



Modélisation cellulaire et simulation physique : contribution à l'analyse de la dynamique de population des insectes ravageurs

Mahamadou Traore

► To cite this version:

Mahamadou Traore. Modélisation cellulaire et simulation physique : contribution à l'analyse de la dynamique de population des insectes ravageurs. Autre [cs.OH]. Université de Bretagne occidentale - Brest, 2018. Français. <NNT : 2018BRES0011>. <tel-01880972>

HAL Id: tel-01880972

<https://tel.archives-ouvertes.fr/tel-01880972>

Submitted on 25 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UBO

université de bretagne
occidentale

UNIVERSITE
BRETAGNE
LOIRE



THÈSE / UNIVERSITÉ DE BRETAGNE OCCIDENTALE

sous le sceau de l'Université Bretagne Loire

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE ET DE
L'UNIVERSITÉ GASTON BERGER DE SAINT-LOUIS

Mention : Informatique

École Doctorale : MathSTIC, ED 601

présentée par

Mahamadou Traore

Préparée au laboratoire des sciences et techniques de l'information, de la communication et de la connaissance (Lab-STICC UMR CNRS 6285) et au Laboratoire d'analyse numérique et informatique (LANI)

Modélisation cellulaire et simulation physique : contribution à l'analyse de la dynamique de population des insectes ravageurs

Thèse soutenue le 9 mars 2018 devant le jury composé de :

Monsieur Christophe Cambier

HDR, Université Pierre et Marie Curie, IRD / *Rapporteur*

Monsieur Jean Pierre Muller

HDR, CIRAD / *Rapporteur*

Monsieur Vincent Rodin

Professeur, Université de Bretagne Occidentale / *Examineur*

Madame Alexandra Fronville

Maitre de conférences, Université de Bretagne Occidentale / *Examineur*

Monsieur Moussa Lo

Professeur, Université Gaston Berger / *Examineur*

Monsieur Bernard Pottier

Professeur, Université de Bretagne Occidentale / *Co-directeur de thèse*

Monsieur Ousmane Thiare

Professeur, Université Gaston Berger / *Co-directeur de thèse*

THÈSE / UNIVERSITÉ DE BRETAGNE OCCIDENTALE

sous le sceau de l'Université Bretagne Loire

pour obtenir le titre de

**DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE ET DE
L'UNIVERSITÉ GASTON BERGER DE SAINT-LOUIS**

Mention : Informatique

École Doctorale : MathSTIC, ED N° 601

présentée par

Mahamadou Traore

Préparée au laboratoire des sciences et techniques de l'information, de la communication et de la connaissance (Lab-STICC UMR CNRS 6285) et au Laboratoire d'analyse numérique et informatique (LANI)

Thèse soutenue le 9 Mars 2018 devant le jury composé de :

Monsieur Christophe Cambier

HDR, Université Pierre et Marie Curie, IRD / *Rapporteur*

Monsieur Jean Pierre Muller

HDR, CIRAD / *Rapporteur*

Monsieur Vincent Rodin

Professeur, Université de Bretagne Occidentale / *Examineur*

Madame Alexandra Fronville

Maitre de conférences, Université de Bretagne Occidentale / *Examineur*

Monsieur Moussa Lo

Professeur, Université Gaston Berger / *Examineur*

Monsieur Bernard POTTIER

Professeur, Université de Bretagne Occidentale / *Directeur de thèse*

Monsieur Ousmane THIARE

Professeur, Université Gaston Berger / *Codirecteur de thèse*

**Modélisation cellulaire et
simulation physique :
contribution à l'analyse de la
dynamique de population des
insectes ravageurs**

Résumé

Les insectes ravageurs constituent depuis longtemps une menace sérieuse pour l'agriculture dans plusieurs régions du monde. En particulier, les criquets pèlerins (*Schistocerca Gregaria*) sont les plus redoutés à cause des dégâts colossaux qu'ils peuvent infliger à une large variété de cultures. Ces derniers, dans leur milieu de reproduction et un ensemble de paramètres climatiques, peuvent être assimilés à un système biophysique. Un système biophysique associe plusieurs composants caractérisés par leur complexité propre et celle de leurs interactions. Par exemple la pluie apporte de l'humidité et le développement du couvert végétal qui sont critiques dans la dynamique de population du criquet pèlerin. Ces systèmes biophysiques peuvent être contrôlés par divers moyens, dont un réseau de surveillance couplant des capteurs sans fil. L'intégration des systèmes biophysiques et des réseaux d'observation peut être un des faits majeurs de cette décennie. Elle permet un échange entre les processus cycliques physiques et les systèmes d'information. Ces derniers permettent de collecter et de traiter les données de terrain et peuvent aussi agir sur le système physique par le biais de boucles rétro-actives. Cette intégration a été possible grâce aux avancées technologiques notées dans les domaines de la micro électronique et de la transmission sans fil. On la trouve dans des domaines émergents tels que l'agriculture fine.

Cette thèse est consacrée à la modélisation cellulaire et à la simulation de systèmes physiques. Pour cela, un échantillonnage spatial et temporel d'une zone de reproduction des criquets pèlerins a été réalisé à l'aide d'outils spécifiques avec différentes résolutions, dans le but de générer les systèmes cellulaires. Ces systèmes cellulaires contiennent des informations telles que le vent, la température, l'humidité relative et la pluviométrie. L'échantillonnage spatial permet une représentation de migration et l'échantillonnage temporel permet de suivre l'évolution locale des individus dans une cellule.

Les systèmes cellulaires sont transcrits en réseaux de processus communicant et évoluent de manière synchrone afin de reproduire et de simuler le phénomène de migration intercellulaire et le cycle de vie du criquet pèlerin dans une cellule.

La construction de ces systèmes a été explorée en profondeur, en faisant varier leur contexte géographique, les critères de classification, la dimension des cellules, la connectivité entre les processus, les comportements collectifs. L'exploration a porté sur la génération de codes parallèles et les performances à l'exécution, essentiellement pour le cas des processus légers.

Ce travail a abouti à la production de deux simulateurs paramétriques, le premier est destiné à la dynamique de population des criquets pèlerins en vue d'évaluer l'état de la situation acridienne par un couplage entre les réseaux de surveillance synchrones et les systèmes biophysiques. Le second permet de planifier le déploiement de réseaux de capteurs sans fil dans une zone afin de déterminer l'emplacement des capteurs.

Remerciements

Au terme de cette rédaction, je suis conscient du fait que la recherche est loin d'être un travail solitaire. En effet, je n'aurai jamais pu réaliser ces travaux de recherche doctorale sans le soutien d'un grand nombre de personnes dont la générosité, l'engagement et l'intérêt manifestés à mon égard et à ce sujet de thèse, m'ont permis d'en arriver là.

Je tiens tout d'abord à remercier et exprimer ma sincère reconnaissance aux deux co-directeurs de thèse, Monsieur Bernard Pottier, *Professeur à l'Université de Bretagne Occidentale et responsable du groupe WSN au LabSTICC* et Monsieur Ousmane Thiaré, *Professeur à l'Université Gaston Berger de Saint-Louis*. Monsieur Bernard Pottier m'a accueilli à bras ouvert au LabSTICC et à accepter de co-encadrer cette thèse, Ses conseils tant sur le plan technique que pédagogique m'ont été utiles. J'aimerais aussi lui dire à quel point j'ai apprécié sa grande disponibilité, ses qualités humaines d'écoute, de patience et de compréhension tout au long de ce travail de recherche.

Je remercie chaleureusement et exprime ma gratitude à Monsieur Laurent Esclade de l'Université de Bretagne Occidentale pour ses contributions dans le domaine entomologique des insectes.

Mes très sincères remerciements vont également à l'endroit des membres du jury :

- Monsieur Alain Hillion *Professeur à Télécom Bretagne* pour avoir bien voulu accepté de présider ce jury,
- Monsieur Christophe Cambier, *Maître de Conférence HDR à l'Université Pierre et Marie Curie et membre de l'Institut de Recherche pour le Développement (IRD)*, rapporteur,
- Monsieur Jean Pierre Muller, *Maître de Conférence HDR au Centre de coopération Internationale en Recherche Agronomique pour le développement (CIRAD)*, rapporteur,
- Monsieur Vincent Rodin *Professeur à l'Université de Bretagne Occidentale*, examinateur,
- Madame Alexandra Fronville *Maître de conférence à l'Université de Bretagne Occidentale*, examinateur,
- Monsieur Moussa Lo, *Professeur à l'Université Gaston Berger de Saint-Louis* et coordonnateur de l'Université Virtuelle du Sénégal (UVS), examinateur.

Aussi, je tiens à exprimer toute ma sympathie à tous ceux et/ou toutes celles qui ont contribué de près ou de loin au bon déroulement de cette thèse. Je pense particulièrement à Dr Pierre-Yves Lucas, à Dr Eloi Badian KEITA, à ma nièce chérie mademoiselle Valerie Gleize, étudiante en licence 3 d'économie à l'UBO, qui m'ont soutenu, encouragé et partagé avec moi des moments inoubliables durant mes séjours à Brest. Ces remerciements seraient incomplets si j'oubliais Bao LAM Hoai, Tuyen Phong TRUONG, Raonirivo N Rakotoarijaona, tous doctorant au LabSTICC. Mes remerciements vont également à l'endroit de tous les enseignants-chercheurs de la section informatique de l'UFR SAT de l'Université Gaston Berger de Saint-Louis pour leur soutien. C'est une thèse de cotutelle entre l'Université Gaston Berger de Saint-Louis du Sénégal et l'Université de Bretagne Occidentale à Brest en France.

Elle a été financée par le programme d'appui et de promotion du personnel enseignant et de recherche (PA2PER) de l'université Gaston Berger de Saint-Louis, du fond d'impulsion pour la recherche scientifique et technique (FIRST2012) et du centre d'excellence africain mathématiques, informatique, technologies de l'information et de la communication (CEA-MITIC). Je remercie et exprime toute ma gratitude aux responsables de ces structures. Je remercie aussi les autorités du centre National de Lutte anti-acridienne (CNLA) à Nouakchott, en Mauritanie, en particulier Monsieur Mohamed Abdallahiould Baba, le directeur et Ahmed Salem Ould Benahi, responsable du service d'information acridienne, ainsi que tout le personnel du centre, pour l'accueil, la sympathie et la disponibilité dont ils ont fait montre lors de mon séjour. Enfin mes remerciements à l'endroit de l'équipe de prospection à Targue situé à un peu plus de 300 Km au Nord-Est de Nouakchott, je pense particulièrement à Bady Ould Oubeid, le responsable de l'équipe, à Moustaph, le chauffeur du désert, à Cha, le cuisinier de l'équipe et Sidy, un bénévole pour leur assistance.

Dédicaces

À la mémoire de Téné TRAORE et à celle de Fatima DOUMBIA, respectivement, ma petite soeur et ma cousine arrachées très tôt à notre affection. Que Dieu les accueille au paradis.

Je dédie cette thèse :

A mes enfants : Abou Bakr Sedikh, Mariama et Abdoul Karim TRAORE
à qui j'adresse ici un message de foi, d'amour, de volonté et de persévérance dans le travail.

A mon épouse : Khadidiatou Camara,
qui a fait preuve d'un soutien indéfectible durant toute cette thèse, en s'occupant, seule, de nos trois jeunes enfants lors de mes séjours à Brest.

A mes parents : Bréhima TRAORE et Khadidiatou DOUMBIA,
pour leurs soutiens, prières et assistance morale.

A mes tantes : Awa Keita, Aissatou Camara et Siramory Camara,
pour leurs soutiens et prières.

A mes frères et soeurs.

Table des matières

Introduction générale	1
1 Le criquet pèlerin	7
1.1 Introduction	7
1.2 Les préférences bio-écologiques du criquet pèlerin	8
1.2.1 Les zones de reproductions du criquet pèlerin	10
1.2.2 Les aires grégariques	10
1.2.3 Le cas de la Mauritanie	11
1.3 Le cycle biologique du criquet pèlerin	12
1.3.1 La fréquence des pontes	12
1.3.2 Le développement des œufs	14
1.3.3 Le développement des larves	15
1.3.4 Le développement des ailés	15
1.4 La migration des criquets pèlerins	16
1.5 Les données acridiennes	16
1.6 Conclusion	17
2 Éléments techniques pour la modélisation de la dynamique des criquets pèlerins	19
2.1 Introduction	19
2.2 La dynamique de population	21
2.2.1 Etat de l'art	21
2.2.2 Le cas des criquets pèlerins	25
2.3 Les automates cellulaires	27
2.3.1 Un bref historique	27
2.3.2 Le modèle du jeu de la vie	27
2.3.3 Les dimensions d'un AC	29
2.3.4 Les domaines d'applications des AC	30
2.3.5 Autres exemples d'AC	30
2.4 Modélisation cellulaire du système physique	30
2.4.1 L'échantillonnage spatial	31
2.4.2 L'échantillonnage temporel	32
2.4.3 Les règles de transition	33
2.5 PickCell, outil de génération de systèmes cellulaires	33

2.5.1	Génération d'un maillage de système cellulaire	35
2.5.2	Algorithme de classification cellulaire par couleurs	35
2.6	Conclusion	37
3	Accès aux sources de données géolocalisées	39
3.1	Introduction	39
3.2	Les promoteurs de bases de données	40
3.2.1	Les initiatives internationales	40
3.2.2	Les initiatives internationales privées	42
3.2.3	EIS-AFRICA, une initiative continentale	44
3.2.4	Les initiatives nationales publiques	44
3.2.5	Les initiatives nationales privées	45
3.3	La correspondance objet-relationnel	46
3.3.1	La bibliothèque de correspondance : "Glorp"	47
3.3.2	Notions de transactions objet et d'unités de travail	48
3.3.3	Connexion à la base de données	49
3.3.4	Modélisation d'une base de données avec "Glorp"	51
3.4	Intégration PostgreSQL et application Visualworks	53
3.4.1	Le support de bases de données : "EXDI"	54
3.4.2	Chargement du support de bases de données	54
3.4.3	Connexion à la base de données	55
3.5	Conclusion	55
4	Programmation d'automates cellulaires représentant la dynamique du cri-	
	quet pèlerin	57
4.1	Introduction	57
4.2	Représentation cellulaire des populations de criquets	58
4.3	Génération du système cellulaire	59
4.4	Les modèles cellulaires d'évolution locale et de migration	60
4.4.1	Les variables d'état et la représentation physique paramétrique	61
4.4.2	La fonction de transition et la quantification des populations	62
4.4.3	Le modèle d'évolution locale	63
4.4.4	Le modèle de migration intercellulaire	65
4.5	Les résultats de simulation	67
4.5.1	Illustration de la migration inter cellulaire	67
4.5.2	Illustration du développement dans une cellule	68
4.5.3	Illustration de la propagation spatiale	68
4.6	Conclusion	70
5	Simulations parallèle et distribuée	71
5.1	Introduction	71
5.2	La simulation distribuée	71
5.2.1	Le couplage par HLA et la puissance par CUDA	72
5.2.2	Fonctionnement du HLA	73

5.2.3	La gestion des horloges avec HLA	75
5.2.4	La communication entre simulateurs	76
5.2.5	La simulation physique distribuée : cas des criquets	77
5.3	Conclusion	81
6	Calcul de couverture radio et déploiement de capteurs	83
6.1	Introduction	83
6.2	La méthodologie de synthèse de systèmes cellulaires	85
6.3	Acquisition de données d'élévation	86
6.3.1	Les modèles numériques d'élévation, de terrain et de surface	86
6.3.2	Exemple d'utilisation d'un MNE	87
6.4	Couverture et algorithmes distribués	88
6.4.1	Algorithme du maximum distribué	88
6.4.2	Algorithme de détermination du leader d'un réseau	89
6.4.3	Calcul du diamètre d'un réseau	90
6.4.4	Algorithme du Bread First Search (BFS)	93
6.4.5	Algorithme de calcul de pente	94
6.5	Etude de cas : calcul de couverture radio à l'UGB	95
6.5.1	Analyse des résultats	97
6.6	Conclusion	97
7	Conclusions et perspectives	101
A	Annexe A	103
A.1	Extrait de fichier Occam du réseau cellulaire	103
B	Annexe B	107
B.1	Extrait du fichier Occam des données cellulaires	107
C	Annexe C	109
C.1	Algorithme de regroupement de cellules en classes	109
C.2	Algorithme de regroupement de cellules par affinité	110
D	Annexe D	111
D.1	Exemple de fichiers GRIB	111
E	Annexe E	113
E.1	Taxonomie des architectures	113
E.2	Les différentes architectures des calculateurs	114
E.2.1	Les architectures parallèles	114
E.2.2	Les modèles architecturaux classiques	116
	Acronymes et définitions	120

Table des figures

1.1	Foyers grégaires du Criquet pèlerin entre 1926 et 1976 (Source : WALOFF, 1976 [233])	10
1.2	Les zones de reproduction en Mauritanie (Source : Duranton & Lecoq, Modifié).	11
1.3	Le cycle biologique du criquet pèlerin (source : Duranton & Lecoq) . .	13
1.4	Déplacement des criquets pèlerins avec le vent, (Source : Duranton & Lecoq)	16
2.1	La dynamique de population du criquet pèlerin (Source : J. Roffey et Al, Modifié [203]).	25
2.2	AC, un exemple d'état initial.	28
2.3	Le voisinage d'une cellule cible.	28
2.4	Nombre de voisins des cellules.	28
2.5	Etat suivant.	29
2.6	Le voisinage de Von Neumann 1 présente quatre voisins correspondant aux quatre directions cardinales alors que celui de Moore 1 présente tous les voisins autour d'une cellule à la distance 1.	30
2.7	Découpage spatial du domaine d'étude sur une superficie de 100 ha . .	31
2.8	Ces figures représentent la migration des individus dans les cellules . .	34
2.9	Le canevas de génération de systèmes cellulaires	35
2.10	Un exemple de distribution d'image en sept classes	36
2.11	Diviseur 2, 8 pour un espace de valeurs de $256 \times 256 \times 256$	36
3.1	Les moyennes mensuelles de température et de pluviométrie, en Mauritanie durant la période de 1990 à 2012 [3]	41
3.2	Couplage de modèles de climat et de migration	42
3.3	Interactions objet-relationnel	46
3.4	Architecture de "Glorp" et interactions avec une base de données	49
3.5	Chargement de la parcelle postgresSQL3EXDI	55

4.1	Illustrations de l'évolution locale dans une cellule et de migration entre cellule. La voisine de l'Ouest envoie une population entrant à la cellule C. Après les différentes transitions entre les micro-états, la voisine Sud-Est reçoit une population sortant. Cette figure résume nos modèles et montre comment les interactions locales entre les micro-états et entre les cellules se passent. Toutes ces interactions se font de manière concurrente : le cycle de vie et ses habitants est un pipeline.	58
4.2	Topologie utilisée dans les modèles cellulaires. Chaque cellule contient un micro-système similaire à celui de la figure 4.1. Les communications figurent les dépendances physiques, ici, les migrations locales.	59
4.3	Edition de coordonnées géographiques d'un système cellulaire	60
4.4	Données d'une cellule. À gauche, on trouve la spécification Occam des structures, et à droite les valeurs associées produites par le générateur de code de PickCell. On reconnaît les pixels extraits de l'image cartographique initiale.	62
4.5	Trois principales étapes peuvent être distinguées dans le cycle de vie du criquet pèlerin : œuf, larve et adulte. La phase larvaire est composée de cinq étapes (L1 à L5). À la dernière étape larvaire, la mue imaginale se produit produisant des ailés immatures qui deviennent matures au bout de quelques semaines et dont les femelles peuvent pondre si l'humidité est suffisante dans une cellule.	63
4.6	Trace de simulation : evolution locale des individus	64
4.7	Trace de simulation : evolution de la densité des individus sur chaque état et micro-état.	66
4.8	Transfert de migrants dans une cellule	68
4.9	Evolution des individus dans une cellule	69
4.10	Propagation spatiale au tour synchrone 30 autour de la cellule P_{x2y6R}	69
4.11	Propagation spatiale au tour synchrone 60 autour de la cellule P_{x2y6R}	70
5.1	Architecture HLA : Une fédération de trois simulateurs reliés à un bus "Run Time Infrastructure" (RTI) (Source [229]).	73
5.2	Interactions entre un simulateur et une fédération (Source [229]).	74
5.3	Progression temporelle dans le simulateur (Source [229]).	75
5.4	Communication entre simulateurs et le bus RTI	76
5.5	Structure de la fédération de simulateurs	78
5.6	Simulation de migration cellulaire avec un voisinage de Von Neumann. La couleur rouge représente la propagation des criquets	79
5.7	Etat des cellules après vingt tours synchrones	80
6.1	Une vue de l'UGB produite avec QuickMap. La tuile OpenStreetMap nous permet d'obtenir cette représentation. D'autres modèles de tuile existent.	84
6.2	Segmentation spatiale. Nous utilisons une grille de dimension 50@50 qui permet de segmenter la vue graphique en 50×50 pixels.	85

6.3	Sur la figure, apparaissent les valeurs d'élévations par rapport au niveau de la mer, résolues par GPS Visualizer. Cette résolution se fait sur la base des positions géographiques des points.	87
6.4	Une vue graphique de la zone d'étude	96
6.5	Carte de visibilité du réseau	96
6.6	Profil d'élévation avec Von Neumann 1 comme topologie de voisinage .	97
6.7	La carte de couverture du réseau avec un voisinage de Moore1	98
6.8	Profil de connectivité avec le voisinage de Von Neumann 2	98
6.9	Profil de connectivité avec le voisinage de Moore2	99
E.1	Schéma fonctionnel simplifié d'un ordinateur	114
E.2	Traitement séquentiel et parallèle	116
E.3	Architecture à mémoire partagée	117
E.4	Architecture à mémoire distribuée	118

Liste des tableaux

1.1	Densités approximatives relatives aux catégories L_i de transformation phasaire chez le criquet pèlerin (Source : [178, 204]). Voir aussi le cycle présenté figure 1.3	9
1.2	Nombre moyen de pontes dans des conditions favorables, d'après [183]	14
1.3	Nombre moyen de pontes dans des conditions défavorables, d'après [183]	14
2.1	Les échantillons temporels	32
3.1	Les méthodes de la classe "DescriptorSystem"	54
4.1	Les éléments de segmentation spatiale de la vue graphique. Les paramètres MinFanOut et MaxFanOut représentent respectivement le nombre minimum de liens sortant d'une cellule et le nombre maximum de liens sortant d'une cellule. MaxFanOut affichera toujours la valeur 8 à cause de la topologie de voisinage utilisée, qui est Moore de rayon 1.	61
4.2	Les paramètres de simulation	67
5.1	Politique temporelle dans notre cas d'étude	76
5.2	Les objets et attributs publiés par les simulateurs fédérés	77

Introduction

Les insectes ravageurs sont considérés depuis longtemps comme une sérieuse menace pour les productions agricoles dans plusieurs régions du monde, en particulier au Sahel, où l'autosuffisance alimentaire et la sécurisation des cultures ne sont toujours pas maîtrisées.

