

Rapport de projet de semestre - Automne 2023

Eugène Bergeron, pour la Chair Of Statistical Field Theory, EPFL.

Supervisé par Vassilis Papadopoulos, dirigé par Clément Hongler.

Abstract

Dans ce projet, nous avons exploré la simulation de mécanismes de la biologie à l'aide d'un modèle d'automate cellulaire: le BIO-LGCA. Grâce à ce modèle, nous avons simulé une version très simplifiée de la reproduction par ARN. Nous avons également démontré qu'il était à la fois possible d'utiliser ce modèle pour faire des simulations de dynamique moléculaire et des automates cellulaires plus traditionnels.

Code disponible sur https://github.com/Asbyx/Alife_bachelor_project.

Motivations

Le domaine de la vie artificielle est un domaine très vaste, comprenant des algorithmes adaptés de la vie biologique, comme les algorithmes génétiques ou les réseaux de neurones, des simulations de comportements biologiques ou l'étude de l'émergence de la vie dans des simulations.

Une branche de cette discipline consiste à étudier l'émergence de patterns et de comportements complexes dans des environnements régis par des règles simples. L'exemple le plus courant et la base de nombreux autres modèles est le Jeu de la Vie de John Conway, créé en 1970. Ce modèle est une simple grille dont chaque case est soit allumée ou éteinte, et dont le prochain état est déterminé par les cases qui l'entourent. 4 règles régissent cet environnement mais elles peuvent être simplifiées en 1 fonction logique du nombre de voisins (n) et de l'état actuel (s_i): $s_{i+1} = 1 \Rightarrow n = 3 \text{ ou } (s_i = 1 \text{ et } n = 2)$. De cette fonction simple, on peut constater l'émergence d'ensembles de cellules, appelés structures, qui ont des comportements spécifiques et non chaotiques. Ces structures peuvent avoir plusieurs types de comportement: stable, oscillatoire, mouvant, etc. À partir de ces structures, on peut démontrer que ce modèle est Turing complet.

Il existe d'autres modèles que celui du Jeu de la Vie, et la plupart sont connus sous le nom "d'automates cellulaires". Ce sont des modèles mathématiques qui consistent en une grille dont chaque cellule contient un état parmi un ensemble fini d'états (dans le cas du Jeu de la Vie, ces états sont simplement "allumé" et "éteint"). L'état d'une cellule est une fonction de son état précédent et des états des cellules voisines. Leur origine remonte à un travail de John Von Neumann qui en créa un pour créer un automate capable de se répliquer à l'infini.

Un autre type de modèle intéressant dans l'étude de la vie artificielle est celui de la simulation de particules, ou dynamique moléculaire. Ce type d'environnements est plus proche de ce qu'on pourrait trouver dans la réalité car ce sont des modèles "continus" (dans la mesure de ce qui est possible informatiquement), à la différence des automates cellulaires qui sont totalement discrets. De plus, ce sont des modèles qui conservent la quantité de matière présente dans le système, là où les automates cellulaires peuvent faire apparaître ou disparaître de la matière selon les règles du modèle. Ces modèles consistent en plusieurs types de particules, attirées ou repoussées les unes par rapport aux autres. De

ces lois d'interaction émerge alors des structures parfois constituées de milliers de particules qui forment un comportement complexe, un peu comme les atomes dans une cellule. Un exemple très explicite du potentiel de ce genre de modèle est le travail de Tom Mohr et son [Particle Life^{\[1\]}](#), dont il fait une démonstration dans [cette vidéo^{\[2\]}](#).

Enfin, à mi-chemin entre les automates cellulaires et la simulation de particules, on trouve le sujet du projet de recherche: BIO-LGCA, dont nous reviendrons juste après sur son fonctionnement. L'intérêt de ce modèle réside dans le fait que c'est un automate cellulaire qui emprunte certaines caractéristiques de la dynamique moléculaire, ouvrant la porte à des simulations simples mais pouvant être facilement associées à de réelles simulations chimiques. Ce modèle a été utilisé pour modéliser des comportements de cellules, notamment leur migration collective et leur adhésivité les unes aux autres, dans le [papier original^{\[3\]}](#) du modèle et dans [ce papier^{\[4\]}](#), mais nous voulions, dans le cadre du projet de recherche, explorer des aspects un peu plus fondamentaux du modèle et comprendre dans quel mesure il était possible de simuler des comportements biologiques classiques à l'aide de ce modèle.

