux du format *kd-tree* au format *bars*. Ce format exprime le noyau sous forme de barres horizontales d'une épaisseur déterminée. Pour un noyau en 2 dimensions, chaque barre est constituée de 3 valeurs : la coordonnée T de départ, de la coordonnée A , et de la coordonnée T d'arrivée (la coordonnée A de départ et d'arrivée sur étant les mêmes).

3.3 Calcul de noyau dans l'intersection : Le noyau collectif

Une fois l'intersection calculée, il est désormais possible de calculer un nouveau noyau à l'intérieur de cette intersection. Ce calcul est effectué pour la dynamique de chacun des utilisateurs. Il est nécessaire car l'intersection de zones viables n'est pas nécessairement viable.

Pour procéder à ce calcul nous transformons l'ensemble au format *bars* en *domain* capable d'être lu par la fonction de calcul de noyaux. Ce domaine est ensuite utilisé comme ensemble de contraintes. Le calcul de noyau personnel avait des contraintes sur A et T qui formaient la zone de recherche d'ensemble viable, pour ce nouveau calcul, l'ensemble de recherche est restreint à l'intersection des noyaux.



FIGURE 9 – Resultat de l'intersection des noyaux (1) et (2) pour chacun des jeu de paramètres

Ici on obtient le même noyau pour les paramètres des deux utilisateurs. De plus ces utilisateurs ont les mêmes contrôles. On a donc une politique de contrôle qui pourra les satisfaire tout les deux.

3.4 Le cas d'un noyau vide

On imagine un scénario avec 3 participants. Chacun calcule son noyau de viabilité avec les bornes et paramètres suivants :

- Utilisateur 1 (bleu) :
 - Paramètres : $l = 0.01$, $\mathbf{g=2}$, $M = 5000$, $h = 0.0001$, $c = 0.01$, $\mathbf{p=0.2}$, $a = 100.0$, $e = 0.01$, $\eta = 0.0005$, $\phi = 1$, $d = 1$, $\delta = 0.01$, $h = 12000$, $m_p = 15$, $m_t = 0.2$
 - Contraintes : $1400.0 < A < 3000.0$, $3750.0 < T < 4100.0$
- Utilisateur 2 (vert) :
 - Paramètres : $l = 0.01$, $\mathbf{g=1}$, $M = 5000$, $h = 0.0001$, $c = 0.01$, $\mathbf{p=0.3}$, $a = 100.0$, $e = 0.01$, $\eta = 0.0005$, $\phi = 1$, $d = 1$, $\delta = 0.01$, $h = 12000$, $m_p = 15$, $m_t = 0.2$
 - Contraintes : $1000.0 < A < 5000.0$, $2000.0 < T < 3600.0$
- Utilisateur 3 (rose) :
 - Paramètres : $l = 0.01$, $\mathbf{g=1}$, $M = 5000$, $h = 0.0001$, $c = 0.01$, $\mathbf{p=0.2}$, $a = 100.0$, $e = 0.01$, $\eta = 0.0005$, $\phi = 1$, $d = 1$, $\delta = 0.01$, $h = 12000$, $m_p = 15$, $m_t = 0.2$
 - Contraintes : $1000.0 < A < 5000.0$, $2000.0 < T < 5000.0$

Ce scénario nous donne les noyaux suivants :

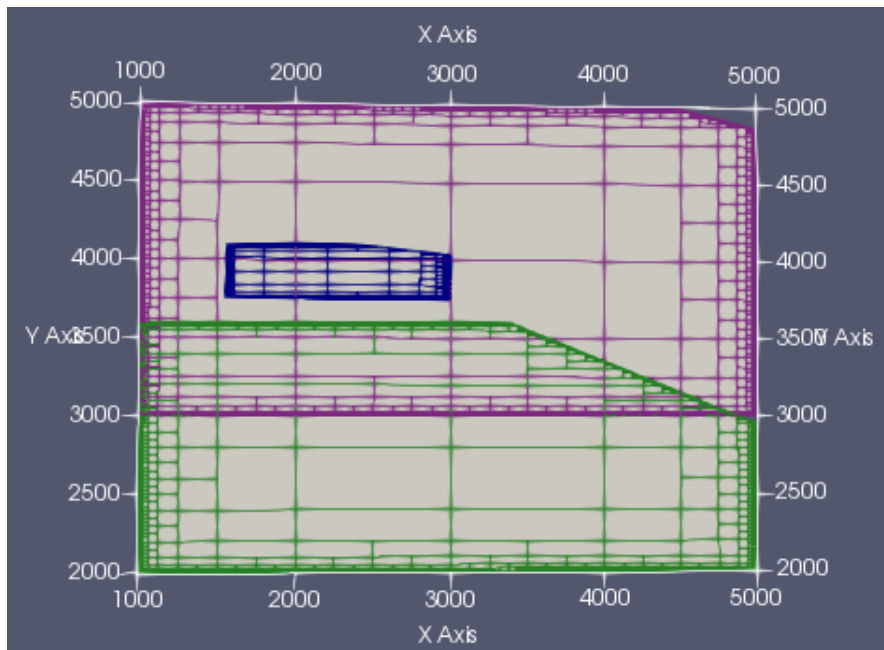


FIGURE 10 – Superposition des noyaux des trois utilisateurs

On remarque que l'intersection des ensembles de contraintes des 3 utilisateurs est vide, et donc avec ces données on ne peut pas trouver de solution qui satisfasse tous les utilisateurs. Il est donc nécessaire de déterminer la marche à suivre dans le cas d'un noyau de viabilité collectif vide. Plusieurs options ont été envisagées (mais non-implémentées) :

- Le noyau le plus petit.