Aucune culture n'est épargnée pendant les périodes d'invasions et les pertes sont évaluées par dizaine de milliers de tonnes par jour. Les dégâts sur l'environnement liés à l'utilisation d'insecticide et de pesticide sont aussi importants.

Des stratégies de lutte efficaces et concertées contre ces insectes permettent cependant de limiter les dégâts. Elles nécessitent des moyens matériels et humains très importants.

La lutte anti-acridienne reste aujourd'hui une priorité *internationale*, parce que les pays sahéliens pris individuellement ne peuvent pas à eux seuls faire face à ce fléau. Cette lutte est donc l'objet d'attentions spécifiques d'organismes internationaux tels que la *Food and Agriculture Organization* (FAO). Plusieurs centres de recherche agissent également dans la surveillance et le développement rationnel des productions agricoles. On peut retenir des opérateurs de recherche généralistes :

- le *Centre de coopération Internationale en Recherche Agronomique pour le Développement* (CIRAD) ;
- l'*Institut pour la Recherche et le Développement* (IRD).

des acteurs spécialisés et des commissions régionales :

- le *Comité permanent Inter-Etats de Lutte contre la Sécheresse dans le Sahel* (CILSS) ;
- la *Commission de Lutte contre le Criquet Pèlerin dans la Région Occidentale* (CLCPRO) ;
- la *Commission de lutte contre le criquet pèlerin dans la Région Centrale* (CRC) ;
- la *Commission de lutte contre le criquet pèlerin en Asie du Sud-Ouest* (SWAC) ;
- l'*organisation de lutte contre le criquet pèlerin en Afrique de l'Est* (DLCO-EA) ;
- et l'*organisation commune de lutte anti-Acridienne et de lutte anti-Aviaire pour les pays d'Afrique de l'Ouest au sud du Sahara* (OCLALAV).

Cette dernière s'est réorganisée pour couvrir la prise en charge des opérations de lutte anti-acridienne dans chacun des états de l'Afrique de l'Ouest, sur leurs territoires respectifs. Cependant, elle conserve son rôle de coordination comme celui des commissions régionales de la FAO. La FAO coordonne les activités de lutte à l'échelle internationale et apporte de l'aide pour les opérations de lutte d'urgence. Elle collecte des informations sur la situation acridienne et diffuse des bulletins de prévisions et d'avertissement.

La lutte anti-acridienne repose sur quatre ressources qui sont :

- la prospection terrestre ;
- les conditions météorologiques fournies par les stations météo ;
- les données de végétation fournies par des observations satellites ;
- et aussi des informations fournies par la population locale.

La prospection terrestre consiste en un déploiement saisonnier de spécialistes de terrain qui évaluent la situation acridienne et les conditions écologiques d'une zone suivant des programmes préétablis. Ces agents utilisent des appareils de saisie, de retransmission de données par satellite, de moyens de communication radio et de navigation par GPS. Ils suivent des itinéraires de prospection étudiés au préalable en fonction de la répartition pluviométrique fournie par les stations météo, et de la couverture végétale produite par des satellites. Les moyens techniques déployés sont importants, mais restent *essentiellement manuels*. Un objectif est, par exemple, le recensement visuel des individus ravageurs effectué par des prospecteurs sur le terrain[34]. Les données collectées et stockées localement, sont archivées par le service acridien de la FAO pour analyse et planification des futures interventions.

Le risque posé par ces insectes peut se résumer à une causalité sur des échelles de temps courtes, où les précipitations amènent à la fois le développement des cultures et de leurs ravageurs. Il est reconnu que la présence des criquets pèlerins dans une zone est liée à une disponibilité en nourriture qui s'installe dans cette zone, après des pluies. En effet la pluie conditionne directement ou indirectement, toutes les étapes de la reproduction, de la ponte à la maturation, en passant par les périodes de développements embryonnaire et larvaire, des milieux favorables. Les milieux les plus favorables sont ceux qui reçoivent une quantité de pluie annuelle comprise entre 25 et 50 millimètres [144].

Les criquets pèlerins sont caractérisés par leur grande capacité à migrer. Un exemple en est la fin du printemps, où ils quittent leur milieu de reproduction printanière, au nord-ouest de l'Afrique, pour rallier les zones de reproductions estivales situées dans plusieurs pays du Sahel. Ces migrations se font sur de très grande distance, les insectes parcourant des milliers de kilomètres en quelques jours [58]. La direction de migration dépend fortement de la direction et de la vitesse du vent, de la température, de l'humidité relative et de la pluviométrie. Les criquets pèlerins volent en suivant la direction des vents principaux, les femelles recherchant de l'humidité pour la ponte des oeufs et de la végétation pour fournir abris et nourriture aux larves.

La maîtrise de la dynamique spatio-temporelle des populations de criquets pèlerins et de leur cycle biologique dans un espace géolocalisé permet de comprendre et prévenir les invasions en intégrant le niveau de développement des individus.

De même l'accès aux bases de données est critique pour les applications de lutte anti-acridienne. Ces applications sont fondées sur des modèles qui interrogent les bases de données, puis peuvent conditionner les cellules d'un découpage spatial. Elles peuvent aussi intégrer des données géographiques, climatologiques et météorologiques, ainsi que les historiques d'actions anti-acridiennes.

Contributions Le travail présenté ici a principalement consisté à modéliser et simuler la dynamique de population des insectes ravageurs et leur cycle de vie. Cette démarche va faciliter des prédictions sur la densité des populations et leur état de développement dans le temps et dans l'espace. Il s'agit d'un problème dont la complexité est attestée par les détails du cycle de vie de ces insectes, et leur capacité à se déplacer en masse sur des grandes distances.

Il a aussi permis de simuler la configuration topologique d'un réseau de capteurs dans une zone donnée. L'objectif recherché ici est de mettre en place un système de communication permettant d'agir sur les capteurs déployés, en envoyant des commandes ou des informations de reconfiguration.

La direction choisie est celle où l'espace et le temps sont subdivisés en grille cellulaire évoluant par étape et en synchronisme, permettant la simulation physique des développements et des mouvements. Cette grille peut encapsuler les résultats d'une observation réelle intégrant la géographie, les constats de terrain, les précipitations, la végétation, etc. . .

La modélisation est une démarche pratiquée dans beaucoup de disciplines scientifiques pour étudier notamment l'évolution des phénomènes physiques. Le calcul des évolutions prédites est le rôle de la simulation.

Les *systèmes cellulaires* sont une représentation segmentée des criquets pèlerins dans leur milieu de reproduction en interaction avec des paramètres climatiques. Un jeu d'outils développé au laboratoire d'informatique de l'université de Brest, a permis de réaliser des représentations cellulaires à différentes échelles, puis de représenter la dynamique des populations de criquets pèlerins en utilisant le paradigme de calcul des automates cellulaires. Ceci a abouti à la production d'un simulateur paramétrique de dynamique de population basé sur le contexte géographique et le cycle de vie.

La représentation informatique associant l'espace, le temps et l'évolution physique des criquets pèlerins dans leur milieu peut être rapprochée à la problématique "cyber-physique [145, 149, 147]". L'idée finalisée est d'équiper les cellules communicantes du monde réel de capteurs programmés mis en réseau, et connectés à un système d'information capable de prises de décision, voire de commander des actions.

Les cellules disposent de canaux de communication abstraits, entrants et sortants, qui convoient les dépendances physiques tels que les passages de flux migratoires.

Le plan de travail a consisté en état de l'art sur l'insecte objet de l'étude, les dispositifs de luttés, les contextes proches de la modélisation et la simulation des populations. Les outils de génération de systèmes cellulaires et la production de simulateurs ont été développés à partir de 2014, en liaison avec ce contexte et leur disponibilité a permis d'envisager un flot méthodologique cellulaire qui a procédé de la manière suivante :

1. représentation des états du cycle de vie du criquet ;
2. représentation des états sur la base de cellules géo-localisées représentant les évolutions de groupes d'individus en disjoignant les différentes étapes par comptage ;
3. intégration des migrations de criquets entrant et sortant des cellules ;

4. production de systèmes parallèles dont les processus sont des cellules échangeant des migrants ;
5. intégration de paramètres climatologiques, en particulier le vent ;
6. maquette de combinaisons de simulation conformes au standard IEEE *High Level Architecture*, à même d'associer plusieurs simulations de natures différentes, ou d'affichage ;
7. expérimentations.

En ce qui concerne la réalisation, le principal élément logiciel est une procédure exécutée par chaque cellule (point 2). Il y a ensuite un parallélisme massif permettant de traiter plusieurs milliers de cellules en même temps. Les échanges entre cellules (migrations) impliquent des synchronisations qui vont suivre les évolutions locales, et le calcul des volumes de migrations,

Des règles de transitions basées sur l'observation des paramètres météo et la densité de végétation conditionnent l'évolution des états des cellules. L'originalité de notre approche réside dans le fait que les données environnementales externes que sont le vent, la température, la pluviométrie sont intégrées dans les cellules grâce à leur géo-localisation.

Cette construction informatique va se heurter à au moins deux difficultés majeures : les facteurs d'échelle qui peuvent être très grands dans le cas d'insectes migrant sur des centaines de kilomètres, la connaissance de ces insectes et de leurs cibles agricoles qui exigent de suivre les travaux de terrain menés dans ces domaines. Les connaissances obtenues sur les insectes et les stratégies de lutte organisées contre eux proviennent de documents techniques disponibles sur Internet, des consultations de chercheurs entomologiques (dont Mr Michel Lecoq, chercheur au CIRAD de Montpellier). Une visite au CNLA¹ à Nouakchott en Mauritanie m'a permis de participer à des prospections sur le terrain et d'observer les moyens de luttés actuels.

La suite de ce manuscrit sera structurée comme suit : le chapitre 1, sera consacré aux criquets pèlerins, à leurs préférences biologique et écologique, aux phénomènes de migration saisonnière qui les caractérise et aux données d'entrée des modèles développés dans ce travail. Le chapitre 2 présentera un état de l'art sur la dynamique des population, les automates cellulaires (AC), la modélisation cellulaire avec AC, PickCell, un outil de génération de systèmes cellulaires. Le chapitre 3 portera sur comment accéder à des bases de données partagées par des promoteurs et comment déployer des bases de données locales qui serviront à alimenter nos modèles. Le chapitre 4 sera consacré à la représentation cellulaire de population de criquets, à la génération de systèmes cellulaires et à la présentation des deux modèles de migration et d'évolution locale. Le chapitre 5 portera sur les principe et méthode de simulations sur architecture distribuée/parallèle. Les concepts de simulation parallèle et distribuée avec un couplage de CUDA et du standard HLA seront présentés, avec une étude de cas sur les criquets dans leur milieu soumis aux paramètres climatiques et quadrillé par des capteurs. Le chapitre 6 portera sur les méthodes de calcul de couverture radio permettant de planifier un déploiement de

1. Centre National de Lutte anti-acridienne

réseaux de capteurs sans fil. Ceci permettra de collecter des informations telles que le sens et la direction de déplacement, la densité d'un nuage de criquets dans un espace déterminé. Et le chapitre 7 conclura ce manuscrit et en posera des perspectives.

1

Le criquet pèlerin

1.1 Introduction

«Le criquet pèlerin (*Schistocerca gregaria* [93])(Forskål, 1775) de la famille des Acrididae et de la sous- famille des Cyrtacanthacridinae, est l'une des espèces acridiennes les plus redoutables en raison de :

- sa grande mobilité (les essaims peuvent parcourir 1 000 km en quelques jours) ;
- son aire d'invasion étendue couvrant une aire de plus de 29 millions de kilomètres carrés, soit plus de 20 % des terres émergées ;
- son grand potentiel reproducteur, il est capable de multiplier très rapidement ses effectifs ;
- sa capacité à consommer chaque jour son propre poids de nourriture fraîche ;
- sa grande polyphagie (en particulier en phase grégaire) le conduisant à s'attaquer à une très large gamme de cultures et à leur causer des dégâts très sévères.

Cet acridien constitue depuis l'antiquité, l'un des principaux fléaux de l'humanité dans l'Ancien Monde ». Dans la Bible, Il est considéré comme la huitième plaie de l'Égypte ancienne, en raison de ses ravages qui s'étendent à la majorité des pays arides et semi-arides, de la côte ouest de l'Afrique à l'Inde.

Les pays du Sahel sont directement menacés par cette espèce qui peut les envahir et compromettre tous les efforts de développement consacrés à l'agriculture. Certains foyers grégarigènes du criquet pèlerin y sont localisés. Les foyers grégarigènes sont des zones propices à la reproduction estivale où prennent naissance les invasions généralisées [77].

Pendant les invasions, des pertes considérables sont infligées aux récoltes, causant des pénalités à la fois à l'économie, à l'environnement et aux populations des pays touchés. La dernière invasion des années 2003 - 2005, en Afrique de l'Ouest a coûté quatre cent

millions de dollars US aux pays affectés et donateurs et des pertes importantes ont été infligées aux productions agricoles destinées à la consommation et à l'exportation [32].

Le développement socio-économique et la sécurité alimentaire des pays sahélo-sahariens dépendent fortement de la disponibilité des ressources naturelles, notamment les terres arables, les ressources hydrauliques, forestières et pastorales. Les zones arides et semi-arides du Sahel à vocation sylvo-pastorale, représentent un ensemble de domaines écologiques et biologiques remarquables qu'il convient de restaurer, conserver et valoriser. Cette valorisation passe par la maîtrise de l'avancée du désert, de la variabilité climatique et des phénomènes d'invasion des insectes ravageurs.

La vulnérabilité face à la désertification, au changement climatique et à la menace acridienne, constitue pour les pays sahélo-sahariens, l'une des plus importantes menaces en ce 21^{ème} siècle. Pour la survie de leur économie et la cohésion sociale, ces pays doivent impérativement mettre en œuvre des réponses efficaces de lutte et de gestion pérenne.

L'une des réponses est le lancement, en 2005 par les états au sud du Sahara de l'Initiative Africaine de la Grande Muraille Verte (IAGMV) qui s'étend de Dakar à Djibouti sur 7600 kilomètres de long et 15 kilomètres de large [10]. Cette initiative, endossée par l'Union Africaine en 2007, se positionne aujourd'hui, comme une solution face aux enjeux environnementaux et climatiques. Le projet de Grande Muraille Verte témoigne de la nécessité de mieux maîtriser la dynamique des populations d'insectes ravageurs, en vue d'une solution pérenne où les limites des aires d'invasion coïncident avec celles de la muraille de protection.

Dans la suite de ce chapitre, nous présenterons en section 1.2 les préférences écologique et biologique et les zones de reproductions du criquet pèlerin, en section 1.3 la description de leur cycle biologique. La section 1.4 portera sur les phénomènes de migrations saisonnières de ces insectes, la section 1.5 sera consacrée aux données d'entrée des modèles et la section 1.6 conclura ce chapitre.

Les détails de cette présentation sont accompagnés de notes *en italiques* précisant quelle représentation sera faite pour la modélisation et la simulation (chapitre 2).

1.2 Les préférences bio-écologiques du criquet pèlerin

Les insectes ravageurs de la zone soudano-sahélienne de l'Afrique de l'Ouest peuvent être classés en deux groupes distincts, les locustes et les sauteriaux. Les locustes sont le criquet nomade, le criquet migrateur africain et le criquet pèlerin. Les sauteriaux sont toutes les autres espèces d'acridiens, ravageurs ou non. Ces deux groupes d'acridiens présentent des comportements différents. Les locustes suivent un phénomène de changement de phase. Ils sont capables, selon la densité de leur population et les conditions climatiques, de se présenter avec deux comportements différents : solitaire et grégaire. Les sauteriaux ne présentent pas ces comportements distincts, néanmoins ils peuvent pulluler et causer des dégâts sur les productions agricoles, mais à un degré inférieur à celui des locustes [170, 102].

Les criquets pèlerins qui colonisent des steppes ou des savanes à faible couvert

ligneux qu'on retrouve principalement au Sahel, recherchent de la chaleur (optimum thermique entre 20°- 25°C), et des milieux moyennement humides (optimum pluviométrique de 50 à 300 mm par an). Dans les régions tempérées, ils peuvent résister aux conditions rigoureuses de l'hiver en développant la *diapause embryonnaire*.

En général, les densités des effectifs solitaires des criquets pèlerins sont faibles et ils vivent de manière isolée dans des endroits désertiques, alors que les individus grégaire vivent en bande pouvant contenir un très grand nombre d'individus qui se comportent comme une masse cohérente [170, 77].

Le passage de la forme solitaire à la forme grégaire n'est pas spontané. Il peut se poursuivre sur au moins quatre générations de criquets et est fortement influencé par une succession de pluies soutenues. Le passage inverse est plus rapide et s'effectue souvent en l'espace d'une ou deux générations [77, 144]. Ce changement de phase se passe dans des zones particulières appelées *zones de rémission*, moins vastes que les *zones d'invasion* (voir Figure 1.4). La rémission est la période entre deux invasions.

Aussi, on reconnaît deux formes intermédiaires dites *transiens* :

- *congregans*, une évolution de la phase solitaire à la phase grégaire ;
- *degregans* dans le cas inverse du passage de la phase grégaire à la phase solitaire.

La densité des individus est le principal facteur déclencheur du changement de phase (table 1.1, [77]). Ce dernier est favorisé par plusieurs facteurs tels que la multiplication des effectifs, et la concentration des individus sur des superficies réduites due au regroupement d'ailés solitaires sous l'effet de systèmes de vents convergents.

Densité	Larves L1 à L3	Larves L4 et L5	Imagos
Forte	10 à 100/m ²	1 à 10 /m ²	1000 à 10 000/ha
Moyenne	2,5 à 10/m ²	0,25 à 1/m ²	250 à 1000/ha
Faible	<2,5/m ²	<0,25/m ²	<250/ha

TABLE 1.1: Densités approximatives relatives aux catégories L_i de transformation phasaire chez le criquet pèlerin (Source : [178, 204]). Voir aussi le cycle présenté figure 1.3

D'autres facteurs peuvent intervenir dans la concentration des populations de criquets, en particulier les phénomènes de restriction de surfaces habitables par dessèchement progressif des biotopes obligent leur rassemblement sur des zones favorables de plus en plus réduites [77].

Les effets de la densité peuvent être immédiats ou différés. La première catégorie se manifeste sans délai chez les individus en présence d'une variation de densité alors la deuxième se manifeste dans la descendance. Les seuils de densité de changement de phase en fonction du stade de développement des individus, sont approximativement de 50 000 par hectare, pour les larves L1, de 5 000 par hectare pour les larves L5 et de 250 à 500 par hectare pour les ailés immatures [144].

Les criquets pèlerins vivent préférentiellement dans des zones ayant reçu des quantités de pluies comprises entre vingt cinq et cinquante millimètres. Ces pluies entraînent

la formation d'un couvert végétal qui constitue une source d'alimentation et de refuge pour les criquets pèlerins. Elles fournissent également de l'humidité favorable au développement des œufs [218].