BIO-LGCA

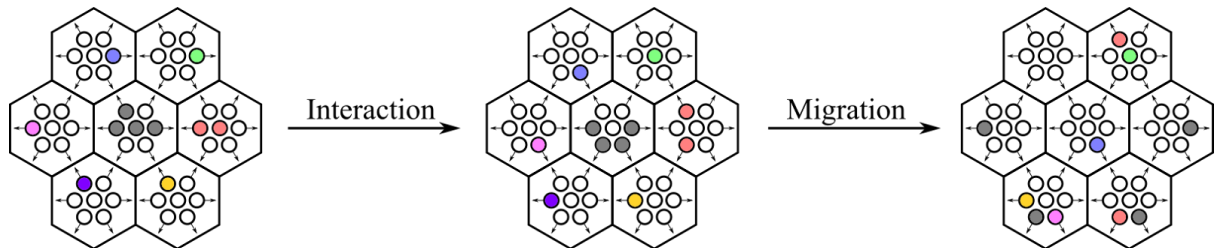
BIO-LGCA (Biological Lattice-Gas Cellular Automata) est un dérivé du Lattice-Gas Cellular Automaton (LGCA). Le LGCA est un automate cellulaire (A.C. pour la suite du rapport) qui conserve la quantité de mouvement du système. Chaque étape de la simulation se fait en 2 temps: propagation puis collision. Lors de la propagation, chaque particule (représentée par un état "allumé" qui contient également une direction, et éventuellement une vitesse) est déplacée dans sa case de destination. Chaque cellule de la grille du LGCA peut donc avoir 5 états, que l'on peut représenter en une séquence de 4 bits, chacun des bits représentant la présence d'une particule allant dans une des 4 directions (par exemple, si on encode les directions telles que la séquence définit les directions haut-bas-droite-gauche, la séquence 1-0-1-0 indique que cette case contient une particule qui bouge vers le haut et une autre vers la droite). Pour les collisions, il n'y en a que 2 possibles qui permettent de conserver la quantité de mouvement: haut+bas \rightarrow droite+gauche, et inversement, car ces deux collisions ont une somme des quantités de mouvement nulle. Vous trouverez un exemple de simulation dans le fichier "lgca.mkv".

Sémantique du BIO-LGCA

- **Lattice:** équivalent d'une case des grilles des A.C. Ce sont les éléments de l'ensemble des espaces occupables par les particules.
- **Particules:** éléments qui occupent les lattices, il peut y en avoir de plusieurs types. Les particules sont le fondement de l'étape de collision, appelée **l'interaction**.
- **Channels:** chaque lattice est composé de plusieurs channels, dans lesquels se placent les particules. Ces channels encodent également la vitesse et la direction de la particule qui les occupe. On sépare les "rest" channels, qui contiennent les particules qui ne bougeront pas à l'étape de migration, des "communication" channels, qui contiennent les particules qui bougeront vers un lattice voisin. On peut

déclarer autant de rest channels et autant de communication channels par voisin que nécessaire.

- **Étape de migration:** équivalent à l'étape de propagation. C'est l'étape où les particules sont transmises de lattice en lattice, selon le channel qu'elles occupaient.
- **Étape d'interaction:** équivalent à l'étape de collision. C'est la fonction qui prend en entrée toutes les particules du lattice et leur channel, et attribue à chaque channel du lattice une nouvelle particule (qui peut être la même que précédemment).