Il est possible de calculer la surface de chaque noyau pour déterminer le plus petit. On peut ensuite demander à l'utilisateur correspondant de relâcher ses contraintes jusqu'à trouver une intersection non vide.

Dans notre exemple le noyau le plus petit est celui de l'utilisateur 1, en relâchant ses contraintes il obtient ce nouveau noyau (bleu clair) :

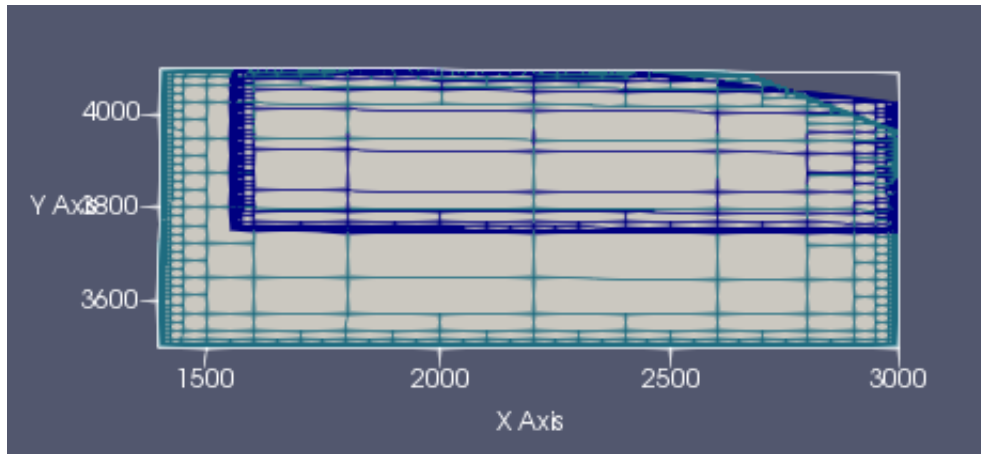


FIGURE 11 – Nouveau noyau de l'utilisateur 1, en comparaison avec l'ancien

Ce qui donne bien une intersection non vide. On peut relancer un calcul de viabilité à l'intérieur de cette intersection, ce qui donne le noyau cyan pour chaque jeu de paramètres.

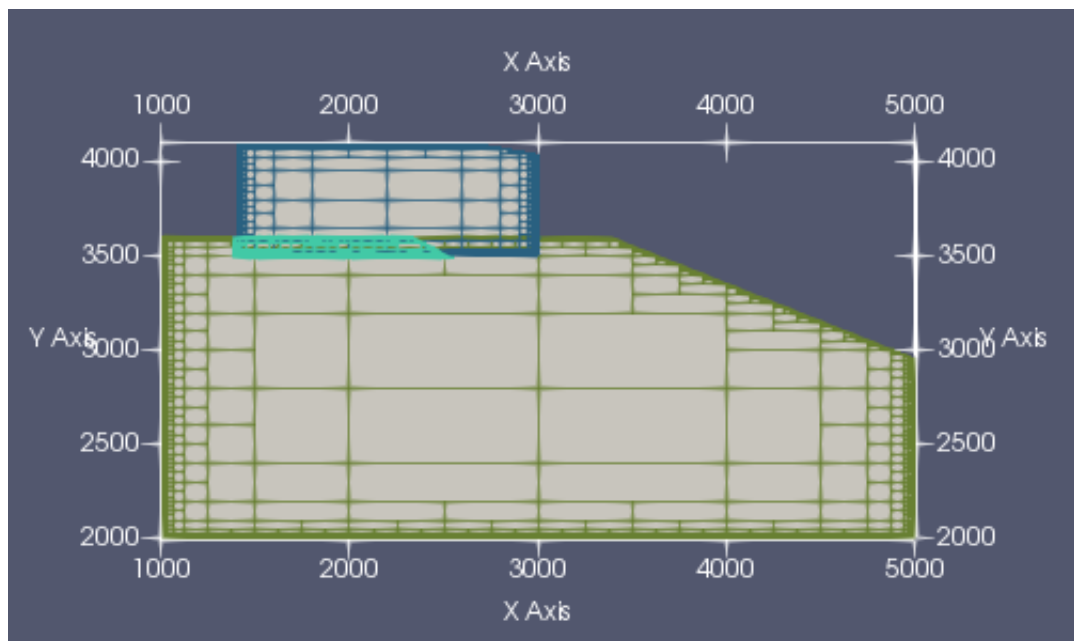


FIGURE 12 – Nouvelle intersection et résultat du calcul de viabilité

- Le(s) noyau(x) qui entraînent une intersection vide, dans le cas de trois utilisateurs ou plus.