1.2.1 Les zones de reproductions du criquet pèlerin

Trois périodes de reproduction, d'importances inégales, se succèdent annuellement selon les phases d'existence du criquet pèlerin. Les solitaires effectuent une ou deux reproductions estivales au sud du Sahara durant les pluies de mousson et une reproduction printanière sub-méditerranéenne. Après la mousson, les grégaires effectuent une reproduction estivale (pluies de mousson au Sahel), une reproduction automno-hivernale, occasionnellement dans l'Ouest et plus rarement dans le Sahara central. Il suit une reproduction hiverno-printanière souvent suivie d'une reproduction printano-estivale en zone méditerranéenne [77].

Les reproductions chez les solitaires se font de manière continue, sans interruption et tant que les conditions sont favorables, pendant les mois d'avril-mai à décembre. Ceci permet le développement de deux, voire trois générations estivales suivies d'une génération hiverno-printanière en zone saharo-méditerranéenne [77].

Une des caractéristiques des criquets pèlerins est leur aptitude à grégariser rapidement pour donner des individus voraces. La *grégarisation* est le passage de la phase solitaire à la phase grégaire. Elle est entretenue durant au moins deux générations de criquets pèlerins, si les conditions éco-météorologiques sont maintenues. Elle se déroule dans des zones particulières appelées aires grégariques.

1.2.2 Les aires grégariques

Ce sont des régions qui s'échangent régulièrement des populations de criquets et qui constituent des points de départ d'invasions généralisées.

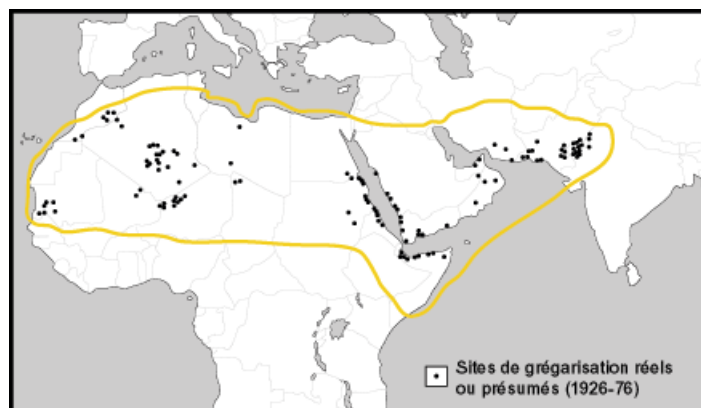


FIGURE 1.1: Foyers grégaires du Criquet pèlerin entre 1926 et 1976 (Source : WALOFF, 1976 [233])

Les aires grégarigènes ont des endroits particuliers appelés des foyers grégaires où se fait effectivement la grégarisation [77]. Les premières étapes de la transformation phasaire conduisant à une invasion généralisée partent de ces zones. En Afrique sud-saharienne, elles sont localisées dans des zones à humidité importante en saison sèche (le delta central du fleuve Niger au Mali, le pourtour du lac Tchad, la région du Nil bleu au Soudan et les oueds du Sahara). À Madagascar, elles sont situées dans les clairières enclavées dans des bush au sud-ouest de l'île.

Plusieurs aires grégarigènes existent dans la zone d'invasion (voir Figure 1.4), les principales se trouvent :

- à la frontière indo-pakistanaise où les régimes de vents favorisent des concentrations de populations importantes ;
- aux bords de la Mer Rouge et du golfe d'Aden où le régime pluvial peut fournir des conditions favorables à la reproduction tout au long de l'année ;
- à la bordure de certains massifs montagneux où les phénomènes d'écoulement favorisent la création de sites favorables (massifs du Sahara central et méridional, bordure sud de l'Atlas, bordure ouest des montagnes de l'Oman, vallées du Mekran au Pakistan et en Iran).

1.2.3 Le cas de la Mauritanie

La Mauritanie est située en Afrique de l'Ouest et présente trois zones de reproductions identifiées, une zone de reproduction estivale, une zone de reproduction hiverno-printanière et une zone de reproduction mixte (Voir Figure 1.2).

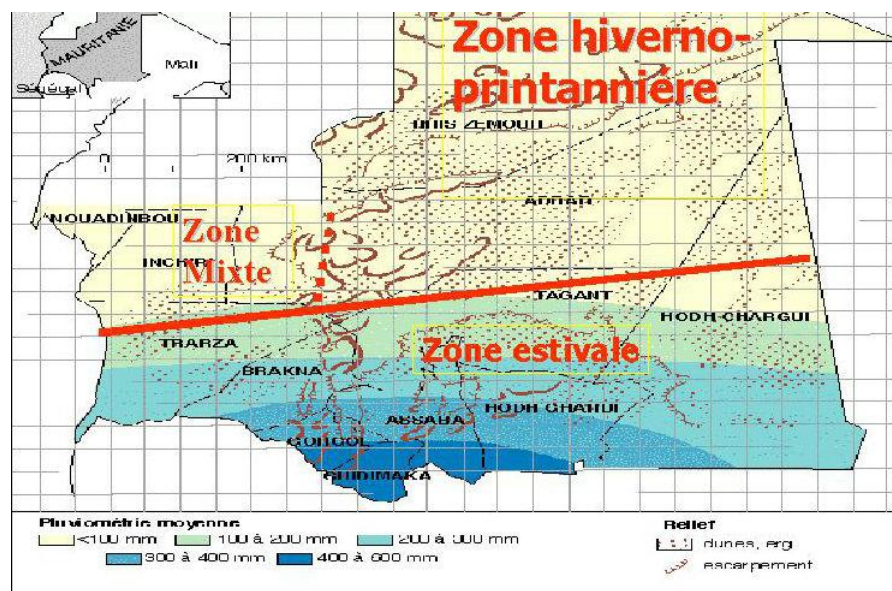


FIGURE 1.2: Les zones de reproduction en Mauritanie (Source : Duranton & Lecoq, Modifié).

La première zone couvre la partie Sud du pays avec un régime pluviométrique tropical

débutant aux mois de juin-juillet et pouvant aller jusqu'en octobre. La deuxième zone couvre les parties Nord et Nord-Ouest du pays, avec un régime pluviométrique à tendance méditerranéenne, débutant aux mois de décembre-janvier et pouvant être précoce à partir d'octobre. Et la troisième zone, à cheval entre les zones estivales et printanières, peut fonctionner à partir du mois de septembre en fonction des aléas météorologiques.

Nous nous intéressons à la Mauritanie parce que ce pays constitue un habitat quasi-permanent pour les criquets pèlerins, plusieurs foyers de grégarisation y sont localisés. C'est un état frontalier du Sénégal et les cas d'invasion acridienne enregistrés au Sénégal et dans les pays environnants sont causés par des individus provenant des zones de reproductions situées en Mauritanie.

Ce pays dispose d'un Centre National de Lutte Anti-acridienne (CNLA) qui est chargé de la surveillance des zones de reproductions situées sur le territoire mauritanien. Le CNLA dispose de données acridiennes qui peuvent servir à valider les modèles développés ici.

1.3 Le cycle biologique du criquet pèlerin

Le cycle biologique du criquet pèlerin peut être résumé en trois étapes successives correspondant aux états *embryonnaire*, *larvaire* et *imaginal* [218]. L'état embryonnaire se passe sous le sol à une profondeur d'environ cinq à dix centimètres (on parle d'état *hypogé*), alors que les deux autres se passent sur le sol, ce sont des états *épigés*. Ces trois états représentent une génération. Le criquet pèlerin effectue deux à trois générations annuelles avec un arrêt de développement facultatif à l'état imaginal, appelé diapause, qui survient généralement, au cours des conditions difficiles comme en saisons sèches [77]. La durée de vie d'une génération est donc de l'ordre d'une cinquantaine de jours. Toutefois, dans les conditions éco-météorologiques défavorables, elle peut atteindre une dizaine de mois, soit par ralentissement du développement sous l'influence de basses températures, soit par arrêt de la reproduction et installation d'une période de diapause [77].

Dans nos modèles, nous supposons des conditions éco-météorologiques favorables à l'évolution et au développement des criquets pèlerins. Ceci permet de considérer les temps minimum de développement des différents stades (référéncé par exemple `maxEggLifePeriod`, pour l'état œuf dans les programmes).

1.3.1 La fréquence des pontes

Les femelles des criquets pèlerins produisent leurs *oothèques* dans des sols humides. Une oothèque est une membrane rigide qui contient les œufs. L'humidité, nécessaire à la ponte, maintient des conditions acceptables pour toute la durée du développement larvaire [77].

Généralement, les femelles pondent peu de fois, la fréquence des pontes varie selon les phases d'existence. Les individus grégaires pondent deux à trois fois par générations

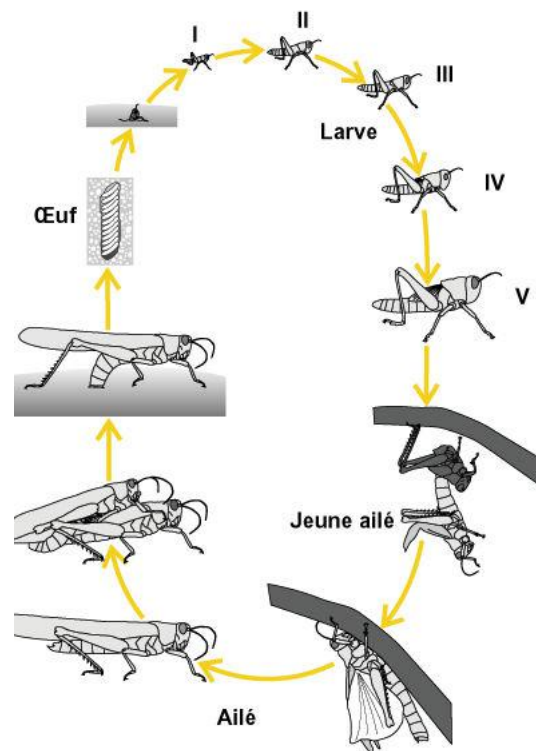


FIGURE 1.3: Le cycle biologique du criquet pèlerin (source : Duranton & Lecoq)

alors que leurs homologues solitaires, ont trois à quatre pontes générationnelles. En moyenne, les femelles pondent deux à trois fois dans leur vie.

Si les conditions éco-météorologiques sont favorables, le nombre d'œufs par oothèque est plus important. La ponte peut durer sept à trente heures et l'intervalle de temps entre deux pontes peut aller de six à onze jours.

En moyenne, les femelles solitaires peuvent pondre cent quarante œufs lors d'une première ponte et les grégaires soixante dix, toujours dans des conditions favorables (voir Tableaux 1.2 et 1.3). Pour les pontes suivantes, ce nombre décroît avec le vieillissement des individus. Par exemple, à la quatrième ponte, il n'est plus que de soixante dix œufs chez les solitaires et de cinquante chez les grégaires.

Nous supposons la quasi totalité des femelles de criquets pondent une oothèque à la première ponte, que 75% d'entre elles environ survivront et pondront une deuxième oothèque, que 25% d'entre elles environ pondront une troisième oothèque, mais que très peu d'entre elles parviendront à pondre une quatrième oothèque. Une moyenne de deux oothèques par femelle et par génération sera retenue dans la suite.

En raison de la mortalité naturelle, tous les œufs n'éclosent pas. De même toutes les larves issues de l'éclosion, n'atteignent pas le stade adulte. Dans des conditions optimales de température et d'habitat, les taux de multiplication des individus peuvent atteindre seize à vingt. En d'autres termes, une femelle peut produire de seize à vingt criquets viables en une seule génération [183].

Dans les tableaux 1.2 et 1.3, P1, P2, P3 et P4 représentent les rangs de la ponte avec P1, la première ponte, P2, la deuxième, ainsi de suite. Le tableau 1.2 représente la variation de la fécondité des femelles dans les conditions éco-météorologiques favorables. Le tableau 1.3 traite les conditions défavorables.

Phases	Moyenne de pontes	P1	P2	P3	P4
Solitaires	+3	140	110	90	70
Grégaires	2-3	70	65	60	50

TABLE 1.2: Nombre moyen de pontes dans des conditions favorables, d'après [183]

Phases	Moyenne de pontes	P1	P2	P3	P4
Solitaires	+3	80	55	–	–
Grégaires	2-3	40	30	–	–

TABLE 1.3: Nombre moyen de pontes dans des conditions défavorables, d'après [183]

1.3.2 Le développement des œufs

On a vu que les œufs se développent en continu à faible profondeur. Ils s'hydratent peu de temps après la ponte en augmentant de volume. Ils ont besoin de leur propre poids en eau pendant les cinq premiers jours suivants la ponte.

Les œufs n'éclosent pas si le sol n'est pas suffisamment humide. Cependant ils peuvent patienter quelques temps après la ponte et terminer leur développement avec les premières pluies (*diapause*). Cette possibilité de diapause est cependant assez limitée car les œufs ne peuvent pas survivre plus de deux mois dans un sol totalement sec. D'ailleurs, le cas d'un sol totalement sec est exceptionnel car les sites de ponte généralement choisis par les femelles des criquets sont dans la majorité des cas suffisamment humides pour garantir une bonne hydratation des œufs [77, 183]. L'éclosion se produit en fin de développement embryonnaire.

Chez les solitaires il n'est pas rare de retrouver des œufs à différents stades de développement au sein d'une même population, alors que chez les grégaires, les éclosions se font en général de manière très synchrones et s'étalent seulement sur deux à trois jours. Toutes les conditions étant égales par ailleurs, les grégaires se développent légèrement plus vite que les solitaires. La durée du développement embryonnaire dépend essentiellement de la température du sol et décroît inversement avec elle si l'œuf a absorbé suffisamment d'eau. Au Sahel, pendant la période de reproduction de mousson située entre les mois de juin et septembre, la durée d'incubation des œufs est en moyenne de treize jours [77].

1.3.3 Le développement des larves

Après éclosion, de jeunes criquets dépourvus d'ailes, passent par cinq ou six stades larvaires successifs selon leurs phases d'existence (figure 1.3). Alors que les grégaires présentent cinq stades larvaires, les solitaires en présentent six. Le facteur qui influe le plus sur le développement des larves est l'humidité du sol, qui permet aussi le développement de la végétation.

À chaque stade, les larves augmentent de taille et se débarrassent de leur enveloppe (*mues larvaires*). La mue imaginale est la dernière mue qui a lieu entre le stade L5 et l'ailé.

La durée du développement larvaire varie essentiellement en fonction de la température de l'air. Par ailleurs, dans les mêmes conditions écologiques, les grégaires se développent plus rapidement que les solitaires. Chez les grégaires, la durée de développement larvaire la plus courte est de vingt cinq jours dans des conditions favorables. Toutefois, elle peut doubler dans des conditions défavorables. Chez les solitaires, la durée du développement larvaire est au minimum de trente jours dans les conditions favorables et elle peut s'étendre sur quatre vingt dix jours dans les conditions très défavorables. Au Sahel, entre les mois de juin et septembre, la durée du développement larvaire est d'une trentaine de jours en moyenne [77].

Dans les modèles ici proposés, les cinq stades larvaires, L1, L2, L3, L4 et L5 des individus grégaires dont le développement se fait en bande ont été considérés. En réalité les différents stades larvaires n'ont pas la même durée de développement, le premier stade est le plus bref et le cinquième est le plus long. En pourcentage, les durées de chaque stade larvaire sont respectivement, du stade L1 au stade L5, de 14, 16, 18, 20 et 32 % de la durée totale du développement, soit approximativement de cinq, cinq, six, sept et onze jours pour une durée totale de développement de trente quatre jours [77].

1.3.4 Le développement des ailés

Les ailés ne muent pas, leur taille ne s'accroît pas mais leur poids augmente progressivement. Leur maturation dure trois semaines, dans les conditions éco-météorologiques favorables et peut aller jusqu'à neuf mois en cas de basses températures et d'habitat aride. En moyenne, elle dure deux à quatre mois.

On a vu que le cycle biologique du criquet pèlerin dure deux à six mois, et que selon les conditions écologiques, deux à trois générations de criquets peuvent exister annuellement avec une possibilité de diapause à l'état imaginal au cas où les conditions sont défavorables [219].

Pour notre simulation, nous supposons que la durée de vie d'une génération de criquet pèlerin dure six mois, répartie ainsi :

- douze jours, pour le développement des œufs ;
- quarante huit jours répartis en six jours pour les stades L1 et L2, neuf jours pour L3, douze jours pour L4 et quinze jours pour L5 ;
- quatre vingt dix jours pour l'apparition des ailés matures ;
- trente jours pour obtenir des individus solitaires ou grégaires.

1.4 La migration des criquets pèlerins

Les changements saisonniers des courants de vent entraînent, selon les saisons, les criquets pèlerins dans des zones où la pluie est la plus probable, dans les limites de la zone de rémission. Ceux-ci les amènent dans des zones du Sahel et du désert indo-pakistanaïsi au cours de l'été et en Afrique du Nord-Ouest, aux rives de la mer Rouge et au Balouchistan au cours de l'hiver et du printemps. Pendant l'automne, ils retournent vers le Nord, zones de basses températures. Les criquets effectuent leur maturation et se reproduisent après les pluies.

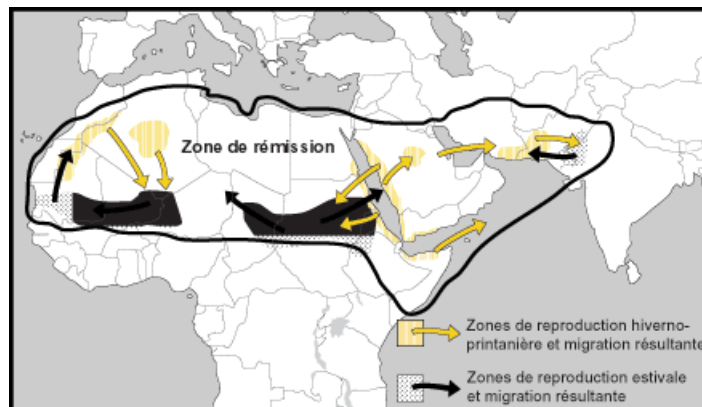


FIGURE 1.4: Déplacement des criquets pèlerins avec le vent, (Source : Duranton & Lecoq

Alors qu'une nouvelle génération d'ailés est capable d'un vol soutenu, le régime des vents peut évoluer, les conditions de reproduction s'avérant médiocres. Dans ce cas, les criquets migrent alors rapidement vers une autre zone, fréquemment sur de très grandes distances. Il faut noter aussi que souvent des déplacements ont lieu au cours de périodes de vents particuliers sans nécessairement coïncider avec le régime des vents dominants.

Ainsi, en automne, en Afrique de l'Ouest, les essaims se déplacent fréquemment vers le Nord et traversent le désert du Sahara, transportés par les vents chauds de secteur Sud associés aux dépressions sur la partie occidentale de la Méditerranée. Les vents de secteur Nord-Est, plus fréquents, sont souvent suffisamment chauds pour permettre le retour des essaims mais cela ne semble pas se produire. Sur la seule base des vents et des températures, les essaims devraient se déplacer vers le Sud et non vers le Nord à partir du Sahel.

1.5 Les données acridiennes

Ces données sont importantes dans l'analyse de la situation acridienne et peuvent aider dans la prédiction des développements des acridiens et à la planification des interventions terrestres. Par exemple les données pluviométriques sont utilisées dans

la stratégie de lutte anti-acridienne pour identifier les zones de pluie récente qui sont favorables à la reproduction des insectes. Les données de température et d'humidité relative sont utiles dans la détermination des vitesses de développement embryonnaire et larvaire et les données de vent sont utiles dans la détermination de la direction des flux migratoires.

Il est important de noter que le nombre de stations météorologiques actives dans les pays situés dans la zone d'invasion est limité, ce qui pose souvent un problème d'hypothèses pour la stratégie de lutte anti-acridienne.

C'est pourquoi nous avons utilisé des données de vent, de pluviométrie, d'humidité relative provenant de fichiers au format "GRIB" que nous avons associé aux données acridiennes. Ces données acridiennes ont été recueillies auprès de l'organisation des nations unies pour l'alimentation et l'agriculture (FAO) sous forme de bulletins mensuels via son service d'information sur le criquet pèlerin (DLIS) durant une période donnée. Ces bulletins présentent un résumé de la situation générale par région et font des prévisions sur six semaines par pays de la zone d'invasion acridienne. Le modèle aussi utilise des données de mortalité sur les différents stades de vie du criquet pèlerin et de leurs durées de vie, recueillies dans un rapport de la FAO [202].

1.6 Conclusion

Ce chapitre a permis de présenter les préférences biologique et écologique du criquet pèlerin, son cycle de vie qui comprend trois principales étapes, ainsi que les migrations saisonnières qui les caractérise. Il a aussi permis de présenter les sources de données qui ont servi à alimenter les modèles développés. Les phénomènes de migration et de reproduction locale seront modélisés en utilisant une approche cellulaire qui est décrite dans le chapitre suivant.