Nous avons dans le cadre du projet de recherche, implémenté une version légèrement différente du BIO-LGCA décrit dans le papier de Deutsch et al^[3]. Cependant, notre version est incluse dans la définition générale donnée par le papier. C'est pourquoi à partir d'ici, lorsque nous parlerons de BIO-LGCA, nous parlerons de la version sur laquelle nous avons travaillé et non celle décrite dans le papier. La liste des différences est donnée en Annexe.

BIO-LGCA reprend les mêmes fondements que LGCA, c'est-à-dire les 2 étapes de simulation, mais allège 2 contraintes: le type fixé des particules et la conservation de la quantité de mouvement. En effet, le BIO-LGCA ne se limite pas à 1 type de particule possible: il est possible d'en définir autant que l'on souhaite et dont on a besoin pour la simulation. Enfin, la fonction de collision est assouplie car cette fonction opère après chaque étape de migration et sur chaque lattice, sa définition est totalement libre, et elle permet la création, le mouvement ou la destruction de particules (pouvant ou non casser le principe de conservation de la masse du système). De plus, le BIO-LGCA se démarque des autres A.C. car la fonction de collision ne prend plus en compte les lattices voisins, mais seulement celui sur lequel elle agit.

Particularités et intérêts des modèles de BIO-LGCA

Le cœur d'un BIO-LGCA est donc sa fonction d'interaction, une fonction qui part d'un lattice et donne un nouveau lattice, et qui est appliquée sur tous les lattices entre chaque étape de migration. Selon cette fonction, on peut ou non définir un modèle qui respecte les principes de conservation de quantité de mouvement, de masse ou d'énergie. Elle permet également de faire des réactions chimiques très proches de la réalité. Imaginons que l'on représente des atomes par les particules de notre modèle, on pourrait définir la fonction d'interaction telle que 2 particules "H" et une particule "O" forment une nouvelle particule "H₂O" lorsqu'elles se retrouvent dans le même lattice. On a donc dans ce cas là une simulation de particules dans un espace discret. Il est également possible de simuler des automates cellulaires tout à fait classiques, comme celui du Jeu de la Vie (nous y reviendrons). Enfin, la fonction d'interaction peut être totalement déterministe ou non.

Une des plus grandes forces du BIO-LGCA selon moi est que l'on peut avoir plusieurs niveaux d'abstraction et d'interprétation de ce qu'on simule à travers un modèle de BIO-LGCA. On peut partir de l'infiniment petit, en considérant que les particules sont des

atomes qui interagissent entre eux dans des espaces clos, représentés par les lattices. On peut également voir les particules comme des protéines, contenues dans les lattices qui représentent alors des cellules. Les cellules peuvent alors s'échanger des signaux ou des protéines via les communication channels et également procéder à des réaction chimiques internes. Mais il est également possible de considérer, comme le papier original proposant BIO-LGCA, chaque particule comme un cellule qui bouge au sein d'un système plus grand. En somme, on peut définir plus ou moins ce qu'on souhaite car la fonction d'interaction autorise tous les comportements possibles.

Recherches

Le projet de recherche en tant que tel a donc été d'explorer la simulation de différents types de comportement possibles dans des modèles de BIO-LGCA à travers la définition des fonctions d'interaction. Notre objectif était donc de simuler des comportements trouvables dans la biologie classique, notamment la reproduction par ARN. Nous avons créé plusieurs modèles que nous décrivons ici.

Afin de créer facilement ces modèles, nous avons créé un framework dans lequel il suffit de programmer une fonction d'interaction, qui prend le monde actuel en entrée, et donne en sortie le monde après l'étape de migration. Couplé à la définition d'une initialisation du modèle, ce framework nous a permis de nous concentrer sur les fonctions d'interaction, plutôt que d'autres détails techniques. Une fois la fonction d'interaction programmée, le framework se charge de 100% de la simulation.

Tweaked LGCA

Nous avons fait un premier modèle pour utiliser une première fois le BIO-LGCA, et pour cela nous avons implémenté une version modifiée du LGCA.