On calcule l'intersection de noyaux deux à deux afin de déterminer quels noyaux bloquent l'intersection. Les utilisateurs dont les noyaux ont le plus d'intersections vides doivent relâcher leurs contraintes.

Ici, les utilisateurs 1 et 2 ont chacun une intersection vide, ils doivent donc chacun relâcher leurs contraintes. L'utilisateur 1 relâche ses contraintes comme précédemment. L'utilisateur 2 obtient le noyau suivant (en jaune) pour chaque jeu de paramètres :

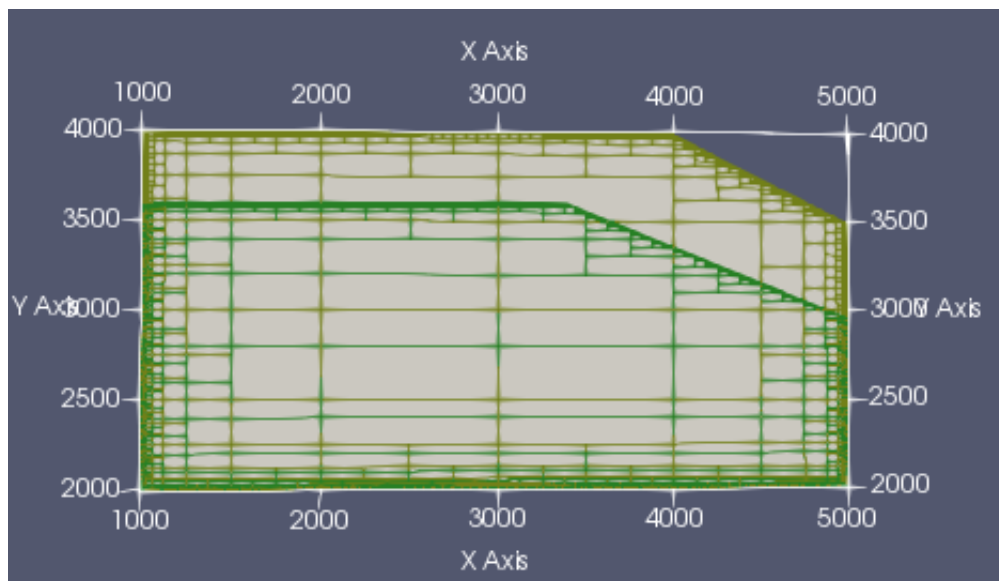


FIGURE 13 – Nouveau noyau de l'utilisateur 12, en comparaison avec l'ancien

On a donc bien une intersection non vide. On peut relancer un calcul de viabilité à l'intérieur de cette intersection, ce qui donne le noyau vert.

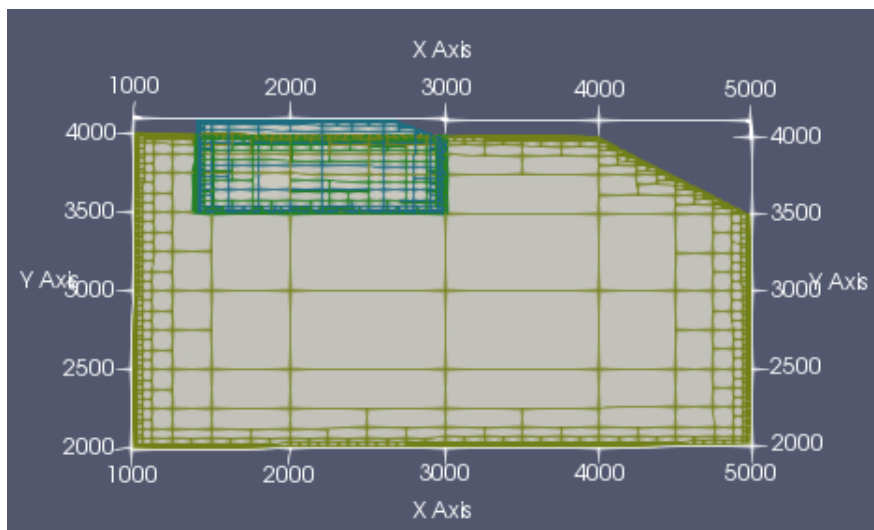


FIGURE 14 – Nouvelle intersection et résultat du calcul de viabilité

Cette méthode contraint donc deux utilisateurs à changer leurs contraintes mais permet aussi de retrouver un noyau final plus grand (et donc plus facile à contrôler) :