Les notes en italiques ont présenté les choix effectués pour le modèle en regard de l'état des connaissances sur les criquets pèlerins.

2

Eléments techniques pour la modélisation de la dynamique des criquets pèlerins

2.1 Introduction

Le chapitre 1 a présenté les préférences bio-écologiques du criquet pèlerin, les différents stades de son cycle de vie et les migrations saisonnières qui les caractérisent. En appui sur ces considérations, il est possible de procéder à la modélisation du cycle de vie, et à celle de la dynamique des populations. Le but final est de représenter les phénomènes de migration et de développement local observés chez les criquets pèlerins dans leur milieu.

Le point d'appui est le modèle des automates cellulaires (AC), qui va donner à la fois la possibilité de décrire les comportements dans le temps et dans l'espace, en fournissant un cadre d'exécution pertinent grâce à son parallélisme intrinsèque. L'utilisation des AC s'explique ainsi par le fait que ces phénomènes couvrent de très grands espaces et nécessitent de grande capacité de calcul. Ces espaces vont être fractionnés en *cellules* correspondant à des zones géographiques bien définies, où se développent une vie locale. Cette vie, et les migrations qu'elle génère seront représentées par les *fonctions de transition*, qui constituent le programme des cellules. Notre tâche principale sera de représenter les évolutions de groupes d'insectes dans ces fonctions.

Le découpage cellulaire consiste à segmenter les aires de vie des insectes en fragments dont la taille peut être ajustée. Ce découpage a été grandement facilité par *PickCell*, un outil de synthèse de systèmes cellulaires géolocalisés, qui permet de générer des programmes complets, exprimant les modèles en termes de données (cellules) et de comportements (fonctions de transition).

La modélisation cellulaire n'est pas une nouveauté, elle a été utilisée en physique, en

biologie, en mécanique des fluides, en sciences sociales. Elle permet la représentation d'un phénomène complexe par un système mathématique ou informatique régulier, et relativement facile à manipuler. L'évolution peut consister à décrire les interactions entre les variables d'état par des équations [59]. Plus généralement, on se contentera d'utiliser des procédures de langage de haut niveau, encapsulant l'état dans un jeu de variables sur lesquelles la procédure opère.

Un modèle peut être dynamique ou statique selon que les interactions entre les variables évoluent ou non au cours du temps. Ces modèles peuvent être déterministes ou stochastiques selon qu'un état initial conduit à un état final parfaitement connu ou aléatoirement à plusieurs états finaux. Cet aspect aléatoire peut entraîner une variabilité dans les mesures [222]. Dans le cas des modèles stochastiques, une analyse statistique sur le nombre de simulations est nécessaire pour formuler des prédictions alors que dans l'autre cas, la répétition des simulations aboutit à un même résultat si les conditions initiales n'évoluent pas.

Les modèles "*population centrée*" et "*individu centré*" sont aussi des approches utilisées dans la représentation de la dynamique de population. Le premier est relatif à une bande d'individus alors que le second se rapporte à l'individu lui-même. Le modèle "*individu centré*" utilise des règles simples pour chaque individu en tenant compte de ses voisins.

À chaque tour t , ces règles sont appliquées pour déterminer l'état de chaque individu au tour $t + 1$. Le modèle multi-agent est un exemple de modèle individu-centré. Ce type de modèle est adapté au cas où le comportement individuel dépend du comportement de groupe. Les modèles développés dans les travaux [20, 19] sont basés sur des systèmes multi-agents appliqués à des comportements individuels.

De plus lorsqu'on veut modéliser des phénomènes complexes, la simulation séquentielle peut se révéler couteuse, il devient dès lors indispensable de répartir l'exécution du modèle sur une ferme de processeurs ou de machines. Deux approches traditionnelles ont existé jusqu'ici dans la simulation des systèmes complexes, la première consiste à utiliser des ordinateurs puissants, performants en terme de traitement informatique et la seconde, à découper le modèle en sous modèles qui peuvent être exécutés sur des calculateurs différents.

Les processeurs graphiques (GPU) permettent d'effectuer l'exécution d'un grand nombre de processus physiques de manière concurrente. Ils proposent une opportunité pour les simulations massivement parallèles. Cette approche est pratiquée au laboratoire de Brest, tant pour les réseaux de capteurs que les systèmes cellulaires [229, 79]. Les résultats de ces simulations parallèles ont été très encourageants en comparaison d'autres techniques utilisant des processus légers s'exécutant sur des processeurs multi-coeurs dans le langage Occam [124]. Ce dernier type de simulations présente toutefois l'avantage de pouvoir cibler des super-calculateurs [207]. Les dernières expérimentations cellulaires montrent que des GPU relativement modestes (2000 processeurs) permettent de traiter des problèmes associant plus de 100000 cellules en des temps très courts. L'avantage en clarté de spécification revient cependant au langage Occam.

Une autre opportunité de partitionnement des exécutions est offerte par le standard

IEEE "High Level Architecture" (HLA) [65, 229], qui fournit à des simulations indépendantes des mécanismes de communication dans un contexte de systèmes distribués synchronisés [98]. Cette approche est bien adaptée aux systèmes physiques complexes, car elle permet la composition de plusieurs simulations parallèles qui peuvent être exécutées simultanément, et séquencées, avec des phases d'échange de données. Les simulations élémentaires peuvent être menées sur des GPUs, ou sous forme de programmes parallèles distribuables sur des fermes de processeurs. L'interopérabilité entre les simulations physiques est assurée par un mécanisme de fédération décrit dans la norme HLA [229].

L'ensemble formé de populations de criquets dans leur milieu de reproduction en interaction avec les paramètres climatiques et le réseau de surveillance basé sur les réseaux de capteurs sans fil est considéré comme un système physique. Ce dernier est composé de plusieurs entités pouvant interagir entre-elles, par exemple, la végétation d'une zone donnée évoluera en fonction de la quantité de pluie reçue par cette zone. Le modèle climatique lui-même inclut quatre sous-modèles qui sont la température, l'humidité relative, la pluviométrie et le vent. On imagine bien que l'interaction entre ces sous-modèles se traduit par des descriptions complexes, augmentant aussi les charges de simulation qu'une simulation séquentielle supportera difficilement. Les entités intéressantes sont physique (le modèle climatique, la population de criquets), et informatique (le réseau de surveillance synchrone qui échantillonne le monde physique).

La migration d'une zone à une autre est une question centrale qu'il faut comprendre pour expliquer les phénomènes de migration saisonnière. Les criquets migrent à la recherche de conditions favorables à leur survie [218]. En réalité, la pluie, la végétation et le vent influencent fortement la dynamique de population du criquet pèlerin, et l'intégration de ces paramètres dans les modèles constitue un enjeu auquel nous proposons de contribuer. Les simulations par une combinaison d'approches hybrides basée sur des simulations massivement parallèles et distribuées sont réalisées.

La suite de ce chapitre sera structurée comme suit : la section 2.2 sera consacrée à la dynamique de population des criquets pèlerins, dans la section 2.3, nous présenterons l'évolution des automates cellulaires et les raisons de son choix comme modèle physique, la section 2.4 sera consacrée aux représentations spatiale et temporelle du système physique que constituent les populations de criquets pèlerins dans leur milieu soumis aux contraintes climatiques, dans la section 2.5, nous décrirons PickCell, un outil de synthèse de systèmes cellulaires utilisé dans la génération des réseaux de cellules, et la section 2.6 conclura ce chapitre.

2.2 La dynamique de population

2.2.1 Etat de l'art

La dynamique de la population est une science qui étudie la variation dans l'espace et le temps, du nombre d'individus d'une population.

Depuis longtemps de nombreux scientifiques se sont intéressés à cette thématique. Historiquement, dans son *Liber Abaci* (1202), Leonardo Fibonacci formula un problème

relatif à la reproduction des lapins, dont la population mensuelle, $p(n)$, évolue suivant l'équation [94] :

$$p(n+1) = p(n) + p(n-1), \text{ avec } p(0) = 0, \text{ et } p(1) = 1 \quad (2.1)$$

La suite de Fibonacci est un modèle erroné car elle ne tient pas compte, ni de la mortalité, ni du flux migratoire. Mais toutefois, elle a permis de poser les premiers jalons de la dynamique de population.

En 1748, Euler, dans le chapitre sur l'exponentielle et le logarithme de sa publication "Introduction à l'analyse des infiniment petits" [84], plusieurs sujets relatifs à l'évolution de la population furent évoqués. En supposant une population $p(n)$, croissant annuellement suivant la formule :

$$p(n+1) = r p(n) \text{ avec } r > 1 \quad (2.2)$$

Il établit une relation entre le taux de croissance r , le nombre de naissances par habitant et par année N , et la probabilité $q(k)$ d'être en vie k années après sa naissance :

$$1 = N (1 + q(1)/r + q(2)/r^2 + \dots + q(k)/r^k + \dots) \quad (2.3)$$

Le nombre de naissances croît avec le taux de croissance r et la composition de la population est alors de :

- $\mu = NP$ (nouveaux-nés),
- $q(1)\mu/r$ (enfants de un an),
- $q(2)\mu/r^2$ (enfants de deux ans).
- etc...

Equation de croissance exponentielle

Le taux de croissance r , la différence entre les taux de natalité et de mortalité, connaîtra son heure de gloire avec Thomas Robert Malthus(1798), dans son essai sur "l'Essai sur le principe de population" en 1798 [163]. Malthus s'était basé sur deux postulats qui étaient pour lui des lois permanentes de la nature : "La nourriture est nécessaire à l'existence de l'homme" et "la passion entre les sexes est une nécessité". Or, "si elle n'est pas freinée, la population s'accroît en progression géométrique" alors que "les subsistances s'accroissent en progression arithmétique". L'équation de variation de population utilisée par Malthus, avec β et γ , les taux de natalité et de mortalité, est :

$$\frac{dN}{dt} = \beta N - \gamma N \quad (2.4)$$

L'équation de Malthus est inquiétante car elle montre que la population va rapidement épuiser ses subsistances. Ce modèle considère que les taux de natalité et de mortalité sont indépendants de la densité de population. Ce qui n'est pas toujours le cas, car ces paramètres peuvent être influencés par l'évolution de la population. Par exemple, en absence de ressources, les taux de natalité et de mortalité sont directement ou indirectement

influencés. Il avait considéré une population idéale constituée d'une seule espèce animale homogène. C'est à dire qu'il avait négligé dans son modèle les variations d'âge, de taille et de périodicité éventuelle pour la natalité ou la mortalité et qui vit seule dans un milieu invariable ou qui coexiste avec d'autres espèces sans influence directe ou indirecte [103].

Ce modèle suppose aussi que le taux de croissance est le même pour tous les individus de la population, alors qu'en réalité, il dépend de l'âge des individus, ou de leur stade de développement. Ainsi par exemple dans une population de criquets pèlerin, œufs, larves et adultes n'ont pas le même taux de natalité ou de mortalité.

L'équation peut être réécrite ainsi :

$$\frac{dN}{dt} = r N, \text{ avec } r = \beta - \gamma \quad (2.5)$$

Cette équation peut être résolue si des conditions initiales sont spécifiées, c'est à dire, avec N_0 , au temps $t = 0$, la solution est :

$$N(t) = N_0 \exp^{rt} \quad (2.6)$$

Equation de croissance logistique

Verhulst montra en 1838 que le modèle proposé par Malthus était trop simpliste car il ne comprenait que des termes linéaires, et proposa un modèle alternatif en introduisant un terme quadratique avec un coefficient négatif :

$$\frac{dN}{dt} = A N - b N^2 \quad (2.7)$$

Pour expliquer la formule proposée par Verhulst, considérons le cas où le taux de natalité diminue avec une densité de population forte. L'approche la plus simple pour réaliser une telle dépendance, consiste à supposer que le paramètre β , introduit dans l'équation de Malthus, est en fait le taux de natalité dans le cas d'une densité de population très faible.

En outre, si N croit, il est supposé diminuer linéairement avec N pour atteindre 0 à une certaine valeur arbitraire de densité de population $N = \Gamma$. Si, en plus, on suppose que le taux de mortalité est aussi indépendant de la densité, on obtient l'équation suivante :

$$\frac{dN}{dt} = \beta N \left(1 - \frac{N}{\Gamma}\right) - \gamma N \quad (2.8)$$

Cette équation peut être réécrite pour donner :

$$\frac{dN}{dt} = r N \left(1 - \frac{N}{k}\right) \quad (2.9)$$

Le facteur k appelé "carrying capacity" qui correspond à la capacité du milieu à supporter la croissance de la population et représente la population limite au-delà de laquelle elle ne peut plus croître. Cette loi, à laquelle Verhulst donne le nom de logistique,

est radicalement différente de celle de Malthus en ce sens qu'elle impose une valeur limite à la population.

En 1924, Lotka publia les bases théoriques des lois de la dynamique de populations pour la première fois dans son ouvrage "Elements of Physical Biology" [154]. Ce premier modèle basé sur l'approche "prédateurs et proies" permet de décrire l'évolution quantitative de deux populations en interaction.

Ses travaux furent complétés par Volterra, un mathématicien et physicien romain, qui posa les formulations mathématiques à base d'équations différentielles.

Ce modèle considère deux populations de proie et de prédateurs dont les effectifs au temps t sont respectivement notés $x(t)$ et $y(t)$.

Avec les hypothèses suivantes :

- les proies disposent d'une quantité illimitée de nourriture, seuls les prédateurs s'opposent à leur croissance et en l'absence de prédateurs la population de proies croît de manière exponentielle (modèle de Malthus) ;
- le nombre de prédateurs est limité par la quantité de proies dont ils disposent pour se nourrir et en l'absence de proies, la population des prédateurs décroît de façon exponentielle (modèle de Malthus) ;
- le nombre de rencontres entre proies et prédateurs est à la fois proportionnel aux populations de proie et de prédateurs, donc proportionnel au produit $x(t)y(t)$;
- Le taux de disparition des proies ainsi que le taux de croissance des prédateurs dûs à ces rencontres sont l'un et l'autre proportionnels au nombre de rencontres entre les deux populations.

La formulation mathématique donne :

$$\begin{cases} \frac{dx(t)}{dt} = \alpha_1 x(t) - \beta_1 x(t)y(t) \\ \frac{dy(t)}{dt} = -\alpha_2 y(t) + \beta_2 x(t)y(t) \end{cases} \quad (2.10)$$

où α_1 représente le taux de natalité des proies, α_2 , le taux de mortalité des prédateurs, β_1 et β_2 sont les coefficients d'interaction entre proies et prédateurs.

La plupart des modèles mathématiques relatifs à l'évolution temporelle de la population [24], comme ceux de Malthus et de Verhulst [26] ou encore celui de Lotka-Volterra [25], sont valables quel que soit l'âge des individus. Ils sont généralement continus, en ce sens que la variable temporelle peut prendre n'importe quelle valeur dans un intervalle de la droite numérique réelle ; ils utilisent des équations différentielles. Cependant, ils ne sont pas toujours fiables, notamment parce qu'ils négligent les durées de gestation et ne distinguent pas les taux de fécondité selon les âges.

Pour remédier à ces lacunes, Sir P. H. Leslie, un statisticien anglais, construit un modèle tenant compte de la structuration de la population selon plusieurs classes d'âges, et qui fait un usage à la fois simple et efficace du calcul matriciel [150].

En 1945, Leslie publie, dans un article [150], un modèle pour décrire l'évolution temporelle du nombre de femelles dans des populations de souris et de rats. La motivation de cette étude était de faire face à la prolifération de ces rongeurs, pendant la seconde guerre mondiale, qui occasionnaient des dégâts sur les réserves alimentaires de la population anglaise. Ce modèle mathématique n'est pas continu mais discret, dans

le sens où le temps t est uniquement considéré à intervalles réguliers et ne prend dès lors que des valeurs entières.

2.2.2 Le cas des criquets pèlerins

La dynamique de population des criquets pèlerins peut se résumer ainsi : les adultes immatures migrent vers des zones de pluie récente où ils peuvent terminer leur maturation et se reproduire. Cette dynamique peut être affectée par les facteurs climatiques. Cette migration n'est pas toujours effective car les individus peuvent rester, terminer leur maturation dans la même zone si les conditions sont favorables.

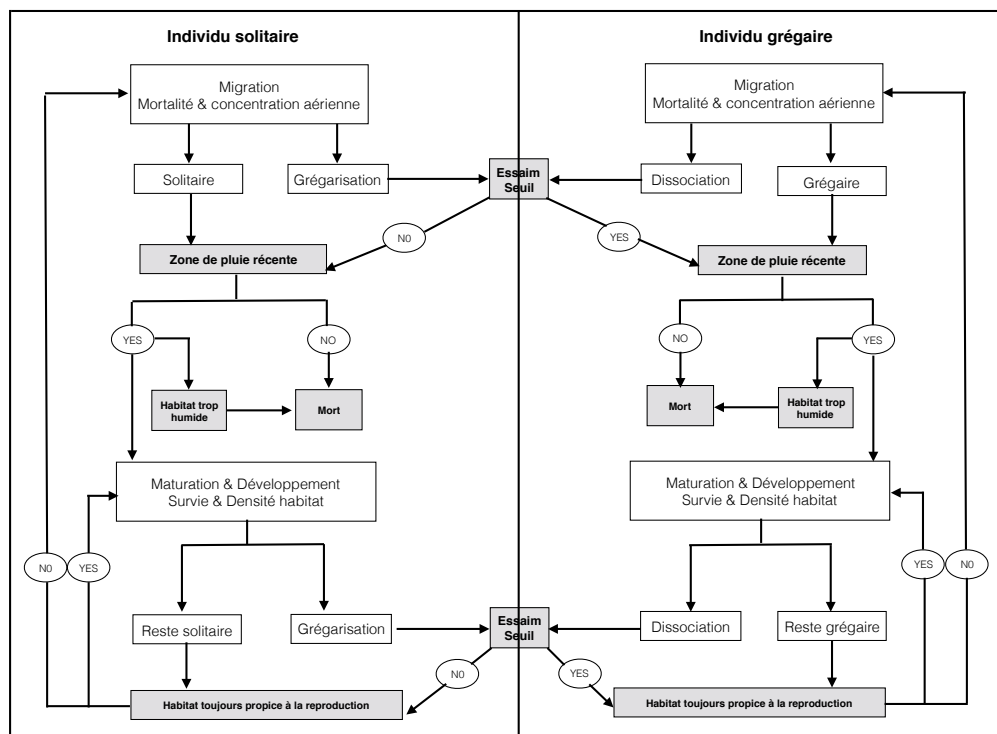


FIGURE 2.1: La dynamique de population du criquet pèlerin (Source : J. Roffey et Al, Modifié [203]).

Des individus solitaires peuvent passer en phase grégaire si la densité de la population atteint un seuil, dans certaines conditions éco-météorologiques. De même que les individus grégaires peuvent se dissocier pour revenir en phase solitaire. L'association et la dissociation des individus passent par une phase intermédiaire (transiens).

Dans notre modèle de migration, quatre flux gouvernent la dynamique de population : la natalité, la mortalité, les immigrés et les émigrés. Dans une cellule, des criquets peuvent provenir des cellules voisines et les femelles peuvent y pondre des œufs. Tous les œufs ne deviennent pas larves et certains individus matures vont migrer à la recherche de meilleures conditions, comme une cellule peut recevoir de nouveaux entrants.

$$\Delta P = \text{Naissance} - \text{Morts} + \text{Immigres} - \text{Emigres} \quad (2.11)$$

Caractérisation de la population

Plusieurs approches organisationnelles peuvent être considérées dans l'étude de la dynamique de population, il s'agit des approches centrées :

- Organe (Ecophysiologie)
- Individu (Ecologie du comportement)
- Population (Dynamique des populations)
- Espèce (Biologie de la conservation)
- Communauté (Ecologie des communautés)
- Paysage (Ecologie du paysage)
- Biosphère (Etude de la biodiversité)

Nous considérons un modèle de population semi structuré ne tenant compte que du sexe des individus.

Les accidents démographiques locaux liés aux conditions du milieu sont rapidement compensés par des apports de migrants, fortement mobiles. Il existe donc un brassage régulier entre les populations portant sur la totalité du domaine de reproduction. Les cellules peuvent accueillir des migrants et laisser partir des individus à la recherche de meilleures conditions, à ce titre, le modèle est basé sur un système ouvert. Il est bien connu que dans un système fermé, les termes immigrés et émigrés sont négligés dans l'équation de la dynamique car il n'y a pas d'échange possible entre les cellules.

Les modèles cellulaires qui ont été développés sont une combinaison de modèles dynamique, stochastique, semi structuré ouvert et centré population. Ce sont des modèles dynamiques à cause des dépendances qui existent entre les facteurs climatiques dans les cellules. Par exemple, le couvert végétal qui fournit refuge et alimentation aux criquets est fortement lié à la pluviométrie. L'utilisation des taux de natalité et de mortalité dans la détermination du nombre d'individus sortant d'une cellule fait de nos modèles des modèles stochastiques. Ces taux varient en fonction du temps et du lieu, ceci a un impact sur les mesures de densité de population.