Ce modèle n'a qu'un seul type de particule. La fonction d'interaction est également simple:

- Si une particule en mouvement (dans un communication channel) rencontre une particule au repos (dans un rest channel), elle rebondit et repart dans le sens opposé (les particules au repos agissent comme des murs qui font rebondir les particules en mouvement).
- Si deux particules de mouvement opposé se rencontrent, la somme de leur quantité de mouvement étant 0, elles deviennent une particule au repos.

Un exemple de simulation est disponible dans le fichier "bio_lgca_tweaked_lgca.mkv".

Depth Aware Lattices

Une fois fait le premier modèle implémenter pour prendre en main le BIO-LGCA, Vassilis a suggéré d'essayer d'implémenter un comportement dans lequel les cellules (lattices dans notre cas), seraient soit vivantes soit mortes, et si elles sont vivantes, qu'elles puissent déterminer leur niveau de profondeur dans l'organisme dans lequel elles sont. On définit un

organisme par un simple regroupement de cellules, et la profondeur telle que: les cellules du bord de l'organisme ont une profondeur de 0 et les cellules au centre ont une profondeur maximale.

Ce comportement est intéressant car il permet d'ouvrir la voie vers une reproduction par ARN. En effet, la façon dont les cellules déterminent leur niveau de profondeur est via des communications entre les cellules. Ces communications peuvent être à terme utilisées pour informer les lattices autour de l'état du lattice communicant, par exemple de si celui-ci a trouvé un acide complémentaire qui permet de construire la chaîne d'ARN.

L'algorithme en lui-même est simple: toutes les cellules transmettent à chaque étape la profondeur qu'elles pensent avoir. Ensuite, le traitement des signaux reçus est le suivant:

- Si un des communication channel ne contient rien, c'est que la cellule touche de l'air, donc qu'elle se situe au bord de l'organisme. Sa profondeur est donc 1.
- Sinon, sa profondeur est le minimum des profondeurs reçues des autres cellules +1.

Il faut donc transcrire cet algorithme en une fonction d'interaction. Pour cela, il faut définir les états possibles. Pour les particules, nous avons défini que les particules pouvaient être n'importe quel nombre entier positif. Cela permet d'envoyer aux lattices voisins directement la profondeur estimée. Les cellules mortes enverraient donc simplement des 0 à leurs voisins. Nous avons donc eu besoin de 1 seul rest channel, qui contient la profondeur estimée du lattice (si c'est une cellule morte, cette profondeur doit rester 0), et un seul communication channel par voisin, qui transmettra cette valeur à ce voisin. Dans notre simulation, les particules sont donc simplement des signaux que se transmettent les cellules voisines et qui contiennent de l'information.

L'implémentation suit ensuite facilement l'algorithme, il faut juste faire attention que les cellules mortes le restent et ne soient pas influencées par les cellules vivantes qui transmettent leur estimation. Vous trouverez un exemple de simulation dans "bio_lgca_depth_aware.mkv".

Naive seed

Dans la continuité d'imiter le vivant, nous avons eu pour idée d'implémenter un mécanisme d'œuf. Le but serait de placer un œuf dans un rest channel, puis que celui-ci crée un organisme multicellulaire, aka sur plusieurs lattices. Cela ouvrirait la porte à la naissance d'organismes qui grandiraient en fonction de l'ADN qui les caractérise. ADN qui serait donc simplement contenu dans les rest channels.

Pour ce comportement, nous avons fait dans la continuité du précédent modèle: on donne la profondeur désirée d'un organisme à une cellule, qui crée les cellules autour d'elle, et ce récursivement jusqu'à ce que la profondeur de l'organisme soit correcte. L'algorithme est encore une fois assez simple:

- Chaque cellule vivante envoie autour d'elle sa profondeur -1.
- Un lattice d'air recevant une valeur devient une cellule vivante de profondeur égale à la valeur reçue.
- Si une cellule vivante reçoit une valeur supérieure à sa profondeur actuelle, elle met à jour sa profondeur en fonction de la nouvelle valeur reçue.

Fonction d'interaction: pareil au modèle précédent, on considère toutes les particules comme des signaux contenant de l'information, cette information étant un nombre entier positif.