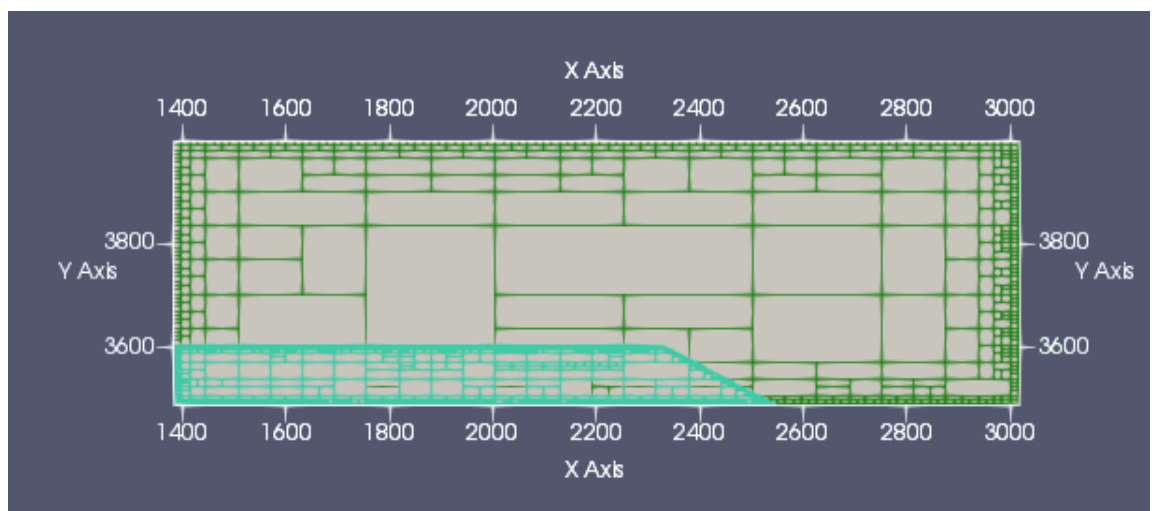


FIGURE 15 – Comparaison des noyaux de viabilité finaux obtenus avec les deux méthodes précédentes

— Les noyaux les plus éloignés de la moyenne.

Procédure : Déterminer le barycentre de chaque noyau et en faire la moyenne.

Les utilisateurs dont les noyaux *sont les plus éloignés* de ce point devront relâcher leurs contraintes.

Ici on obtient les barycentres suivants, avec en rouge leur moyenne :

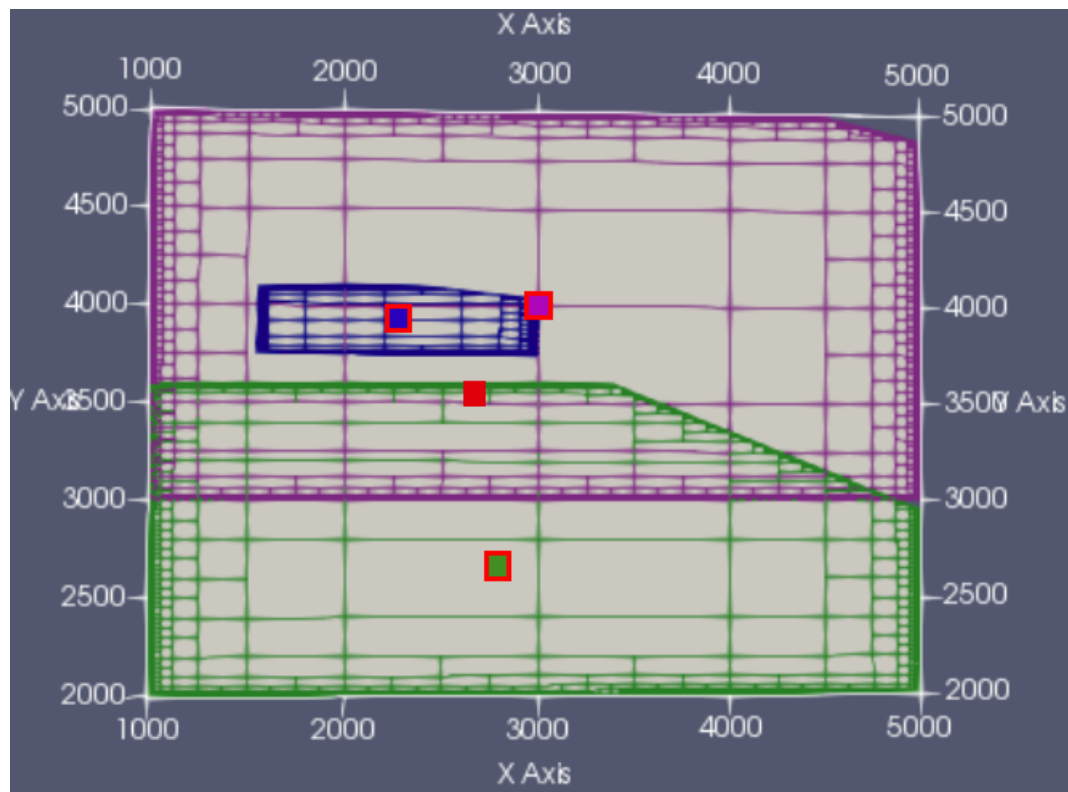


FIGURE 16 – Comparaison des noyaux de viabilité et calcul de leurs barycentres (encadrés en rouge) et de la moyenne des barycentres

S'il on choisit le noyau dont *la bordure* est le plus éloigné de la moyenne, il s'agit du noyau bleu (utilisateur 1) et l'on obtient le même cas de figure que pour la méthode du noyau le plus petit (Figure 11 et 12).

S'il on choisit de relaxer les contraintes du noyau dont *le barycentre* est le plus éloigné de la moyenne, il s'agit du noyaux vert (utilisateur 2). Sa relaxation nous donne le noyau de la figure 13.