Nous nous sommes intéressés au comportement de groupes car les essaims de criquets sont constitués de bandes pouvant contenir plusieurs milliers d'individus, de plus les individus grégaires présentent le même comportement dans leur évolution. Ils évoluent de façon synchrone. Nous tenons aussi compte des femelles qui peuvent pondre des oeufs dans les cellules au cours des migrations.

Le monde physique est caractérisé par la concurrence des événements. Cette concurrence devrait se retrouver dans les abstractions des processus et des réseaux. Cette caractéristique apparaît dans nos modèles car si nous prenons chaque cellule individuellement, elle doit exécuter un ensemble de processus durant chaque étape. Ces exécutions se font de manière parallèle. Les processus sont le déroulement du cycle de vie pour chaque état et les migrations d'individus entre cellule.

Pour mieux appréhender cet aspect concurrent et les interactions pouvant exister entre les différents composants du modèle, le découpage en cellule du système physique en

s'articulant sur les automates cellulaires, se positionne aujourd'hui comme une alternative face aux problèmes posés par les modèles basés sur les équations différentielles.

2.3 Les automates cellulaires

2.3.1 Un bref historique

Stanislas Ulam [226] est à l'origine des AC. Ce mathématicien s'était intéressé vers les années quarante à l'évolution des cristaux [180]. Son AC était un espace de cristaux à deux dimensions découpé en cellules. Chaque cellule pouvait avoir deux états, allumé ou éteint. À partir d'un état initial, l'état suivant était déterminé en fonction de règles de transition relatives aux états des cellules voisines. Par exemple, une cellule cible en contact avec deux cellules voisines allumées s'allume, sinon elle s'éteint. Ceci permettait de générer des figures esthétiques complexes [195].

Pendant la même période, John von Neumann se basant sur des travaux de Alan Turing [225], s'était intéressé à la théorie des automates auto-réplicateurs relative à la conception de robot. Cette théorie est à la base de kinématon [231], une machine pouvant générer n'importe quelle autre machine à partir des éléments de son environnement. Ceci avait contribué à la compréhension du fonctionnement de la molécule d'ADN découverte quelques années plus tard.

Les travaux de recherche sur la théorie d'auto-réplication de John Von Neumann avaient donné une autre impulsion aux AC avec la simulation du comportement du vivant par le biais de structures artificielles [51, 173]. Des phénomènes physiques sont représentés de manière discrète spatialement et temporellement [239]. Comme exemple de phénomènes physiques, nous pouvons citer les inondations, les feux de forêt, la diffusion de pandémies, etc.

Les AC génèrent de macro-comportements complexes en utilisant des règles de transitions simples et des structures régulières [238, 240]. Ils sont sortis des laboratoires en 1970 avec le fameux Jeu de la vie de John Horton Conway [141].

2.3.2 Le modèle du jeu de la vie

Nous proposons une description de ce modèle afin d'illustrer l'intérêt des AC pour notre problème. Un espace cellulaire se présente sous forme de grilles rectangulaires. La dimension choisie ici est 5×3 , avec des cellules numérotées horizontalement de 0 à 4 et verticalement de 0 à 2. Les cellules de couleur verte sont à l'état vivant et celles en gris, à l'état mort (voir Figure 2.2).

Le voisinage d'une cellule cible est l'ensemble des cellules contiguës, y compris les cellules diagonales. Par exemple, sur la figure ci-dessous (voir Figure 2.3), la cellule cible Numéro 12 dispose de huit voisines. Donc, c'est un voisinage de Moore de distance unité qui est utilisé ici. Il existe d'autres types de voisinage comme celui de Von Neumann de distance unité où le nombre de voisins est quatre.

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24

FIGURE 2.2: AC, un exemple d'état initial.

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24

FIGURE 2.3: Le voisinage d'une cellule cible.

Parmi les huit voisines de la cellule numéro 12, deux seulement sont dans l'état vivant. Les règles de transition régissant cette dynamique sont définies ainsi :

- une cellule à l'état mort à coté de trois cellules aux états vivant, renaît ;
- une cellule à l'état vivant à coté de deux ou trois cellules aux états vivant, reste à l'état vivant ;
- dans tous les autres cas, la cellule meurt.

Avec ces règles universelles et communes à l'ensemble de l'espace de l'AC, la naissance d'une cellule nécessite un certain nombre de présence autour d'elle. Une cellule isolée (moins de deux voisines) ou ayant une forte concentration de cellules autour d'elle (plus de trois voisines) ne peut pas survivre.

Les AC évoluent de manière discontinue, avec un temps segmenté en étapes. À chaque étape, une cellule évalue son état et décide de changer ou pas d'état pour l'étape suivante. L'aspect parallèle apparait car toutes les cellules évoluent simultanément et de manière indépendante. Les cellules peuvent avoir plusieurs processus à exécuter. Le passage d'un état au suivant se fait lorsque toutes les cellules ont fini d'évaluer leur état.

Sur la figure qui suit (voir Figure 2.4), le nombre de voisines vivantes de chaque cellule est indiqué

1	2	3	2	1
1	1	2	1	1
1	2	3	2	1

FIGURE 2.4: Nombre de voisins des cellules.

- les cellules 00, 04, 10, 14, 20 et 24 sont à l'état mort et ont une voisine vivante, donc elles restent à l'état mort ;
- les cellules 01, 03, 21 et 23 sont à l'état mort et ont deux voisines à l'état vivant, donc elles restent à l'état mort ;

- les deux cellules restantes à l'état mort (02 et 22) ont trois voisines à l'état vivant, donc elles deviennent vivantes ;
- les cellules 11 et 13 sont dans l'état vivant et n'ont qu'une voisine à l'état vivant, elles évoluent vers l'état mort ;
- enfin la cellule 12 dans l'état vivant avec deux voisines dans le même état, reste en vie.

À l'étape suivante, seules les cellules 02, 12 et 22 seront donc actives.

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24

FIGURE 2.5: Etat suivant.

Le jeu de la vie représente un point de départ intéressant pour construire des simulateurs dans le domaine de la physique et du vivant. Il faut noter que ce n'est qu'un exemple d'AC parmi plusieurs. Il est possible de jouer sur l'ensemble des règles qui régissent son univers ou sur le nombre d'états pour avoir d'autres types d'AC.

2.3.3 Les dimensions d'un AC

La dimension d'un AC influe sur le nombre de combinaisons à exécuter. Par exemple une augmentation des dimensions fait croître le nombre d'automates possibles. Il est possible de créer des automates à trois dimensions voire plus. En considérant le cas d'un voisinage à trois (la cellule cible et ses voisines de droite et de gauche) dans un AC à une dimension et deux états, il existerait 2^8 , soit 256 combinaisons possibles.

La représentation des automates à une dimension, utilise souvent une seconde dimension pour représenter le temps. Il est aussi possible de jouer sur le voisinage.

Avec un AC, les règles sont simples et complètement connues. Ce qui permet de tester et analyser le comportement global d'un univers simplifié.

Avec les AC à deux dimensions, les topologies de voisinage les plus courantes sont :

- Von Neumann , les voisins Nord, Sud, Est, et Ouest.
- Moore, avec les diagonales comme dans le cas du Jeu de la vie.
- Moore étendu, avec une extension de la distance de voisinage au-delà de un.
- Margolus, avec des ensembles de 2×2 cellules éventuellement alternés.

Le voisinage de Margolus est quant à lui utilisé dans la simulation du comportement des gaz. Par exemple, l'AC de Fredkin utilise un voisinage de Moore. Il est basé sur la parité du voisinage [95]. Il est de type sommatif, c'est-à-dire que l'état des cellules dépend du nombre de voisines actives, indépendamment de leur position. La reproduction ne se passe que si la valeur de voisinage est impaire. Cet automate a la propriété remarquable de reproduire toute configuration de base en neuf exemplaires.

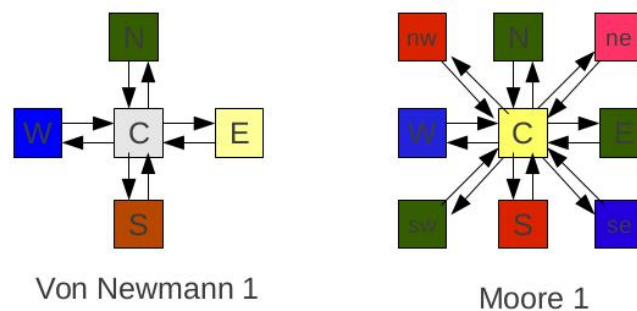


FIGURE 2.6: Le voisinage de Von Neumann 1 présente quatre voisins correspondant aux quatre directions cardinales alors que celui de Moore 1 présente tous les voisins autour d'une cellule à la distance 1.

2.3.4 Les domaines d'applications des AC

Les AC peuvent être utilisés dans plusieurs domaines pour simuler :

- le comportement d'un gaz ;
- les feux de forêts et la propagation des épidémies ;
- les processus de cristallisation ;
- le développement urbain ;
- etc...

Dans la modélisation de la dynamique de populations, les modèles AC s'avèrent bien adaptés et flexibles. En particulier ils permettent d'intégrer les interactions entre les différents composants de l'univers [126].

2.3.5 Autres exemples d'AC

Les automates stochastiques intègrent des notions de probabilité dans les règles de transition. D'une manière générale, il est possible de créer plusieurs variantes d'automates en jouant sur les règles structurelles et fonctionnelles. Les premières définissent la structure spatiale liée aux dimensions, aux arrangements des cellules (rectangle, carré) et à la détermination du voisinage. Les secondes déterminent les règles de transition et le nombre d'états [179]. Le nombre d'états ne se limitent pas seulement à deux états, par exemple, l'AC présenté par Brian Silverman en 1984 utilise trois états (vivant, fantôme et mort) dans la génération de configurations graphiques complexes [86].

2.4 Modélisation cellulaire du système physique

Nos modèles cellulaires discrets au sens spatial et au sens temporel représentent des systèmes distribués formé d'un réseau maillé de processus physiques représentant les

cellules. Ces processus communiquent de manière synchrone. Les cellules coordonnent leurs actions par échange de messages représentant les flux migratoires d'individus ailés.

Le système physique qui est la zone de reproduction estivale des criquets pèlerins, situé en Mauritanie et soumis aux paramètres climatiques peut ainsi être découpé en cellules de dimension finie pouvant évoluer, et échanger selon des règles de transition définies en fonction des conditions météorologiques dans les cellules.

2.4.1 L'échantillonnage spatial

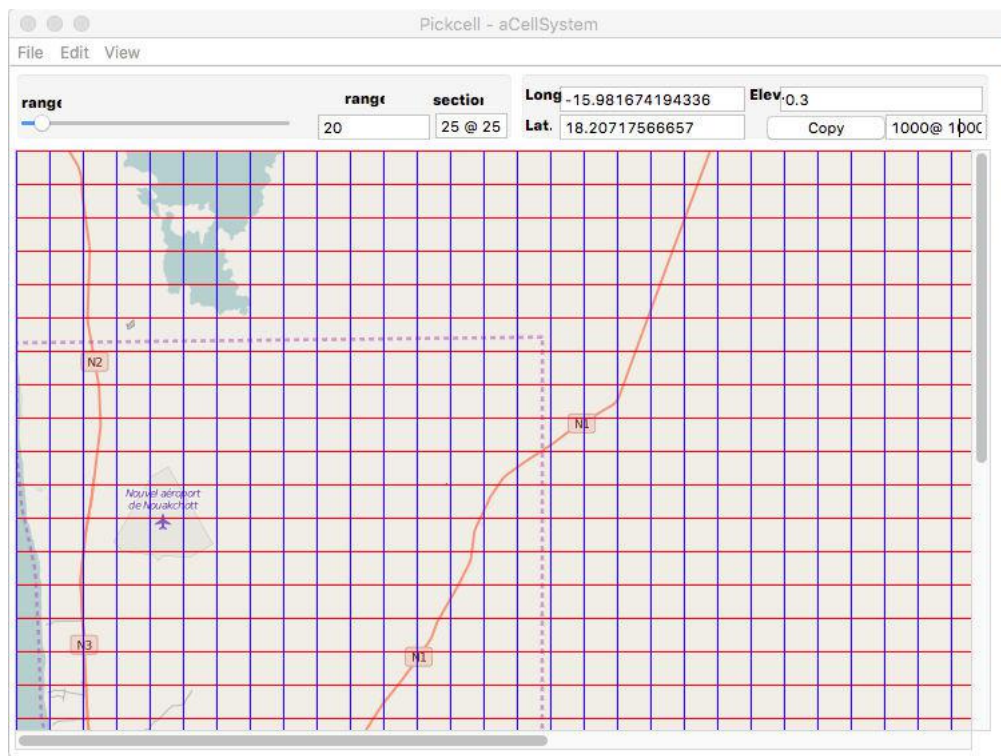


FIGURE 2.7: Découpage spatial du domaine d'étude sur une superficie de 100 ha

La taille d'une cellule est définie à l'aide d'un paramètre du générateur de réseaux cellulaires sous-jacent à l'outil interactif. La figure 2.7 montre une segmentation géométrique de 25×25 pixels, la transposition géographique correspond à une superficie de 1000×1000 mètres carrés.

Les coordonnées géographiques (longitude et latitude) et les élévations des zones sont présentes dans les cellules. Elles permettent une association précise avec des aires de reproduction, et de garder l'historique des pullulations simulées dans un système d'information. L'altitude est un élément qui entre dans l'appréciation des comportements climatiques.

Dans les cellules nous plaçons des individus initiaux qui vont évoluer puis se propager dans différentes directions en fonction des conditions climatiques. Il s'agit de choisir les

circonstances de départ dont on souhaite apprécier les évolutions. Avec ces individus, les cellules disposent de variables représentant le vent, la pluie et la température.

Selon le paradigme des AC, l'ensemble des variables internes à la cellule constitue son état. A chaque tour, chaque cellule examine cet état au temps t , échange des informations avec les cellules voisines également au temps t et évolue vers un nouveau état au temps $t + 1$.

2.4.2 L'échantillonnage temporel

Dans un tel système, le temps réel est critique. Il permet de déterminer les délais relatifs à un événement particulier. Par exemple, l'éclosion des œufs dans une cellule ou l'apparition des premiers ailés sont deux événements importants. Le premier nous permet de savoir qu'une dynamique va débuter dans une zone. Si des conditions favorables sont maintenues, les individus vont mener à terme leur développement pour donner des ailés. Ces derniers constituent une autre étape car ce sont eux qui vont migrer vers les cellules voisines si les conditions sont favorables. Les études portant sur l'espèce fournissent des informations sur le séquençement en temps vrai de ces évolutions.

Lors des simulations, nous raisonnons en terme de *tours synchrones* et non en termes de nombre de jours. Les tours sont exécutés par toutes les cellules. La synchronisation ne repose pas sur l'existence d'une horloge commune mais sur un temps abstrait constitué par les échanges d'informations entre cellules qui intègrent les migrations. L'étude entomologique nous a amené à choisir un tour synchrone correspondant à trois jours (voir Tableau 2.1). Cette valeur représente le plus grand commun diviseur entre les différentes durées moyennes de développement des différents stades.

Durée synchrone	Nombre de jours	Nombre de tours synchrones
MaxEggsLifePeriod	12	4
MaxLarvae1LifePeriod	6	2
MaxLarvae2LifePeriod	6	2
MaxLarvae3LifePeriod	6	2
MaxLarvae4LifePeriod	9	3
MaxLarvae5LifePeriod	12	4
MaxWingedLifePeriod	90	30
MaxSolitariousLifePeriod	15	5
MaxGregariousLifePeriod	90	30

TABLE 2.1: Les échantillons temporels

Lors de la réalisation de l'AC, il faut affecter des constantes ou des variables de temps aux délais séparant les événements. Ainsi, MaxEggsLifePeriod représente la durée de développement des œufs, qui est de douze jours, ce qui correspond à quatre tours synchrones. Les durées de développement des trois premiers stades larvaires MaxLarvae1LifePeriod, MaxLarvae2LifePeriod et MaxLarvae3LifePeriod sont de six jours ou deux tours synchrones. le quatrième stade, MaxLarvae4LifePeriod, dure neuf jours et

le dernier, `MaxLarvae5LifePeriod`, douze jours. La durée de vie des ailés immatures, `MaxWingedLifePeriod`, est de quatre vingt dix jours correspondant à trente tours synchrones. La durée de vie des individus solitaires, `MaxSolitariousLifePeriod`, est de trente tours synchrones correspondant à quatre vingt dix jours et celle des individus grégaires, `MaxGregariousLifePeriod`, de quatre vingt dix jours.

2.4.3 Les règles de transition

Nous disposons de deux modèles cellulaires paramétriques.

- le premier est relatif au développement local dans lequel une population d'oeufs est placé dans le premier micro-état d'une cellule. Un micro-état est une subdivision de l'état d'une cellule en `eggsLifePeriod`, `larvae1LifePeriod`, `larvae2LifePeriod`, `larvae3LifePeriod`, `larvae4LifePeriod`, `larvae5LifePeriod`, `wingedLifePeriod`, `solitariousLifePeriod` et `gregariousLifePeriod`.

Par exemple dans le micro-état `eggsLifePeriod`, les oeufs vont évoluer suivant les taux de mortalité vers d'autres micro-états. Les règles de transitions sont basées sur la durée de vie de chaque micro-état, c'est à dire qu'un certain nombre d'oeufs passeront à l'état larves lorsqu'ils auront consommé toute la durée du `MaxEggsLifePeriod`.

- Le second modèle est une représentation de la migration que les ailés immatures issus du premier modèle vont effectuer si les conditions s'y prêtent. Les règles s'appuient ici sur les paramètres climatiques. Pour cela, nous avons supposé que dans les conditions favorables, la température dans les cellules est comprise entre 15 et 30°C, que la pluviométrie est comprise entre 50 et 300 mm. Dans ces conditions les individus restent dans la cellule et poursuivent leur développement sinon, ils migrent. Les règles de transitions sont liées aux conditions météorologiques dans les cellules, à la vitesse et la direction du vent.

2.5 *PickCell*, outil de génération de systèmes cellulaires

PickCell permet la représentation abstraite d'un système cellulaire sous forme d'un graphe de cellules nommées et de liens de communication. Le réseau est décrit dans une représentation textuelle qui reflète l'organisation du système cellulaire (voir Annexe A). Chaque ligne fournit des informations sur les attributs d'une cellule (élévation, coordonnées géométrique et géographique, données climatiques), les noms du processus et de la procédure exécutée par les cellules. Elle fournit également des informations sur les cellules voisines directement accessibles par les liens sortants, le nom du système cellulaire et une liste de messages circulant sur les canaux de communication.

Le flot méthodologique (voir Figure 2.9) suit un canevas conventionnel dont le point de départ est une vue graphique provenant de cartes, d'images satellitaires obtenues avec QuickMap [156], un navigateur de cartes également développé à l'Université de Brest. Il consiste à segmenter une vue graphique et à la classer selon des critères basés sur les composantes rouge verte et bleue (RVB) des couleurs.