Implémenté naïvement, ce comportement grandit en un carré tourné de 90°. Nous avons rapidement essayé de faire en sorte que le carré ne soit pas tournée de 90°, mais l'ADN donnant la forme étant une simple valeur et la profondeur calculée étant celle de Manhattan ($distance = (x_a - x_b) + (y_a - y_b)$), cela rend compliqué de générer d'autres formes. Mais avec une fonction de croissance qui se baserait sur la position de la cellule dans l'organisme désiré, dont toutes les informations seraient contenues dans l'ADN, il est normalement possible de faire grandir un organisme de forme quelconque. Autrement dit: une cellule de l'organisme déjà grandie crée des cellules autour d'elle en leur donnant la position qu'elles occupent dans l'organisme. Ainsi elles savent quels lattices de leur voisinage elles doivent activer ou non. Vous trouverez à nouveau un exemple de simulation dans le fichier "bio_lgca_naive_seed.mkv"

Moving lattices

Avec des lattices qui peuvent communiquer et grandir à partir d'une séquence ADN contenue dans un œuf, il ne nous manquait que le déplacement d'organismes pour pouvoir implémenter la reproduction par ARN.

Ici, l'algorithme est un peu plus compliqué: bien que le déplacement de particules est nativement présent dans le BIO-LGCA, "déplacer un lattice", ce qui revient à faire en sorte que toutes les particules présentes dans un rest channel soient transférées dans un autre lattice, est un autre défi. La solution triviale consisterait à faire en sorte qu'il y ait autant de communication channels par voisin que de de rest channels, puis que lors d'un déplacement, toutes les particules des rest channels soient transférées dans les bons communication channels. Une fois l'étape de migration effectuée, les particules seraient alors dans le communication channel du nouveau lattice et il ne reste plus qu'à les transférer dans leur rest channels.

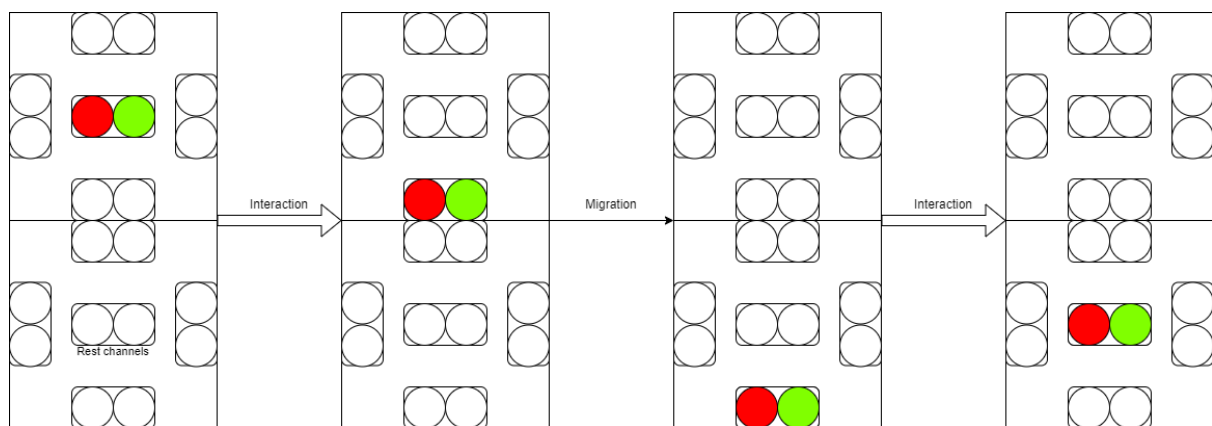


Illustration de l'algorithme naïf de déplacement de lattice

Pour des raisons techniques demandant de modifier le framework pour qu'il supporte plusieurs communication channels par voisin, nous avons préféré une autre solution se basant sur l'œuf que nous avons précédemment implémenté. Au moment d'un déplacement, les rest channels sont compressés en une seule particule œuf, qui contient toutes les informations nécessaires à la reconstruction des channels. Cette particule est

envoyée dans un communication channel, se déplace avec la migration step, puis est décompressée dans le nouveau lattice. Les particules des anciens rest channels sont simplement effacées. Dans notre cas, l'ADN d'un lattice consiste simplement en son type (si c'est un lattice d'air (valeur = 0), en mouvement (1) ou arrêté (2)) et de sa direction de mouvement. L'oeuf envoyé est donc $10 + direction$, avec la direction étant un nombre dans $[0, 3]$.