— Système de négociations

Mettre en place un système de négociation entre les joueurs, en s'inspirant de l'existant (Simparc, [3]). Les joueurs se mettent donc d'accord entre eux pour désigner les utilisateurs qui devront reformuler leur problème de viabilité.

Une fois qu'un ou plusieurs noyau(x) ont été désignés comme ceux devant s'élargir, il est possible d'aider l'utilisateur à relâcher ses contraintes. Un calcul préalable à la session de jeu peut être effectué pour trouver des noyaux qui correspondent à chaque profil et qui fonctionnent bien ensemble.

Il est aussi possible d'utiliser la méthode proposée par Chandrasekaran [13] qui consiste à relâcher les contraintes dans un ordre précis : de celle qui agrandirait le plus les noyaux à celle qui l'agrandirait le moins. Cela nécessite cependant des données supplémentaires sur le modèle et sa dynamique.

Conclusion

Nous avons ainsi établi des dynamiques et programmé certaines des fonctionnalités de l'assistant. Cependant, bien que nous ayons exploré plusieurs possibilités de dynamiques (voir Annexe), les dynamiques que nous avons choisies n'ont pas encore été confrontées à la réalité et sont sujettes à améliorations. De plus les noyaux que nous avons calculés ne correspondent pas pour l'instant à des utilisateurs ayants des rôles et intérêts bien définis, bien que nous en ayons imaginés plusieurs (pêcheurs, agence touristique, anthropologues, environnementalistes)

Dans la suite du projet, nous aimerions définir ces rôles en détails afin d'établir plusieurs profils qui seraient intéressés par le domaine. De plus, nous aimerions continuer l'implémentation des fonctionnalités de l'assistant afin qu'il puisse trouver, parmi l'union des contrôles des utilisateurs, une politique qui satisfasse chacun des acteurs sociaux. Il serait aussi intéressant de combiner notre approche automatisée avec de la négociation entre les utilisateurs et notre approche mathématique avec de la simulation multi-agents.

Références

- [1] Alvarez, I., Reuillon, R., & de Aldama, R. (2016). Viabilitree : A kd-tree Framework for Viability-based Decision. Research report, LIP6, Paris. <https://hal.archives-ouvertes.fr/hal-01319738>
- [2] www.motive.cemagref.fr/vino
- [3] J.-P. Briot et al. (2017) Participatory management of protected areas for biodiversity conservation and social inclusion – Experience of the SimParc multi-agent based serious game. In D. Adamatti (Ed.), Multi-Agent Based Simulations Applied to Biological and Environmental Systems, IGI Global, pp. 299–333.
- [4] Wei, W., Alvarez, I., & Martin, S. (2013). Sustainability analysis : Viability concepts to consider transient and asymptotical dynamics in socio-ecological tourism-based systems. *Ecological Modelling*, 251, 103–113. doi :10.1016/j.ecolmodel.2012.10.009
- [5] Casagrandi, R. and S. Rinaldi. 2002. A theoretical approach to tourism sustainability. *Conservation Ecology* 6(1) : 13.
- [6] Jean-Pierre Aubin (1991). Viability Theory. Birkhauser. ISBN 0-8176-3571-8.
- [7] C. Le Page, N. Becub, P. Bommel, F. Bousquet (2012). Participatory agent-based simulation for renewable resource management : The role of the Cormas simulation platform to nurture a community of practice. *Journal of Artificial Societies and Social Simulation*, 15(1) :10
- [8] Martinez-Maldonado, R., Kay, J., & Yacef, K. (2013, July). An automatic approach for mining patterns of collaboration around an interactive tabletop. In *International Conference on Artificial Intelligence in Education* (pp. 101-110). Springer, Berlin, Heidelberg.
- [9] Katz, S., Albacete, P., Ford, M. J., Jordan, P., Lipschultz, M., Litman, D., ... & Wilson, C. (2013, July). Pilot test of a natural-language tutoring system for physics that simulates the highly interactive nature of human tutoring. In *International Conference on Artificial Intelligence in Education* (pp. 636-639). Springer, Berlin, Heidelberg.
- [10] Tai, M., Arroyo, I., & Woolf, B. P. (2013, July). Teammate relationships improve help-seeking behavior in an intelligent tutoring system. In *International Conference on Artificial Intelligence in Education* (pp. 239-248). Springer, Berlin, Heidelberg.
- [11] Yannier, N., Koedinger, K. R., & Hudson, S. E. (2013, July). Tangible collaborative learning with a mixed-reality game : Earthshake. In *International Conference on Artificial Intelligence in Education* (pp. 131-140). Springer, Berlin, Heidelberg.
- [12] Blessing, S. B., Skowronek, J. S., & Quintana, A. A. (2013, July). ExploreIT! An Adaptive Tutor in an Informal Learning Environment. In *International Conference on Artificial Intelligence in Education* (pp. 607-610). Springer, Berlin, Heidelberg.
- [13] Chandrasekaran, B. (1990). Design problem solving : A task analysis. *AI magazine*, 11(4), 59.
- [14] Finn, D. P., Grimson, J. B., Harty, N. M. (1992). An intelligent modelling assistant for preliminary analysis in design. In *Artificial Intelligence in Design'92* (pp. 579-596). Springer Netherlands.