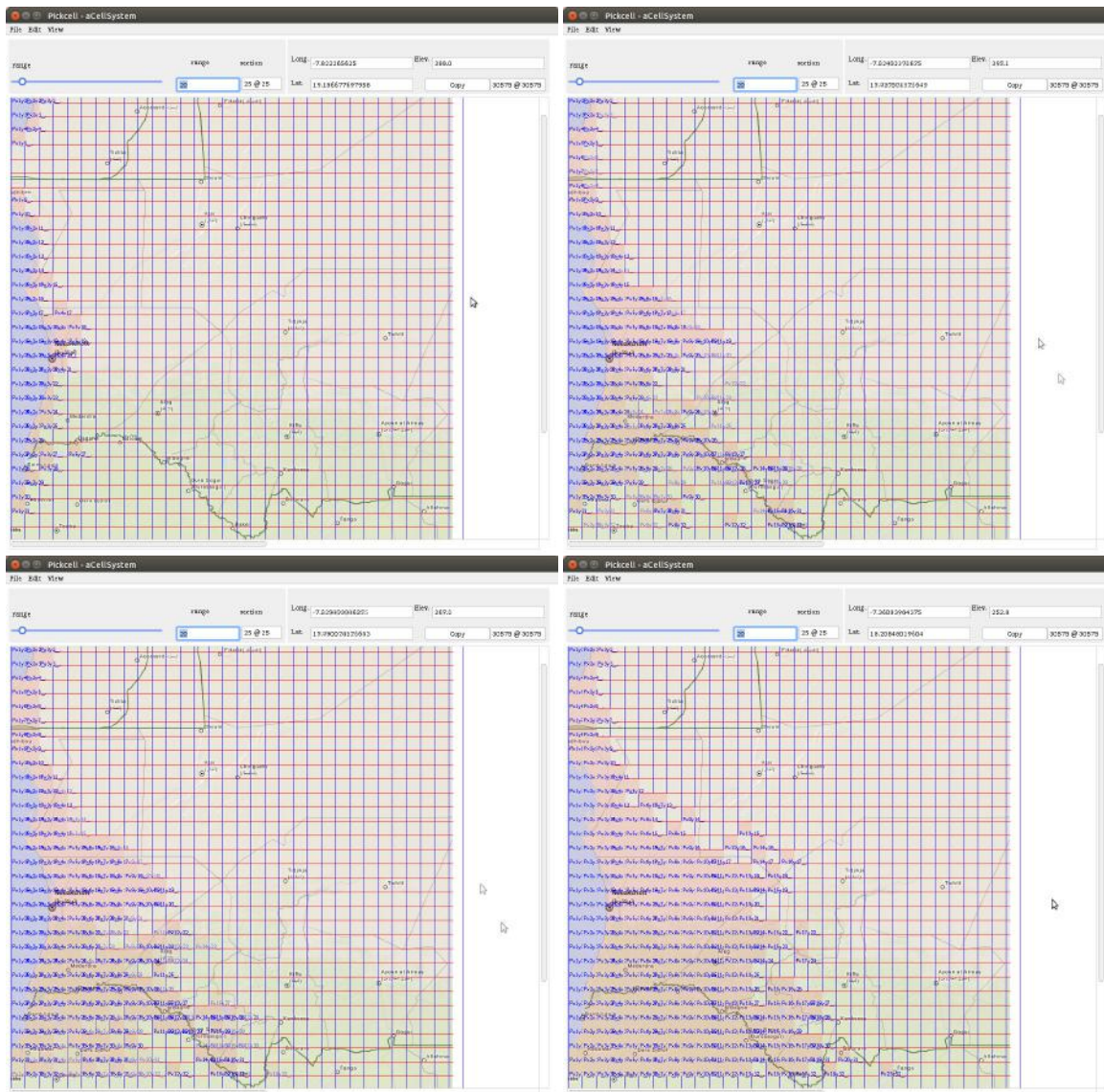


FIGURE 2.8: Ces figures représentent la migration des individus dans les cellules

Une des fonctionnalités principales de *PickCell* est d'extraire des propriétés des systèmes physiques des domaines géographiques. Ces propriétés peuvent être décodées dans une carte ou une photo, ou obtenue d'une base de données géographique. On peut citer l'exemple des aires de reproduction des insectes, les étendues de forêts ou de rivières.

PickCell permet, par exemple, de segmenter une image et de classer les différents segments obtenus en fonction des couleurs RVB. Cette classification consiste à regrouper des pixels en fonction des couleurs pour former des régions homogènes.

Le regroupement des classes obtenues est une reconstitution de l'image originale. Les réseaux cellulaires peuvent donc être compacts, ou arbitrairement irréguliers.

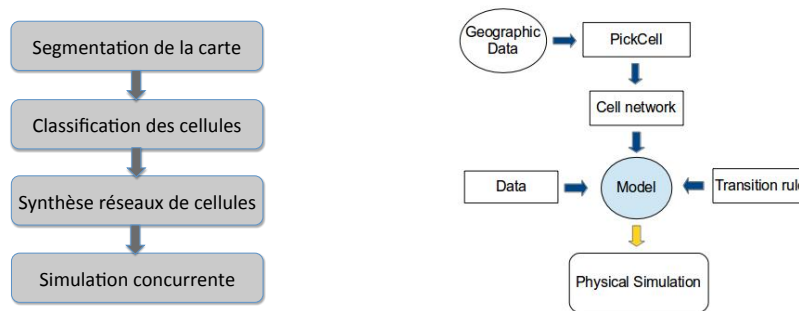


FIGURE 2.9: Le canevas de génération de systèmes cellulaires

2.5.1 Génération d'un maillage de système cellulaire

L'opération consiste à associer une signature à chaque cellule et l'ensemble des cellules ayant la même signature est regroupé dans une classe identifiée par celle-ci. Le système cellulaire est décrit sous la forme d'un réseau de processus où chaque processus dispose d'un ensemble de voisins dont le nombre dépend de la topologie de voisinage utilisée (Von Neumann ou Moore).

D'abord, les classes sont balayées ligne par ligne et colonne par colonne pour permettre la création d'un système cellulaire dans lequel chaque cellule est associée à un processus. Ensuite, la classe est encore balayée cellule par cellule. Pour chaque cellule cible, les positions des voisines sont recherchées dans la classe. Pour chaque voisin, des liens sont établis avec la cellule cible. Ainsi un réseau de cellules est créé en se basant sur le voisinage de la classe (voir Annexe B).

2.5.2 Algorithme de classification cellulaire par couleurs

Les images intégrées dans les cellules sont un élément de classification possible, parmi d'autres critères, l'élévation, par exemple. La classification cellulaire repose sur un algorithme qui calcule le minimum, la moyenne ou le maximum sur les trois composantes RVB d'une couleur (voir détail en Annexe C).

Un partitionnement [*minimum, maximum*] sur chacune des composantes est effectué en appliquant un diviseur.

Pour chaque cellule, un critère basé sur le minimum, la moyenne ou le maximum de pixels par cellule est appliqué, ceci permet de choisir le cube RVB où se trouve la cellule représentant la classe. Ainsi pour chaque classe, un plan cellulaire est construit.

Regroupement des cellules en classe

Après une segmentation de l'image, l'espace de cellules est divisé selon une méthode qui consiste à calculer la moyenne des pixels pour les trois composants RVB (moyenne rouge, moyenne verte et moyenne bleue).

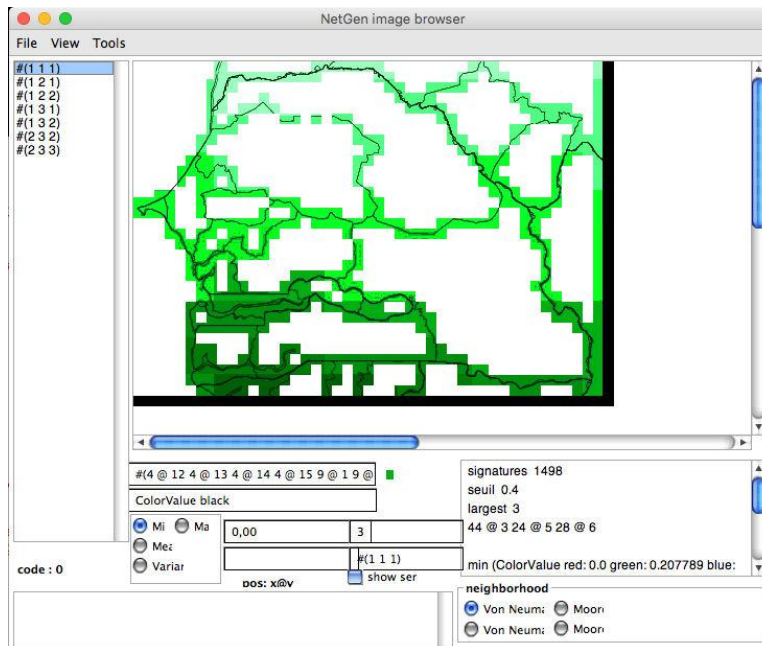


FIGURE 2.10: Un exemple de distribution d'image en sept classes

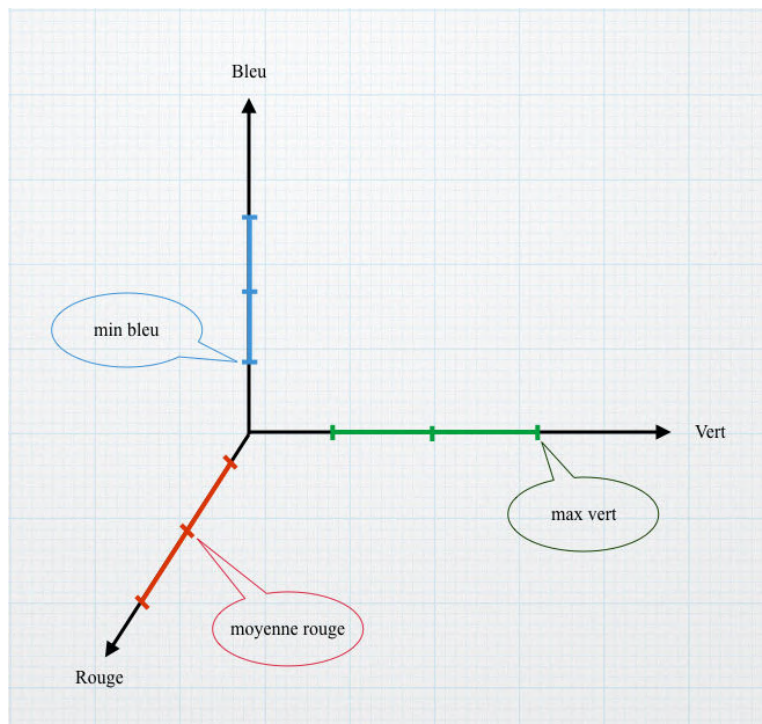


FIGURE 2.11: Diviseur 2, 8 pour un espace de valeurs de $256 \times 256 \times 256$

On obtient un ensemble de 3×3 paramètres dans chaque cellule, permettant alors de calculer les caractéristiques globales de l'image.

Après cette étape, pour chaque composante dans l'espace des couleurs, [min,max] mesures globales sont obtenues. Les valeurs calculées dans chaque cellule sont utilisées pour regrouper les cellules par affinité.

Pour N égal à 2, les valeurs minimum(rouge, vert, bleu), maximum(rouge, vert, bleu) ou moyenne(rouge, vert, bleu) dans chaque cellule permettront de classer chaque cellule de manière unique dans un espace de dimension $2^3 [(R_0, R_1) \times (V_0, V_1) \times (B_0, B_1)]$. Le cube de coordonnées va de 0 (0,0,0) à 7 (1,1,1).

En résumé, *PickCell* permet de produire une signature pour chaque cellule et de regrouper les cellules par affinité. Chaque cellule dispose d'une image et d'un vecteur de paramètres extensible. La fonction majeure de cet outil est la reproduction des systèmes cellulaires en *programmes concurrents* où chaque processus représente une fraction géo-localisée de l'espace étudié.

2.6 Conclusion

Ce chapitre a permis de présenter les principes suivis pour la simulation de la dynamique de population des criquets pèlerins. La modélisation des distributions d'individus, et l'évolution de ceux-ci est fondée sur les automates cellulaires correspondant à des fragments géographiques. Un second niveau de concurrence a été introduit, interne aux cellules, avec des micro tâches représentant les évolutions des étapes du cycle des criquets. Ce niveau va être détaillé et justifié au chapitre 3.

En pratique, beaucoup de données, dont les paramètres climatiques peuvent aussi être intégrés aux cellules.

Cette approche cellulaire permet de coupler le monde réel aux systèmes d'information à condition d'automatiser l'observation de ce monde. Elle peut donc être rapprochée des systèmes cyber-physiques.

Nous avons aussi ici un modèle complexe de grande dimension, intégrant la concurrence entre ses composantes géographiques, et la concurrence interne à la vie des insectes.

Des règles de transition simples ont permis de simuler et d'interpréter l'impact des paramètres climatiques sur la dynamique de population locale du criquet pèlerin.

Ce chapitre a permis aussi de présenter *PickCell*, un outil frontal de synthèse de systèmes cellulaires, permettant la génération de réseaux de processus en vue d'exécutions à haut niveau de parallélisme.

3

Accès aux sources de données géolocalisées

3.1 Introduction

Aujourd'hui, il est vital de disposer de données structurées fiables car la maîtrise et le contrôle des catastrophes naturelles passent par une disponibilité des données. C'est ce qui a poussé plusieurs organismes nationaux et internationaux à s'intéresser à la publication de bases de données pouvant être privées ou publiques. Cependant il faut souligner le manque de standards internationaux dans la description de leur format.

Quels sont les formats de données les plus utilisés, comment peut on faire pour assurer l'interopérabilité entre ces différents formats et dans quelle mesure peut on les intégrer dans les modèles cellulaires et simulations physiques ? Un ensemble d'interrogations auquel nous répondrons dans ce chapitre.

Avec l'amélioration des techniques informatiques de gestion et diffusion des données, des règles juridiques, les sources de données deviennent de plus en plus nombreuses et plus accessibles au public.

En effet, de nombreuses sources de données reflétant des points de vue différents peuvent exister dans un même espace géographique. La différence se trouve au niveau du contenu thématique, de la qualité et des choix de présentation des données.

Les promoteurs de bases de données s'intéressent aussi à l'interopérabilité des sources de données afin d'améliorer les processus de création, de maintenance et de mises à jour parce qu'ils disposent souvent de plusieurs bases de données avec différents niveaux de détail.

La propagation des mises à jour d'une base à une autre peut être fastidieuse du simple fait que les données mises en jeu ne respectent pas le même format.

Par ailleurs, les promoteurs de données jouent de plus en plus un rôle d'intégrateur de données produites par différentes organisations. Cette intégration s'avère nécessaire pour assurer une certaine cohérence entre les données de sources diverses et variées. UMS-3468 bases de données biodiversité, écologie, environnements et sociétés (BBEES) est un exemple de support technique et scientifique pour les chercheurs du muséum national d'histoire naturelle et de l'institut écologie et environnement du CNRS [1].

Consciente de la nécessité de l'intégration des données, l'Union Européenne (UE), à travers la directive INSPIRE [38] a formalisé les problèmes liés à la diversité des formats de données par la mise en place d'une infrastructure d'information géographique. À cet effet, trois directives connues sous le nom de triangle juridique ont été émises :

- la directive 2003/98/CE du 17 novembre 2003 dite Public Sector Information directive (PSI) relative à la réutilisation libre des données publiques [190] ;
- la convention d'Aarhus, signée en juin 1998 relative à l'accès de l'information environnementale [191] ;
- et de la directive INSPIRE 2007/2/CE du 14 mars 2007, sur le partage de l'information géographique publique entre autorités publiques [38].

De même, les Etats Unis (EU) ont mis en place des mécanismes de diffusion de données par le biais d'outils de technologies de l'information et de la communication.

Ces deux exemples illustrent bien le fait que les pays occidentaux ont compris les enjeux liés à la gestion des données, à leur intégration et leur disponibilité et se sont attelés à poser un cadre juridique et des infrastructures de partage.

Le but de ce chapitre est de procéder à une classification des sources de données environnementales selon qu'elles soient nationales ou internationales, privées ou publiques, de ressortir les formats de données utilisés par les promoteurs et de présenter des exemples d'intégration de données avec les modèles cellulaires.

Il existe un projet, dénommé "Open Data Inception" qui présente une liste de portails Open Data. À ce jour 2600 portails Open Data géolocalisés sont disponibles à travers le monde.

La suite de ce chapitre sera organisée comme suit : la section 3.2 présentera un état de l'art sur les promoteurs de bases de données environnementales, la section 3.3 sera consacrée à la correspondance objet-relationnel et présentera un exemple de modélisation d'une base de données, la section 3.4 portera sur un exemple d'intégration d'une base de données avec une application "smalltalk" et la section 3.5 conclura ce chapitre.

3.2 Les promoteurs de bases de données

Les promoteurs de bases de données sont des organisations internationales, nationales pouvant être privées ou publiques qui mettent à disposition des sources de données.

3.2.1 Les initiatives internationales

La Banque Mondiale La banque mondiale est l'un des principaux promoteurs de sources de données dans le monde. Les données portent entre autres, sur l'économie,

l'environnement, l'agriculture, etc. . .

L'accès à ces bases de données est possible à travers une initiative dénommée "Open Data". Elle vise à fournir à tous les usagers un accès selon des conditions d'utilisation. Leur catalogue de données répertorie des bases de données, des tableaux préformatés, des rapports et d'autres ressources [3].

Les formats de données proposés sont des "Statistical Data and Metadata eXchange" (SDMX), "Comma-Separated Values" (CSV), "Tab-Separated Values" (TSV). Les deux derniers, sont des formats ouverts, représentant les données tabulaires sous forme de valeurs séparées respectivement par des virgules ou tabulations.

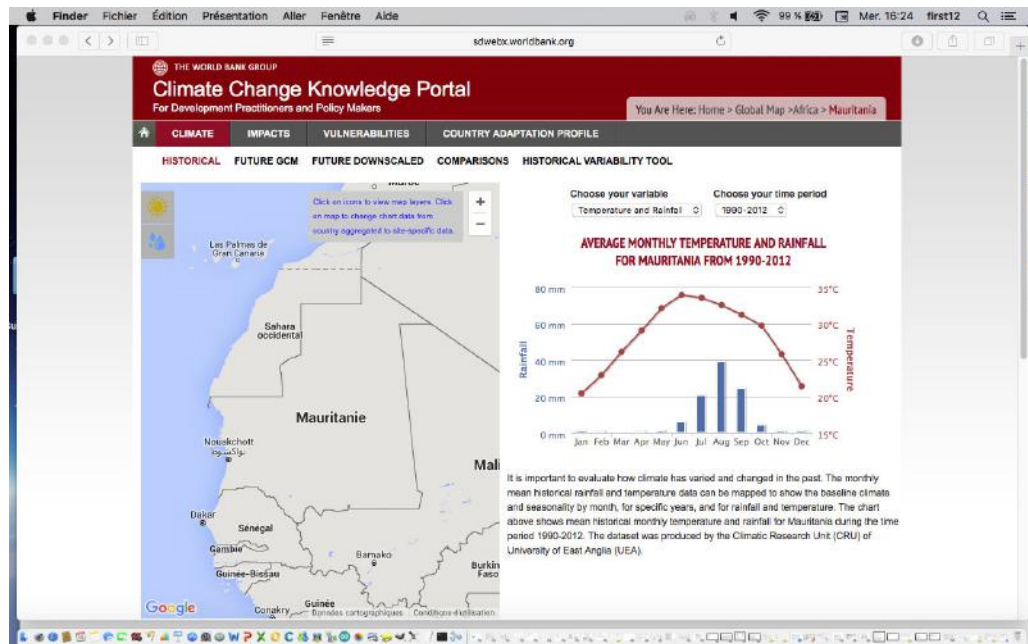


FIGURE 3.1: Les moyennes mensuelles de température et de pluviométrie, en Mauritanie durant la période de 1990 à 2012 [3]

Cette zone (voir Figure 3.1) est un milieu de reproduction naturelle des criquets pèlerins à cause de son climat. Elle est située dans la zone tropicale avec deux saisons, une saison sèche qui dure de Novembre en Avril et une saison de pluies, entre les mois de Mai et Octobre. Ces valeurs de température et d'humidité relative peuvent être considérées dans les modélisations cellulaires et simulations physiques.

En couplant des modèles climatiques aux modèles de migration et de reproduction locale, un modèle générique comme sur la figure (voir Figure 3.2) peut être utilisé pour les modélisations comme dans le domaine météorologique.

Organisation des nations unies pour l'alimentation et l'agriculture (FAO) La FAO est aussi un promoteur de données couvrant un large éventail de thématiques. Des données sur la sécurité alimentaire, l'agriculture, la pêche, la gestion des terres et

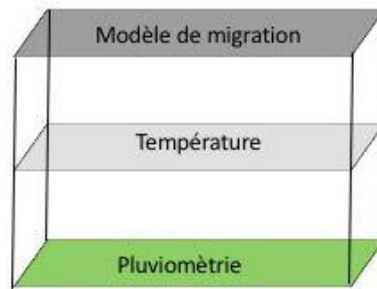


FIGURE 3.2: Couplage de modèles de climat et de migration

des ressources en eau avec des mises à jour mensuelles régulières sont publiées via la plateforme FAOSTAT [72].

Cette plateforme fournit en accès libre et simple aux données statistiques relatives à 245 pays et 35 zones régionales de 1961 à nos jours sur plusieurs thèmes comme la population, la production agricole, la sécurité alimentaire [73]. Ce dernier thème couvre le phénomène acridien.

AQUASTAT, un volet de FAOSTAT met à disposition des ressources aquatiques de plusieurs pays. Il s'agit des ressources en eau souterraine, de surface et de pluviométrie. Ces ressources aquatiques peuvent constituer des points d'entrée pour les modélisations et simulations de cours d'eau. Les données sont fournies sous format de tableaux CSV et TSV, téléchargeables depuis la plateforme [75].

3.2.2 Les initiatives internationales privées

Quelques organismes internationaux privés s'intéressent aussi à la promotion de sources de données environnementales, parmi lesquels, on trouve :

- International Research Institute (IRI), une coopération entre le bureau du programme climatique de la National Oceanic and Atmospheric Administration (NOAA) et de l'Université de Columbia aux Etats Unis [123] ;
- GeoNetwork, de la fondation Geospatiale "Open Source" [2] ;
- OneGeology, une initiative internationale impliquée dans les services géologiques.

L'International Research Institute (IRI) IRI développe mensuellement un modèle de prévision de précipitations et de température appelé General Circulation Model (GCM). Ce sont des prévisions climatiques saisonnières et mi-saisonnières à temps réel pour une meilleure prise de décision et une bonne planification.