La dernière chose à faire était d'éviter les collisions entre les lattices en mouvement. Pour cela, un signal "réservation" (i.e une particule dont la fonction est de transmettre de l'information, dans notre cas de déclarer une réservation) a été ajouté à la simulation. Si un lattice reçoit un signal de réservation, il renvoie un signal "stop", qui a pour fonction d'arrêter le lattice en mouvement qui souhaitait se déplacer. Ainsi, un lattice qui reçoit un signal "stop" s'arrête (n'essaye plus de bouger). Un lattice d'air (avec aucune particule dans ses rest channels) qui reçoit une réservation renvoie un signal "move". Cela permet aussi de faire en sorte que 2 lattices ne collisionnent pas en voulant aller au même endroit: si un lattice d'air reçoit 2 réservations, il envoie un signal "move" à l'un des lattices, et un "stop" à l'autre.

Nous avons également rajouté une possibilité aléatoire pour un lattice de changer de direction. Un exemple de simulation est trouvable dans la vidéo "bio_lgca_moving_lattices.mkv".

Reproducing pairs

Ayant réussi à implémenter les modèles précédents avec succès, la prochaine étape était donc de tenter de simuler une reproduction par ARN avec BIO-LGCA. Dans notre simulation, les organismes ne peuvent être constitués que de 2 lattices, car nous avons privilégié le mouvement des lattices par rapport à la complexité des organismes.

Ainsi notre simulation se présente comme suit: le monde est initié aléatoirement avec des lattices en mouvement. Si deux lattices se percutent, ils forment un organisme (une paire). Un organisme est statique et attrape les lattices en mouvement qui passent à leur portée. Une fois que les 2 lattices de l'organisme ont attrapé chacun un lattice, les 2 lattices attrapés forment donc un nouvel organisme. Ce nouvel organisme est relâché, se déplace de quelques cases, puis commence à son tour à se reproduire. La seule particularité est que si un organisme nouvellement relâché rencontre un organisme en reproduction sur son chemin, celui-ci tue le nouveau organisme dont les lattices sont séparés et redeviennent de simples lattices en mouvement.

L'algorithme est considérablement plus compliqué, c'est pourquoi nous ne l'explorerons qu'en surface ici.

- Pour attraper un lattice en mouvement, les lattices d'un organisme envoient en permanence un signal de capture qui arrête un lattice en mouvement. Celui-ci répond qu'il a bien été attrapé avec un autre signal. Quand un lattice de l'organisme reçoit ce signal, indiquant que la capture a été réussie, il prévient sa paire de l'emplacement de sa capture. Ainsi la paire n'essaie d'attraper un lattice que du côté où un premier lattice a été attrapé.