4 Annexes

A Portraits de phase en 2D du modèle CAT

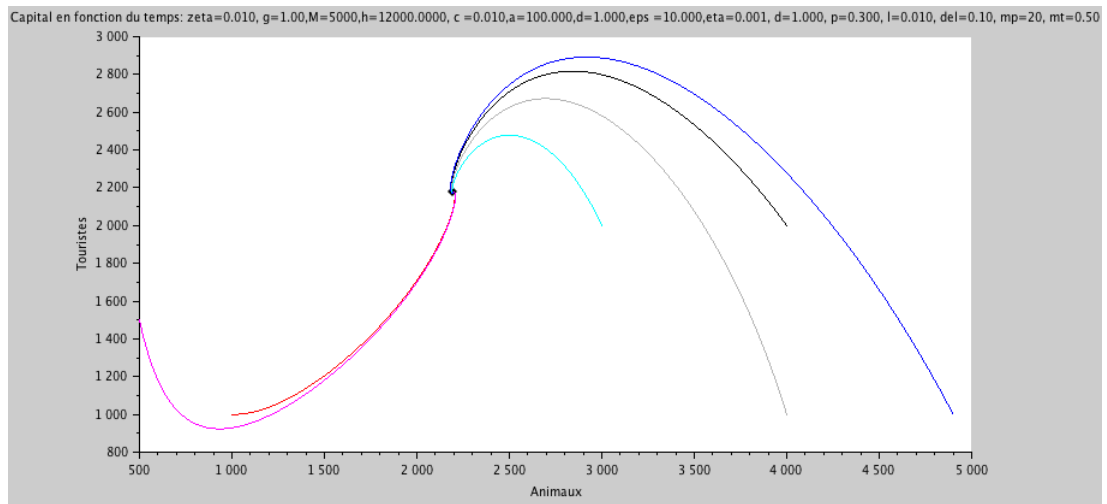


FIGURE 17 – Nombre d'animaux en fonction du nombre de touristes

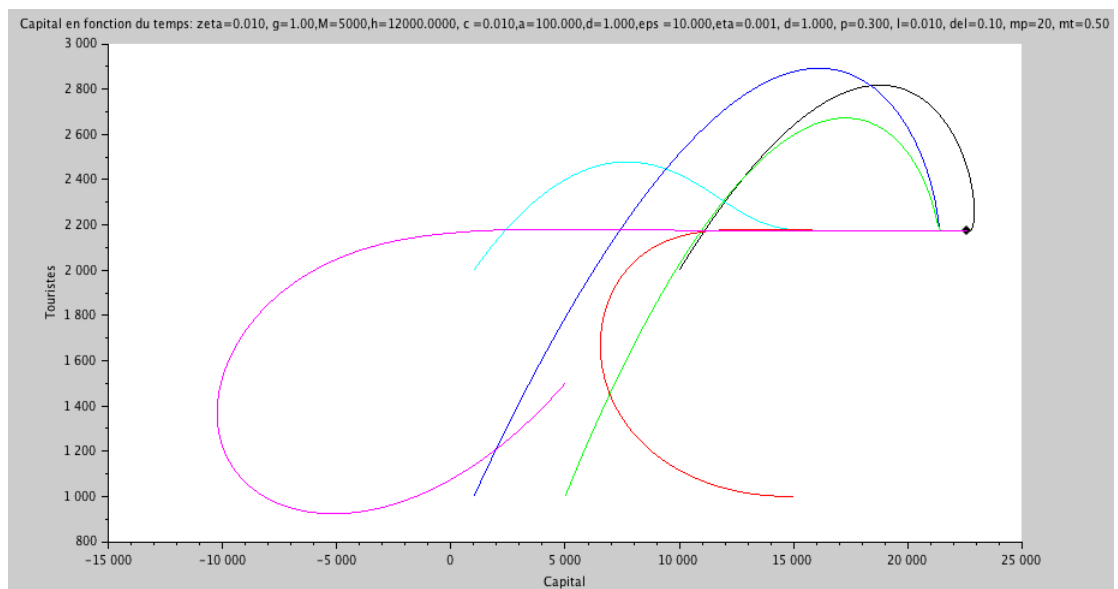


FIGURE 18 – Nombre de touristes en fonction du Capital

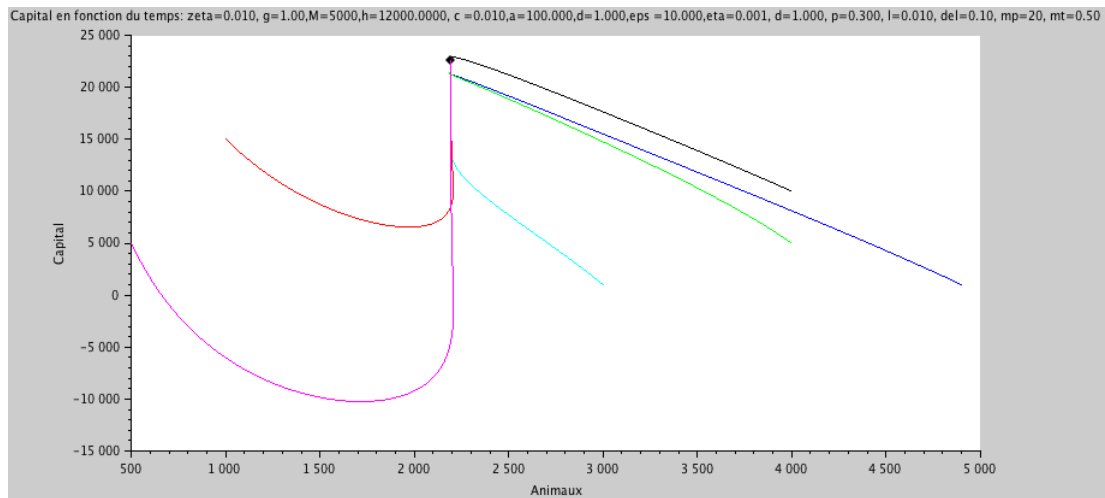


FIGURE 19 – Capital en fonction du nombre d’animaux

B Variations des valeurs de contrôles pour le modèle 2D

Paramètres : $M = 5000$, $c = 1$, $a = 1$, $\eta = 0.005$, $l = 0.0001$ Variation des paramètres ζ , ϵ ζ varie de 1 à 50 (en colonne) et ϵ de 0.1 à 100 (en ligne). On affiche l’équilibre pour A : le nombre d’animaux et T : le nombre de touristes.

	0.1	1	10	20	30	40	50	60	70	80	90	100
1	A : 947 T : 937	A : 1228 T : 1218	A : 2379 T : 2369	A : 2914 T : 2904	A : 3236 T : 3226	A : 3459 T : 3449	A : 3625 T : 3615	A : 3754 T : 3744	A : 3859 T : 3849	A : 3944 T : 3934	A : 4016 T : 4006	A : 4078 T : 4068
2	A : 686 T : 1362	A : 898 T : 1787	A : 1819 T : 3629	A : 2287 T : 4564	A : 2587 T : 5165	A : 2805 T : 5600	A : 2973 T : 5937	A : 3108 T : 6207	A : 3220 T : 6431	A : 3315 T : 6620	A : 3396 T : 6782	A : 3466 T : 6923
3	A : 564 T : 1683	A : 740 T : 2212	A : 1519 T : 4548	A : 1925 T : 5766	A : 2190 T : 6562	A : 2385 T : 7147	A : 2538 T : 7605	A : 2662 T : 7977	A : 2765 T : 8287	A : 2853 T : 8551	A : 2929 T : 8779	A : 2996 T : 8978
4	A : 489 T : 1946	A : 642 T : 2559	A : 1319 T : 5266	A : 1672 T : 6681	A : 1904 T : 7607	A : 2074 T : 8289	A : 2208 T : 8824	A : 2317 T : 9258	A : 2407 T : 9621	A : 2485 T : 9930	A : 2551 T : 10197	A : 2610 T : 10430
5	A : 436 T : 2170	A : 572 T : 2851	A : 1169 T : 5838	A : 1478 T : 7384	A : 1679 T : 8389	A : 1827 T : 9126	A : 1942 T : 9700	A : 2035 T : 10165	A : 2112 T : 10552	A : 2178 T : 10880	A : 2234 T : 11164	A : 2284 T : 11411