Ces informations climatiques permettent de développer des indicateurs et de comprendre les processus impliqués dans la variabilité décennale naturelle et les changements climatiques régionaux dans plusieurs endroits parmi lesquels, le Sahel et l'Amérique du Sud. Elles permettent aussi de fournir des analyses de la prévision saisonnière des quantités de précipitations à l'échelle locale comme la fréquence quotidienne de précipitations

et leur intensité moyenne.

L'IRI détecte et corrige les biais individuels systématiques des CGM des dernières années, puis consolide les prévisions du CGM en une prévision de probabilité unique. Les produits de prévision sont des cartes montrant des régions ayant une forte probabilité d'être plus humides ou sèches, plus chaudes ou plus froides que la normale pour les saisons à venir.

Ces informations sont fournies sous forme de répertoire de données en ligne libre d'accès. L'IRI propose aussi des outils d'analyse qui permet de visualiser, d'analyser et de télécharger des centaines de téraoctets de données liées au climat.

Une grande variété de données en tout type de format (fichiers de données, tables), de tracé visuel (contour, couleurs, vecteurs), et de format graphique (PostScript, GIF) est disponible.

Les formats de données disponibles dans la bibliothèque de données sont de type :

- ascii avec entête ArcInfo pour ArcInfo, un outil SIG propriétaire ;
- NetCDF (Network Common Data Form), un ensemble de bibliothèque de formats de données qui permet la création, l'accès et le partage de données scientifiques ;
- des tables ;
- des images JPEG, GIF et PS.

GeoNetwork GeoNetwork propose une interface web permettant d'accéder à des données géospatiales issues de plusieurs sources de données externes. La plateforme permet :

- d'améliorer l'accès et l'usage des informations et données spatiales ;
- d'aider à la prise de décision ;
- de favoriser l'approche multidisciplinaire nécessaire au développement durable ;
- d'améliorer la compréhension des apports de l'information géographique.

La recherche sur cette interface est basée sur du texte intégral, des mots clés, des types de ressources, organisations et échelle [74]. Les utilisateurs ont la possibilité de redéfinir les critères de recherche pour obtenir rapidement les données souhaitées. L'accès aux données spatiales permet d'alimenter les modèles cellulaires. Ce qui nous renseigne sur les positions géographiques des zones d'étude. Actuellement ceci est assuré par l'outil *Mercator* qui nous fournit directement des données géospatiales dans les cellules.

OneGeology : une source de données géologiques C'est un projet phare de l'année internationale de la planète Terre 2008. Son objectif est de rassembler à l'échelle mondiale les cartes géologiques numériques fournies par chaque pays. Le soutien des pays participants a permis de faire progresser rapidement cette initiative qui a recueillie l'adhésion des services géologiques et des utilisateurs de données.

L'initiative repose sur des échanges scientifiques entre géologues du monde entier. Ses objectifs sont de :

- rendre accessibles les données cartographiques géologiques disponibles dans chaque pays quel que soit le format numérique ;

- contribuer à la définition de normes cohérentes d'accès aux données et garantir ainsi l'interopérabilité de ces données ;
- assurer un transfert de savoir-faire à ceux qui en ont besoin, reconnaissant ainsi par cette approche que certaines nations participantes disposent de capacités différentes ;
- encourager la mise en oeuvre rapide de l'interopérabilité en développant et en utilisant le langage de description, GeoScience Markup Language (GeoSciML), un schéma XML pour l'échange de données.
- améliorer et accroître l'utilisation et la convivialité de leurs données.

3.2.3 EIS-AFRICA, une initiative continentale

C'est une organisation panafricaine qui fournit des sources d'informations géospatiales et environnementales à l'échelle nationale et internationale sur le continent africain. Elle suscite un débat autour du partage des architectures de données et du cadre juridique de l'exploitation des données, afin de faciliter la prise de décisions.

Des unités de formation fonctionnelles ont été mises en place dans des pays comme le Madagascar, la Tanzanie et le Ghana pour répondre aux besoins de vulgarisation de personnel qualifié. Par la suite, Madagascar et Ghana ont créé une série complète d'ensembles de données environnementales.

3.2.4 Les initiatives nationales publiques

La plupart des états proposent des séries de données sur des plateformes web du type <http://data.gouv.cc>. Aux Etats Unis, par exemple, la plateforme (<https://www.data.gov/>) est proposée par le gouvernement fédéral. Elle fournit des données relatives à plusieurs thématiques parmi lesquelles on trouve, le climat, les océans, l'agriculture, etc. . .

La France aussi dispose d'un portail web similaire (<http://www.data.gouv.fr>) depuis février 2011. Ce portail propose à peu près les mêmes thématiques que celui des EU.

Les pays comme le Royaume-Uni (<https://data.gov.uk/>), la Nouvelle-Zélande (<https://data.govt.nz/>) ne sont pas en reste, ils proposent aussi des sources de données.

En Afrique, des pays comme l'Afrique du Sud à travers la South African Open data, le Burkina Faso (<http://www.data.gov.bf>), la Côte d'Ivoire (<http://www.data.gov.ci>) et le Maroc (<http://www.data.gov.ma>) proposent aujourd'hui des sources d'informations Open Data. Ces plateformes proposent un ensemble de données relatives à l'administration, au climat, l'environnement, etc. . .

Le Sénégal propose une plateforme de base de données géospatiales accessible via le lien (<http://www.basegeo.gouv.sn/>). Cette plateforme propose des données géospatiales relatives aux reliefs, à l'hydrographie, la végétation et le climat. Ces données se présentent sous forme d'images au format *tiff* et de données vectorielles normalisées aux formats *shapefile* et *kml*. Les données proposées sur cette plate-forme sont produites dans le cadre d'une mission de service public. L'inscription et l'utilisation de la plate-forme sont gratuites sous réserve d'enregistrer son identité de personne physique ou morale

[69]. Ces données peuvent servir à définir un état hydrographique ou climatique dans les cellules lors des simulations physiques.

La plateforme (<http://http://senegal.opendataforafrica.org>) propose des données sur le Sénégal. Cette plateforme est une initiative de Agence Nationale De Statistique Et De La Démographie (ANSD).

UMS3468 Bases de données biodiversité, écologie, environnements et sociétés (BBEES) est un soutien scientifique et technique auprès des chercheurs du muséum national d'histoire naturelle (MNHN) et l'institut écologie et environnement (InEE) du CNRS [1]. Leur mission est de structurer et optimiser le travail autour des bases de données de recherche sur la biodiversité, l'écologie, l'environnement et sociétés.

Il favorise la diffusion des bases de données au travers d'un portail dédié (<http://www.bdd-inee.cnrs.fr/>) accessible aux chercheurs, aux pouvoirs publics et aux citoyens.

L'UMS BBEES propose quelques recommandations afin de faciliter l'intégration de bases de données dans des dispositifs nationaux et internationaux :

- la constitution des corpus et le traitement des données ;
- le choix des outils ;
- la structuration des données ;
- les métadonnées.

Ces recommandations sont en phase avec les standards et les normes en vigueur, comme par exemple la directive européenne Inspire pour les informations géographiques.

3.2.5 Les initiatives nationales privées

Alliance nationale de recherche pour l'environnement (AllEnvi) Sur le plan national, la France a mis sur pied AllEnvi pour la gestion des données environnementales. Un séminaire sur les données environnementales, organisé en collaboration avec le Muséum national d'histoire naturelle, tenu le 21 mai 2014, a permis de regrouper plus d'une centaine de participants. Ce séminaire a porté sur les enjeux de la gratuité des données et la difficulté à trouver un modèle économique satisfaisant, compte tenu du coût important d'acquisition et de gestion des données en accès libre. Les données sont stockées sur des ordinateurs qui coutent relativement chers et leur maintenance a aussi un coût.

ECOSCOPE Le portail ECOSCOPE [8] est un point de diffusion des métadonnées et d'accès à des jeux de données. C'est un outil ouvert qui favorise la circulation des métadonnées entre des systèmes d'information.

Ce portail a pour objectif de mettre à disposition des données d'observation pour la recherche sur la biodiversité afin de fournir des documents et de comprendre l'état et la dynamique et des mécanismes sous-jacents. Ceci permet de fairer des prédictions en construisant des scénarios du futur de la biodiversité.

ECOSCOPE est une infrastructure qui doit contribuer à mieux organiser la collecte, la gestion et la valorisation des données. Ces données proviennent d'observatoires de recherche sur la biodiversité, de centres de ressources biologiques et de collections, mais également des données d'expérimentation et restent gérées par les laboratoires.

Ces données sont plus destinées à alimenter la recherche et fournir des informations et synthèses pour l'appui à la décision politique.

3.3 La correspondance objet-relationnel

"Smalltalk"[182] est un langage de programmation orienté objet. Il a servi au développement des outils de modélisation et de simulation utilisés dans ces travaux. Ces outils utilisent un ensemble d'objets qu'on peut stocker dans une base de données relationnelle. Cependant, le stockage direct de ces objets est impossible car le modèle objet décrit un système par le biais d'éléments dotés d'une entité, d'un comportement et d'un état alors que le modèle relationnel décrit un système par le biais de ses données.

Par ailleurs, les codes du langage de programmation orienté objet sont différents de ceux utilisés pour identifier les tables et colonnes dans une base de données relationnelle. Par conséquent, une conversion s'avère nécessaire dans la génération de données de modèle objet dans un modèle relationnel ou vice-versa.

Le schéma suivant (voir Figure 3.3) montre les interactions entre les classes d'un modèle objet et les tables d'un modèle relationnel.

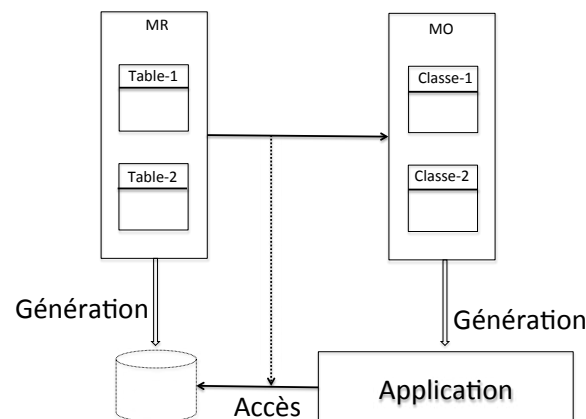


FIGURE 3.3: Interactions objet-relationnel

La correspondance objet-relationnel permet d'assurer une persistance de données entre modèles orienté objet et relationnel. Autrement dit, elle permet de convertir des objets "Smalltalk" dans le but de les stocker dans une base de donnée relationnelle. La notion de persistance des données est relative à la sauvegarde et à la restauration des données en évitant les pertes de données. Cette conversion est possible grâce à deux modules de "Smalltalk" : "Active Record" et "Glorp" (Generalized Lightweight Object-Relational Persistence).

Les modèles objet et relationnel ne respectent pas les mêmes règles de cohérence interne, "Smalltalk" utilise un ensemble de collection de classes bien fourni tel que les

dictionnaires, les collections ordonnées alors que les bases de données relationnelles utilisent des tables. Par conséquent les correspondances entre ces deux modèles ne sont pas évidentes. Par exemple, pendant que les types "SQL" (String, Number, etc. . .) peuvent être "mappés" sur des classes "Smalltalk" correspondant, le "mapping" entre des types de dimensions différentes est délicat.

Aussi, les bases de données relationnelles utilisent une approche déclarative, c'est à dire qu'on spécifie les propriétés des données recherchées, alors que les modèles orientés objet utilisent une approche comportementale, c'est à dire que les données sont décrites par des fonctions. Les objets ne suivent pas un schéma explicite, les variables d'instance d'une classe ne sont pas typées et peuvent être polymorphes, par contre les tables d'une base de données relationnelle sont généralement définies à l'aide de schémas qui est modèle de données, c'est à dire la description de l'organisation des données. De plus, les classes héritent des comportements de leur superclasse, mais il n'existe pas d'abstraction correspondant à l'héritage pour les modèles relationnels. C'est à dire que les classes considèrent les expressions comme un comportement hiérarchique et les tables, comme des taxonomies.

3.3.1 La bibliothèque de correspondance : "Glorp"

Il permet la conversion, en vue de stockage, de données de modèles orientés objet dans des bases de données relationnelles. À cet effet, "Glorp" s'interface entre la sémantique des modèles relationnels et ceux orientés objet afin de fournir des transactions de niveau objet.

La bibliothèque supporte plusieurs bases de données : *Oracle*, *Oracle ODBC*, *PostgreSQL*, et *MySQL*.

Sa particularité réside dans le fait qu'elle n'a pas besoin d'intégrer des instructions "SQL" dans un code "smalltalk", celles-ci sont générées automatiquement avec les métadonnées.

Elle présente les fonctionnalités suivantes :

- un support évolutif pour les modèles simples et complexes ;
- une correspondance déclarative avec les métadonnées ;
- des requêtes dynamiques et complexes ;
- une gestion transparente et non-intrusive des relations entre les objets ;
- la lecture et l'écriture partielles des objets ;
- les transactions niveau objet.

Avec "Smalltalk" les données ne sont pas typées, elles sont représentées par des classes. Donc les langages de définition de données ou "Data Definition Language" (DDL) comme on en trouve avec SQL ne sont pas utilisés. La correspondance objet-relationnel requiert non seulement une conversion entre les objets et tables, mais aussi une table de correspondance. Pour cela, des descriptions déclaratives particulières, appelées métadonnées sont utilisées.

Structure des métadonnées La structure des métadonnées est l'un des aspects fondamentaux à comprendre dans le processus de "mapping" objet-relationnel. Ces métadonnées sont utilisées dans les transactions de bases de données telles que la lecture, l'écriture, la jointure, etc. Ces dernières sont pour la plupart, transparentes et gérées automatiquement. Ainsi, il n'est pas nécessaire d'écrire du code SQL, car les métadonnées gèrent automatiquement la correspondance entre les objets "Smalltalk" et leur représentation dans la base de données.

Avec l'API "Active Record", une autre bibliothèque de correspondance, l'écriture des métadonnées est plus facile, car pour bon nombre de modèles, quelques étapes simples suffisent à configurer un schéma de base de données. Les métadonnées sont écrites en "Smalltalk", et constituent une partie de l'application.

La représentation des tables en classe La plupart des outils de persistance orientés objet représentent les variables d'instance des classes en utilisant des colonnes de la table. Chaque instance de la classe correspond à une ligne.

"Glorp" suit ce canevas général en proposant des schémas de correspondance qui définissent comment les classes sont mappées dans la base de données. Plusieurs exemples de schémas existent, chacun définit de manière spécifique le mode de conversion entre un objet "Smalltalk" et une colonne d'une table.

L'écriture en métadonnées des correspondances objet-relationnel permet de se passer de celle des correspondances entre les tables, lignes, ou colonne et les objets.

De même "Glorp" ne nécessite pas de code pour chaque classe en termes de lecture et/ou écriture des objets. Il utilise un modèle unique pour chaque application, dont la méta-représentation est exprimée en utilisant la classe "DescriptorSystem". Les variables d'instance d'une classe sont représentées avec des objets distincts, tous réunis dans la classe "DescriptorSystem".

La définition des métadonnée et correspondance d'un modèle consiste à créer cette classe.

Après la création du modèle avec une annotation des métadonnées dans la classe "DescriptorSystem" (voir Figure 3.4), des objets peuvent être stockés dans une base de données. Pour cela, la classe "DatabaseAccessor" sera utilisée pour établir une connexion et ouvrir une session pour coordonner les requêtes de lecture et d'écriture sur la base de données.

3.3.2 Notions de transactions objet et d'unités de travail

Les bases de données sont généralement transactionnelles afin d'assurer l'intégrité des données, mais comme "Glorp" fonctionne avec des objets, les contraintes de transactions sont légèrement différentes. À la place des transactions, une partie de l'API de session est utilisée pour exécuter des unités de travail. Ces dernières sont des transactions de niveau objet, sauf que la manipulation d'objets complexes pouvant s'étendre sur plusieurs tables, est automatisée. Un cache contenant les identités des objets permet de revenir sur les

changements d'objets "Smalltalk" en cas d'annulation de la transaction. Tous les objets utilisés au cours d'une unité de travail sont stockés dans ce cache.

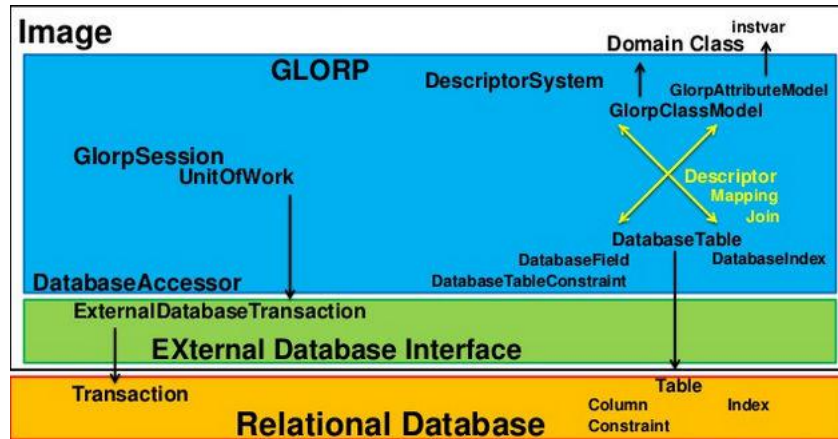


FIGURE 3.4: Architecture de "Glorp" et interactions avec une base de données

"Glorp", à travers des bibliothèques clientes, fournit une interface d'interaction avec un serveur de base de données.

3.3.3 Connexion à la base de données

Un serveur de bases de données peut être exécuté localement en mode "standalone" ou sur une autre machine. Dans les deux cas, il est chargé de la gestion des fichiers de stockage et effectue des opérations pour le compte de chaque client.

Pour utiliser la base de données, un client ouvre une connexion avec le serveur, et envoie des commandes à travers une session. Au sein de chaque session, des unités de travail sont exécutées.

Une connexion au serveur est créée avec deux objets : le "DatabaseAccessor" et le "Login". Le premier fournit un mécanisme indépendant pour se connecter et se déconnecter du serveur et le second contient les paramètres spécifiques à la connexion tels que le nom d'utilisateur, le mot de passe, et l'environnement.

Création d'un objet connexion La connexion est gérée par la classe "DatabaseAccessor" qui requiert un objet "Login" comme paramètre. Cet objet, créé une fois pendant la configuration, est sauvegardé dans une variable partagée et utilisé lors de l'établissement de la connexion.

Le code qui suit montre comment créer un objet "Login" pour une base Postgres sur la machine opale, avec "toto" comme nom d'utilisateur et "passer" comme mot de passe :

```
login := Login new database: PostgreSQLPlatform new; username: 'toto';
password: 'passer';
connectString: 'opale' , '_' , 'db'.
```

L'objet "Login" contient toutes les informations relatives aux connexion et déconnexion à une base de données. La variable d'instance "secure" renseignée à 'true' permet à cet objet de garder les paramètres après une connexion réussie.

```
aLogin secure: true.
```

Création de l'interface "DatabaseAccessor" L'objet "DatabaseAccessor" fournit une interface abstraite, indépendante des bases de données. Ses instances sont utilisées comme des objets de connexion. Une fois l'objet "Login" créé, une instance de la sous-classe de "DatabaseAccessor" est définie comme suit :

```
accessor := DatabaseAccessor forLogin: login.
```

Pour envoyer les commandes par le biais d'un "accessor" de base de données, un objet "session" sera utilisé. Généralement, chaque objet "accessor" appartient à un seul objet "session".

Connexion à la base de données La connexion à une base de données se fait en utilisant un "accessor" spécifique :

```
accessor login.
```

La déconnexion utilise le bout de code suivant :

```
accessor logout.
```

Test d'une connexion Après une connexion, il est utile de procéder à des tests pour voir si la connexion est bien établie, le bout de code qui suit permet de faire cela :

```
[accessor login; logout]
on: Error
do: [:ex | Transcript show: ex messageText].
```

Si "l'accessor" ne peut pas accéder à la base de données, à cause de paramètres incorrects passés à l'objet connexion, par exemple, une exception est levée.

Exécution de requêtes SQL L'objet "DatabaseAccessor" permet aussi d'exécuter directement des requêtes SQL sur une base de données. Ce n'est pas recommandé, mais toutefois, dans certaines situations, il peut être utile.

Pour exécuter un String contenant une expression SQL :

```
accessor beginTransaction.
result := accessor executeSQLString: aSQLString. accessor commitTransaction.
```

L'objet "result" est un tableau de valeurs représentant les lignes retournées, si elles existent.

Si le String passé à "#executeSQLString :" contient du SQL invalide, une exception est levée. Le bout de code qui suit permet de gérer les exceptions.

```
[result := accessor executeSQLString: aSQLString] on: Error
do: errorBlock
```

La connexion au serveur de bases de données nécessite une session qui est chargée de coordonner les interactions entre les applications et la base de données.

Création des objets "session" et "Login" Un objet "session" s'occupe de la gestion des requêtes, du cache d'objets, des descripteurs et des unités de travail.