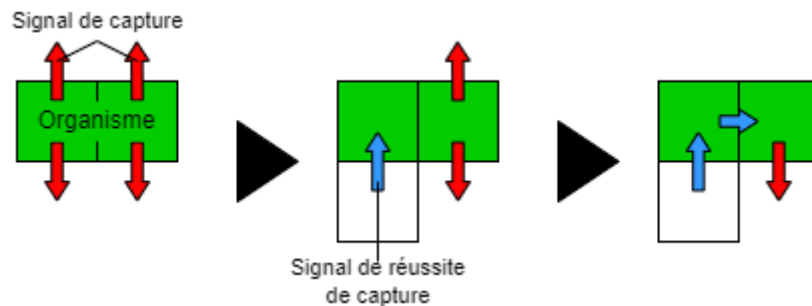


Illustration des signaux transmis lors d'une capture

- Le déplacement des organismes a été le plus compliqué à implémenter. Même s'il se base sur le même système de réservation que les lattices simples en mouvement, des problèmes liés au fait que les deux lattices doivent bouger à la même étape de migration, afin qu'ils ne se désynchronisent pas. Pour cela, nous avons ajouté un rest channel dédié à de la mémoire. En effet, les deux lattices ne peuvent bouger que s'ils reçoivent un signal de feu vert de la part de leur paire, et qu'eux-mêmes aient déjà prévenu leur paire qu'ils peuvent bouger. La mémoire sert donc à se souvenir que le lattice a prévenu sa paire qu'il peut bouger. Nous avons aussi fait en sorte qu'une réservation d'organisme a la priorité sur une réservation de lattice simple. Dans ce cas là, le lattice simple attend simplement que l'organisme soit passé pour continuer sa route.
- Nous avons également ajouté un temps de récupération, une fois qu'un organisme s'est reproduit, durant lequel un organisme ne peut pas capturer de lattice. Nous l'avons ajouté à l'aide d'un simple compteur dans un rest channel.

Sans différenciation des types de cellules

Dans la première implémentation que nous avons réalisée, les lattices n'étaient pas différenciés: les organismes captureraient n'importe quel lattice en mouvement qui passait à côté d'eux. Cela a permis l'émergence de comportements que nous n'avions pas anticipés. vous trouverez dans la vidéo "bio_lgca_reproductible_pairs_clocks.mkv" un exemple de ce qu'on pourrait apparenter à une clock, qui se fait stopper au bout d'un moment par un autre clock qui se fait elle même mettre en pause.

Les clocks fonctionnent comme suit: un organisme se reproduit, et son enfant se place un peu plus loin. Ensuite, si seulement un des deux organismes se reproduit avec son enfant dans la direction de l'autre organisme, l'enfant va osciller entre les deux organismes. En effet, l'enfant atteint un organisme, se fait tuer par l'organisme, mais celui-ci récupère immédiatement les 2 lattices de l'enfant pour former un nouvel enfant qu'il renvoie dans la direction opposée. Pour arrêter la clock, il suffit qu'entre le moment où l'organisme renvoie l'enfant et qu'il le récupère, il capture un lattice de l'autre côté. Ainsi, l'enfant ne se fait pas immédiatement tuer par l'organisme et reste à attendre. Une fois que l'organisme à terminer son enfant de l'autre côté, l'enfant qui attendait se fait tuer, et la clock est donc perdue.

Faute de temps nous n'avons pas eu le temps d'explorer plus en profondeur ce modèle, mais je suis certain qu'il est possible de démontrer qu'il est Turing complete.

Avec différenciation des types de cellules

L'étape finale du projet consistait à rendre la reproduction des organismes plus proche de ce qu'on trouve dans une reproduction par ARN, c'est-à-dire que les lattices représentent une nucléotide (A, U, G ou C). Ainsi, les lattices dans un organisme ne pourrait attraper qu'un lattice compatible. Comme cela divise par 4 les chances d'une reproduction, nous avons décidé qu'au bout d'un certain moment, si un organisme ne s'est pas reproduit, il meurt et est dissout. Cela permet de renouveler les lattices en mouvement et d'éviter un stade de la simulation où il n'y a plus que des organismes sans aucune chance de se reproduire.

Le principal challenge pour l'implémentation a été de ne pas perdre le type d'un lattice lorsqu'il est en mouvement, que ce soit dans lattice individuel ou pour les lattices individuels. Nous avons au final simplement augmenté la capacité de stockage d'information dans un œuf. Un oeuf d'enfant se présente donc tel que:

$valeur = 1000 + 100 * direction + 10 * type + n$, avec n le nombre de pas que doit encore parcourir l'enfant avant de se poser.