6	A : 396 T : 2366	A : 518 T : 3102	A : 1050 T : 6295	A : 1321 T : 7920	A : 1495 T : 8962	A : 1621 T : 9719	A : 1719 T : 10304	A : 1797 T : 10774	A : 1862 T : 11164	A : 1917 T : 11492	A : 1964 T : 11774	A : 2004 T : 12019
7	A : 364 T : 2539	A : 475 T : 3321	A : 952 T : 6660	A : 1189 T : 8318	A : 1339 T : 9365	A : 1446 T : 10115	A : 1528 T : 10689	A : 1593 T : 11147	A : 1647 T : 11523	A : 1692 T : 11837	A : 1730 T : 12106	A : 1764 T : 12338
8	A : 337 T : 2692	A : 440 T : 3512	A : 869 T : 6945	A : 1076 T : 8602	A : 1204 T : 9628	A : 1295 T : 10352	A : 1363 T : 10900	A : 1417 T : 11332	A : 1461 T : 11683	A : 1498 T : 11975	A : 1529 T : 12223	A : 1555 T : 12436
9	A : 315 T : 2830	A : 409 T : 3679	A : 796 T : 7162	A : 977 T : 8789	A : 1087 T : 9776	A : 1163 T : 10460	A : 1220 T : 10971	A : 1264 T : 11369	A : 1300 T : 11690	A : 1329 T : 11955	A : 1354 T : 12178	A : 1375 T : 12369
10	A : 296 T : 2953	A : 383 T : 3825	A : 733 T : 7320	A : 890 T : 8896	A : 983 T : 9829	A : 1047 T : 10465	A : 1094 T : 10933	A : 1130 T : 11294	A : 1159 T : 11583	A : 1182 T : 11819	A : 1202 T : 12016	A : 1219 T : 12183
20	A : 182 T : 3635	A : 225 T : 4496	A : 352 T : 7040	A : 389 T : 7779	A : 406 T : 8122	A : 416 T : 8321	A : 423 T : 8452	A : 427 T : 8545	A : 431 T : 8614	A : 433 T : 8668	A : 436 T : 8710	A : 437 T : 8745
30	A : 124 T : 3715	A : 145 T : 4360	A : 191 T : 5743	A : 200 T : 6015	A : 204 T : 6124	A : 206 T : 6183	A : 207 T : 6220	A : 208 T : 6246	A : 209 T : 6264	A : 209 T : 6278	A : 209 T : 6289	A : 210 T : 6298
40	A : 88 T : 3511	A : 98 T : 3933	A : 116 T : 4638	A : 118 T : 4746	A : 119 T : 4787	A : 120 T : 4808	A : 120 T : 4822	A : 121 T : 4831	A : 121 T : 4837	A : 121 T : 4842	A : 121 T : 4846	A : 121 T : 4849
50	A : 64 T : 3202	A : 69 T : 3464	A : 76 T : 3831	A : 77 T : 3880	A : 78 T : 3898	A : 78 T : 3907	A : 78 T : 3913	A : 78 T : 3917	A : 78 T : 3919	A : 78 T : 3921	A : 78 T : 3923	A : 78 T : 3924

C Autres systèmes envisagés

C.0.1 Modèles 3D : Animaux, Touristes, Environnement

L'environnement sans touriste suit une dynamique de Verhulst et est impacté par les touristes (suivant Casagrandi et Rinaldi, 2002 [5]). Il est possible de restaurer l'environnement par un effort u .

Hypothèse sur u : u a le même effet qu'une taxe de séjour et prend donc la forme $\epsilon T(t)$ ou $(u_0 + \epsilon T(t))$. u_0 est un contrôle binaire, c'est par exemple l'effort qu'il faut effectuer pour restaurer l'environnement sans touriste en 100 ans ou 0.

u est le contrôle pour les modèles suivants :

C.0.2 Premier modèle

$$\begin{aligned}\frac{dA}{dt} &= A(t)g(1 - \frac{A(t)}{K_A(t)}) - hT(t)A(t) \\ \frac{dE}{dt} &= E(t)r(1 - \frac{E(t)}{K}) - sT(t)E(t) + \epsilon T(t)(K - E(t)) \\ \frac{dT}{dt} &= T(t)(aA(t) + bE(t) - cT(t) - d)\end{aligned}$$

Avec $K_A = 1 + M\frac{E}{K}$.

C.0.3 Deuxième modèle

On remplace d par $d_0(1 - \frac{\delta}{1+T})$ dans $\frac{dT}{dt}$, à la manière de Casagrandi et Rinaldi [5].

$$\begin{aligned}\frac{dA}{dt} &= A(t)g(1 - \frac{A(t)}{K_A(t)}) - hT(t)A(t) \\ \frac{dE}{dt} &= E(t)r(1 - \frac{E(t)}{K}) - sT(t)E(t) + \epsilon T(t)(K - E(t)) \\ \frac{dT}{dt} &= T(t)(aA(t) + bE(t) - cT(t) - d_0(1 - \frac{\delta}{1+T}))\end{aligned}$$

Avec $K_A = 1 + M\frac{E}{K}$.

C.0.4 Troisième modèle

On remplace $aA(t)$ par $\frac{A(t)}{A+\phi_A}$ dans $\frac{dT}{dt}$, à la manière de Casagrandi et Rinaldi [5].

$$\begin{aligned}\frac{dA}{dt} &= A(t)g(1 - \frac{A(t)}{K_A(t)}) - hT(t)A(t) \\ \frac{dE}{dt} &= E(t)r(1 - \frac{E(t)}{K}) - sT(t)E(t) + \epsilon T(t)(K - E(t))\end{aligned}$$

$$\frac{dT}{dt} = T(t)(a\frac{A(t)}{A + \phi_A} + bE(t) - cT(t) - d)$$

Avec $K_A = 1 + M\frac{E}{K}$.

C.0.5 Quatrième modèle

On remplace $\epsilon T(t)$ par $(u_0 + \epsilon T(t))$ dans $\frac{dE}{dt}$.

$$\frac{dA}{dt} = A(t)g(1 - \frac{A(t)}{K_A(t)}) - hT(t)A(t)$$

$$\frac{dE}{dt} = E(t)r(1 - \frac{E(t)}{K}) - sT(t)E(t) + (u_0 + \epsilon T(t))(K - E(t))$$

$$\frac{dT}{dt} = T(t)(a\frac{A(t)}{A + \phi_A} + bE(t) - cT(t) - d_0(1 - \frac{\delta}{1 + T}))$$

Avec $K_A = 1 + M\frac{E}{K}$.

C.1 Modèles 2D : Version avec d, ϵ comme contrôle

$$\frac{dA}{dt} = A(t)g(1 - \frac{A(t)}{1 + \frac{M}{K_\epsilon}}) - hT(t)A(t)$$

$$\frac{dT}{dt} = T(t)(aA(t) - cT(t) - d)$$

Avec $K_\epsilon = \frac{\eta T}{\epsilon + 1}$.