Les simulations fonctionnent, mais nous n'avons pas encore repéré de comportement qui sort de ce que nous avons anticipé. Cela s'explique notamment par le fait que les chances de reproduction ont drastiquement chuté, entre la contrainte du type et la durée de vie d'un organisme. Un exemple de simulation est trouvable dans la vidéo "bio_lgca_reproducing_pairs_typed.mkv"

Conclusions

Nous avons réussi à utiliser BIO-LGCA pour reproduire de manière simplifiée des comportements existants dans la vie. Il est sûrement possible de continuer à utiliser BIO-LGCA et de l'optimiser pour avoir des modèles à la fois plus simples et plus réalistes. Par exemple, nous avons réfléchi à une manière d'implémenter des chaînes d'ARN plus longues que 2 lattices, et le problème majeur auquel il faudrait faire face est la collision entre une longue chaîne et d'autres éléments de la simulation.

De plus, nous nous sommes concentrés sur un aspect de dynamique de particules, mais comme dit en introduction, il est tout à fait possible de revenir à un automate cellulaire plus classique et de le simuler à l'aide de BIO-LGCA. Nous en avons fait la démonstration en implémentant le Jeu de la Vie avec BIO-LGCA, trouvable dans la vidéo "bio_lgca_gol.mkv".

Je suis conforté dans mon idée que BIO-LGCA est un très bon automate cellulaire dans le sens où il est possible de faire beaucoup de choses avec et de simuler beaucoup d'environnements avec des règles très différentes. Le modèle permet même d'à la fois simuler des simulations de particules et des automates cellulaires classiques. De plus, l'exportation de ces règles est très simple puisque la fonction d'interaction est une fonction qui va d'un vecteur vers un autre vecteur. Si elle est déterministe, elle peut être encodée dans un simple dictionnaire.

Nous nous sommes concentrés sur de la simulation de comportements biologiques à travers une sorte de simulation de particules, grâce aux déplacements de lattices. D'autres pistes de recherches seraient par exemple d'étudier comment obtenir de l'émergence avec un petit nombre de types de particules et des règles simples qui régissent leurs interactions, ou implémenter des oeufs qui grandissent en un pattern défini dans l'ADN de l'œuf.

Annexe

Différences entre le BIO-LGCA original et notre version

- Vitesses possibles: Le papier original précise que les communication channels peuvent également encoder différentes vitesses, permettant à une particule de “sauter” un ou plusieurs lattices voisins. Dans notre cas, nous avons omis cette caractéristique en ne permettant qu’une seule vitesse possible, qui permet d’aller dans un lattice voisin.
- Voisinage: Dans le papier original, la fonction d’interaction peut prendre plusieurs lattices en plus de celui sur lequel elle agit, considérés comme le voisinage. Nous n’avons pas permis cela dans nos recherches car nous considérons plutôt un point de vue cellulaire de la simulation et il semblait moins réaliste qu’une cellule agisse différemment en fonction des molécules présentes dans des cellules voisines.
- Stochasticté: Les fonctions d’interaction décrites par Deutsch sont des processus stochastiques. Dans notre cas, toutes les probabilités étaient définies à 0 ou à 1, décrivant des processus déterministes.

Références

[1] <https://particle-life.com/>, Tom Mohr. Exemple de dynamique moléculaire pour étudier la vie biologique.

[2] <https://www.youtube.com/watch?v=p4YirERTVF0>, Tom Mohr. Vidéo de présentation de Particle-Life.

[3] *Deutsch A, Nava-Sedeño JM, Syga S, Hatzikirou H. BIO-LGCA: A cellular automaton modelling class for analysing collective cell migration. PLoS Comput Biol. 2021 Jun 15;17(6):e1009066. doi: 10.1371/journal.pcbi.1009066. PMID: 34129639; PMCID: PMC8232544.*
Papier original décrivant le BIO-LGCA

[4] *Francisco Merino-Casallo, Maria Jose Gomez-Benito, Silvia Hervas-Raluy & Jose Manuel Garcia-Aznar (2022) Unravelling cell migration: defining movement from the cell surface, Cell Adhesion & Migration, 16:1, 25-64, DOI: 10.1080/19336918.2022.2055520.*
Papier mentionnant le BIO-LGCA en tant qu’exemple pour la modélisation des migrations de cellules.