À la fin d'une unité de travail, l'objet "session" décide de l'insertion, de la mise à jour ou de la suppression de données dans la base. Il est également chargé de générer des expressions SQL appropriées pour compléter les unités de travail.

L'objet "Login" contient un objet "DatabaseAccessor" et une instance de la classe "DescriptorSystem" de l'application. Ces paramètres doivent être définis lors de la création de cet objet.

```
session := MyDescriptorSystem sessionForLogin: aLogin
```

Un objet "session" est obtenu en utilisant les objets "sessionForLogin :" et "Login".

Alternativement, l'objet "session" peut être créé explicitement avec les objets "Login" et "DatabaseAccessor", et la classe "DescriptorSystem".

```
session := GlorpSession new.
session system: (MyDescriptorSystem forPlatform: myLogin database).
session accessor: accessor.
```

En définissant le "DescriptorSystem" avec l'objet "system :", la session fait une copie du temps d'exécution pour une utilisation pendant la session. Des modifications du comportement de la classe "DescriptorSystem" n'interfèrent pas les sessions actives. Pour cela, un nouvel objet session doit être créé.

"Glorp" peut être utilisé pour modéliser un ensemble de tables d'une base de données relationnelle ou en créer de nouvelles. Dans les deux cas, la classe "DescriptorSystem" est utilisée.

3.3.4 Modélisation d'une base de données avec "Glorp"

La classe "DescriptorSystem" contient des métadonnées pour faire correspondre les objets du modèle aux tables de la base de données. Typiquement, le modèle de l'application requiert une seule classe "DescriptorSystem". Mais, plusieurs classes peuvent exister dans le modèle, toutes modélisées en utilisant un "DescriptorSystem" unique. C'est à dire, toutes les métadonnées sont encapsulées dans cette classe.

La classe "DescriptorSystem" fournit un comportement qui spécifie les noms de toutes les tables dans le modèle, les classes auxquelles ils correspondent, la structure de chaque table, et les variables d'instance à laquelle ils sont mappés. Ce comportement se traduit par l'ajout de méthodes d'instance qui spécifient toutes les correspondances.

La modélisation d'une base de données nécessite les tâches suivantes :

1. Créer une classe "DescriptorSystem" ;
2. Ajouter des méthodes pour modéliser les tables de la base de données ;
3. Ajouter des méthodes pour construire des descripteurs ;
4. Ajouter des méthodes pour construire des modèles de classe.

Il faut noter qu'avec "Active Record", les deux premiers points suffisent pour faire persister un modèle "Smalltalk" dans une base de données.

Méthodes de création de classes Les objets convertis par "Glorp" en tables, sont représentés en utilisant des classes. Par exemple, les instances d'une classe nommée MaClasse peuvent être converties en lignes dans une table nommée MaTable. Généralement, les colonnes de la base de données correspondent à des variables d'instance dans une classe. Si la table comporte une colonne utilisée comme clé primaire, cette dernière doit être considérée également comme une variable d'instance.

Convention de nommage de "Glorp" "Glorp" respecte les conventions de nommage des systèmes de gestion de bases de données relationnelles, mais en propose également d'autres. Par exemple, les noms des tables et colonnes sont écrits en minuscules, avec des mots séparés par des "underscores" alors qu'en "Smalltalk", la plupart des noms utilisent la casse de chameaux "camel case" et les "underscores" ne sont pas recommandés. La casse de chameau est une pratique qui consiste à écrire un ensemble de mots en mettant en majuscule la première lettre des mots liés.

"Glorp" supporte ces deux conventions, mais avec deux exigences, d'abord que les noms des table et colonne commencent par une lettre. Les noms peuvent utiliser toute combinaison de caractères alphanumériques et de soulignement à la suite de cette lettre, mais, ils ne peuvent pas commencer par un "underscore".

Aussi il exige que les noms utilisent le même nom que la table à laquelle ils sont appliqués, annexé avec un suffixe particulier. Par exemple, pour une table appelée "Table" une clé primaire appliquée à une colonne serait appelée "Table_PK".

Création d'un "DescriptorSystem" Avec "Glorp", une application est modélisée en utilisant une sous-classe de "DescriptorSystem". Ce dernier contient tous les descripteurs du modèle (le schéma) en utilisant un certain nombre de méthodes d'instance secondaires.

Le comportement de "DescriptorSystem" définit les classes du modèle et précise les tables de la base de données auxquelles elles doivent être "mappées". Les classes du modèle et les tables correspondant dans la base de données sont représentées par des méthodes distinctes de la classe "DescriptorSystem". Cette classe unique, contient

toutes les spécifications de métadonnées et de correspondance nécessaires pour utiliser l'application avec "Glorp".

La création d'une classe "DescriptorSystem" se fait comme suit :

```
MyAppNameSpace defineClass: #MyDescriptorSystem superclass: #{Glorp.DescriptorSystem}
indexedType: #none
private: false
instanceVariableNames: '' classInstanceVariableNames: '' imports: 'Glorp.*'
category: 'MyApplication'
```

On remarque que la classe "DescriptorSystem" doit être définie dans le même espace de nom que le modèle (e.g, "MyAppNameSpace.*") et il devrait importer l'espace de nom de "Glorp.*".

Pendant l'exécution, "Glorp" crée une instance de cette classe pour chaque session de base de données et l'utilise pour gérer les correspondances objet-relationnel.

Pour exprimer la correspondance entre les tables de la base de données et les objets "Smalltalk", un certain nombre de méthodes seront ajoutées à la classe "DescriptorSystem". Ces méthodes permettent de :

1. Modéliser les tables des bases de données ;
2. Construire des descripteurs pour toutes les classes du modèle ;
3. Construire des modèles de classes.

La première étape suffit pour faire persister le modèle avec Active Record. Ce dernier génère dynamiquement les descripteurs et classes du modèle.

Toutefois, il est possible d'ajouter des méthodes à la classe descripteur, qui écrasera la correspondance dynamique.

De plus, avec Active Record, si les tables existaient déjà dans la base de données, aucune de ces méthodes ne serait nécessaire tant que les classes et les tables de la base de données respectent la convention de nommage de Active Record.

Les méthodes de la classe descripteur système sont présentées dans le tableau ci dessous (voir tableau 3.1) :

3.4 Intégration PostgreSQL et application Visualworks

VisualWorks [115] propose plusieurs parcelles permettant d'intégrer des bases de données dans les applications "Smalltalk". Cette intégration est basée sur une API pour un accès de bas niveau, connue sous le nom d'interface de base de données externe "External Database Interface" (EXDI).

Il existe un support EXDI pour *Oracle*, *PostgreSQL*, *Sybase*, *ODBC*, *MySQL*, *SQLite*, et les bases de données *DB2*. En plus de la composante PostgreSQL3EXDI qui utilise la version 3.0 de Postgres, la parcelle PostgresSQLEXDI est utilisée par les versions antérieures à la version 2.0.

TABLE 3.1: Les méthodes de la classe "DescriptorSystem"

"classModelForName :"	<ul style="list-style-type: none"> • Attend une instance de "ClassModel" comme paramètre, qui reçoit des informations sur les variables d'instance de la classe NAME. • Pour chaque classe, il existe une méthode "classModelForName :" correspondant.
"tableForName :"	<ul style="list-style-type: none"> • Envoie des informations sur les colonnes dans la table NAME.
"descriptorForName :"	<ul style="list-style-type: none"> • Définit le "mapping" entre la variable d'instance de la classe "NAME" et une ou plusieurs tables des méthodes "classModelForName :" correspondant. • Attends une instance de Descriptor comme paramètre

Pour de nombreuses applications, l'API EXDI est suffisante pour interagir avec une base de données. Les applications nécessitant un mapping relationnel objet plus évolué peuvent utiliser les modules "Glorp", "ActiveRecord" ou "lens". Le dernier est compatible uniquement avec *Oracle*, *Sybase* et *DB2*.

3.4.1 Le support de bases de données : "EXDI"

Le paquetage "EXDI" fournit un ensemble de supports à plusieurs superclasses, mais pas directement à une base de données particulière. Les bases de données supportées, viennent avec des parcelles "EXDI" (*OracleEXDI*, *DB2EXDI*, *PostgreSQL3EXDI*). Le chargement de l'une d'elles charge aussi la parcelle qui fournit le module "EXDI".

Pour une compatibilité, Une plateforme 64 bits nécessite une image VisualWorks 64 bits et une bibliothèque client appropriée de 64 bits. De même, une image 32 bits tournant sur une plateforme 32 bits requiert une bibliothèque client 32 bits.

3.4.2 Chargement du support de bases de données

Les modules "Glorp", "ActiveRecord" et Lens fournissent le support pour la correspondance entre schémas de bases de données et "Visualworks". Cette correspondance est appelée correspondance objet-relationnel.

Le module Lens, obsolète utilise par exemple les trois parcelles présentées ci dessous :

- *Lens-Runtime.pcl*, pour l'exécution ;
- *Lens-Dev.pcl*, pour un développement complet ;
- *LDM-Framework.pcl*, utilisé par le compilateur.

"Visualworks" dispose d'un gestionnaire de parcelles permettant de charger les parcelles dans une image. Pour cela, sélectionner système dans la fenêtre de lancement

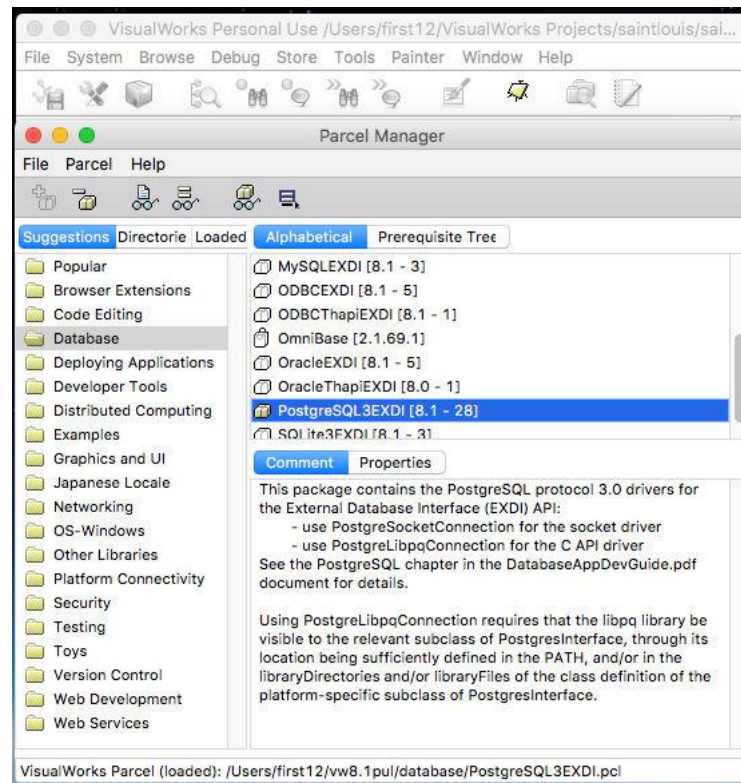


FIGURE 3.5: Chargement de la parcelle postgresSQL3EXDI

principal, puis gestionnaire de parcelles, ensuite sélectionner les extensions de base de données proposées, et charger les parcelles en double-cliquant sur les éléments souhaités dans la partie supérieure droite de la liste de gestionnaire de parcelles (voir Figure 3.5).

3.4.3 Connexion à la base de données

La connexion à la base de données nécessite de renseigner certaines informations. Le nom de la base de données, une chaîne de caractère qui l'identifie, **<host_name>_<dbSID>** (exemple **opale_pickcelldb0**) identifie une base de données nommée pickcelldb0 sur le système nommé opale.

Toutes les bibliothèques de base de données sont dynamiquement chargées dans l'image via des bibliothèques partagées. Pour accéder à ces dernières, il est essentiel que la variable d'environnement UNIX **LD_LIBRARY_PATH** soit renseignée.

3.5 Conclusion

Ce chapitre a permis de présenter les sources de données environnementales disponibles. Nous avons procédé à une classification des promoteurs de bases de données selon qu'ils sont nationaux, internationaux, privés ou publics. Cette classification a

servi de prétexte pour inventorier les formats de données mis à disposition et comment accéder aux de sources de données environnementales. Du fait de la variété et de la diversité des formats de données, les initiatives normatives pour leur harmonisation ont été présentées.

Nous avons aussi présenté les mécanismes de correspondance objet-relationnel. Ces mécanismes permettent d'assurer la persistance des données dans les bases. Cette correspondance est nécessaire car les modèles orientés objet et relationnel ne respectent pas les mêmes règles de cohérence de données. Pour cela nous avons utilisé deux bibliothèques qui permettent de mettre en oeuvre cette persistance, il s'agit de "Active Record" et "Glorp"

Enfin, un exemple d'intégration de bases de données Postgres avec une application "Smalltalk" a été présenté en prélude à un stockage des données météorologiques dans une base de données. L'objectif est de rendre disponible en local un ensemble de données utilisables dans les modèles cellulaires.

4

Programmation d'automates cellulaires représentant la dynamique du criquet pèlerin

4.1 Introduction

Il faut maintenant se préoccuper de considérations plus précises sur les programmes qui vont permettre d'animer les quadrillages de terrain et leurs habitants.

Une note préliminaire concerne les données utilisées dans les simulations. Celles-ci proviennent de sources géographiques ou de photographies satellites. Il est souvent nécessaires de compléter ces données par des sources annexes. Les élévations proviennent des campagnes de mesures SRTM de la NASA, via des serveurs publics que l'on interroge pour chaque grille. Les paramètres climatiques sont distribués dans des fichiers standardisés au format GRIB par les organismes météorologiques tels que la NOAA (National Oceanic & Atmospheric Administration américaine), ou Météo-France. Il s'agit de fichiers encodés contenant des données météo numériques et des prévisions sur une zone particulière.

Dans la suite de ce chapitre, la section 4.2 sera consacrée à la représentation cellulaire de populations de criquets pèlerins, la section 4.3 décrira le canevas de découpage de la zone de reproduction cible située en Mauritanie et de génération de systèmes cellulaires, la section 4.4 portera sur la présentation des modèles cellulaires paramétriques, la section 4.5 présentera les résultats de simulation de deux scénarios, le premier relatif à l'évolution locale du criquet dans une cellule et le second, aux migrations entre les cellules et la section 4.6 conclura ce chapitre.

4.2 Représentation cellulaire des populations de criquets

La représentation cellulaire n'est pas une nouveauté. Elle est utilisée en météorologie pour étudier et simuler les évolutions du climat et dans beaucoup d'autres problématiques physiques. Dans ce domaine météorologique, les grilles 3D permettent la représentation de l'atmosphère avec des résolutions spatiale et temporelle de plus en plus fines, afin d'améliorer la qualité des prévisions. Dans les grilles cellulaires, les équations des modèles physiques permettent de prédire l'évolution de l'état de l'atmosphère.

Les systèmes cellulaires utilisés pour les criquets ne se comportent pas autrement. Plus la résolution est fine, plus les cellules sont nombreuses et ceci influe sur la précision des résultats, mais comme le nombre de cellules augmente, le temps de simulation est impacté (voir Tableau 4.1).

Fragmentation du cycle de vie dans une population cellulaire

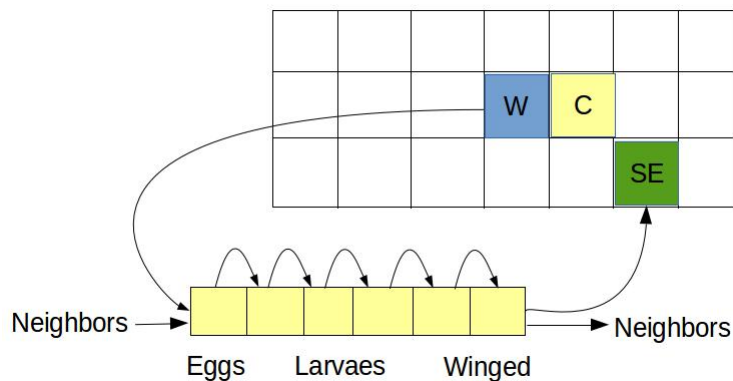


FIGURE 4.1: Illustrations de l'évolution locale dans une cellule et de migration entre cellule. La voisine de l'Ouest envoie une population entrant à la cellule C. Après les différentes transitions entre les micro-états, la voisine Sud-Est reçoit une population sortant. Cette figure résume nos modèles et montre comment les interactions locales entre les micro-états et entre les cellules se passent. Toutes ces interactions se font de manière concurrente : le cycle de vie et ses habitants est un pipeline.

Nos systèmes physiques sont découpés en grilles de dimension deux dont les structures de données sont décrites comme suit : chaque cellule contient neuf tableaux correspondant au cycle de vie de l'insecte, pour les œufs, les cinq stades larvaires, les ailés, les individus solitaires et grégaires. Chacun de ces tableaux est à son tour subdivisé en micro-états dont la durée totale correspond à celle de la phase de vie considérée. Par exemple, `MaxEggsLifePeriod` représente le nombre fini de micro-états relatifs aux œufs, correspondant à la durée de vie des œufs dans le cycle biologique. Les micro-états permettent de représenter finement les aléas d'évolution à l'intérieur d'une phase.

On a vu que le programme associé à un système cellulaire peut être formulé en terme de processus communicants. L'organisation spatiale des différents processus voisins des

cellules, est présentée sur la figure (voir Figure 4.2). Elle présente le système cellulaire autour d'une cellule cible. Il s'agit maintenant de processus mais l'interconnexion est conforme aux voisinages, Moore, dans notre cas.

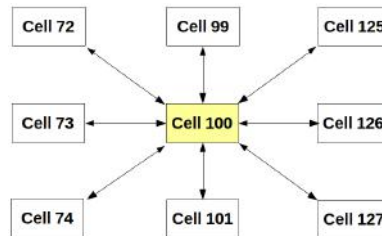


FIGURE 4.2: Topologie utilisée dans les modèles cellulaires. Chaque cellule contient un micro-système similaire à celui de la figure 4.1. Les communications figurent les dépendances physiques, ici, les migrations locales.

4.3 Génération du système cellulaire

La zone de reproduction du criquet pèlerin située au Trarza, au Sud-Ouest de la Mauritanie a été traitée en utilisant différents niveaux de granularité afin de produire des systèmes cellulaires. C'est une zone sous surveillance qui fait l'objet de prospections régulières pour s'enquérir de la situation acridienne par les équipes du centre national de lutte antiacridienne (CNLA) de la Mauritanie en collaboration avec le service d'observatoire acridien de la FAO. C'est donc un support source d'informations qui peuvent être précieuses lors des validations.

La figure 4.3 présente l'allure de la liste des cellules obtenues après une opération de segmentation. Les processus correspondants sont désignés par des coordonnées géométriques (P_{xy}) sur la colonne 5. La géo-localisation calculée par PickCell est complétée par l'élévation correspondante extraite d'un serveur SRTM montrée sur la colonne 4.

Le même principe est utilisé pour extraire d'autres types d'informations tels que les vecteurs U, V pour les données relatives au vent, les températures minimales, maximales et moyennes pour les données de température, et des données de précipitation par exemple.

La transposition entre coordonnées géométriques x, y sur la carte, et coordonnées géographiques est effectuée par projection de Mercator. L'algorithme utilisé est implanté dans une classe fondée sur les systèmes de tuilages standardisés utilisés sur la cartographie Internet [88]. C'est une projection cylindrique, conservant les angles tangents à l'équateur du globe terrestre sur une carte plane formalisée par le géographe flamand, Gerardus Mercator en 1569. Ceci permet de reporter directement sur la carte les angles mesurés au compas, et vice-versa. Elle s'est imposée comme le planisphère standard dans le monde grâce à sa précision pour les voyages marins [237]. Le brosseur de tuiles QuickMap [156] permet d'adapter l'échelle de la carte par navigation en surface et en profondeur.

type	latitude	longitude	altitude (m)	name
W	18.483516449	-16.051025391	2.5	Px1y1
W	18.483516449	-16.044158936	3.5	Px2y1
W	18.483516449	-16.037292480	3.7	Px3y1
W	18.483516449	-16.030426025	3.6	Px4y1
W	18.483516449	-16.023559570	3.2	Px5y1
W	18.483516449	-16.016693115	3.1	Px6y1
W	18.483516449	-16.009826660	1.4	Px7y1
W	18.483516449	-16.002960205	3.2	Px8y1
W	18.483516449	-15.996093750	2.5	Px9y1
W	18.483516449	-15.989227295	-0.4	Px10y1
W	18.483516449	-15.982360840	-1.4	Px11y1
W	18.483516449	-15.975494385	1.9	Px12y1
W	18.483516449	-15.968627930	0.0	Px13y1
W	18.483516449	-15.961761475	-1.7	Px14y1
W	18.483516449	-15.954895020	-2.7	Px15y1
W	18.483516449	-15.948028564	-1.0	Px16y1
W	18.483516449	-15.941162109	-2.0	Px17y1
W	18.483516449	-15.934295654	-0.8	Px18y1
W	18.483516449	-15.927429199	-2.8	Px19y1
W	18.483516449	-15.920562744	-1.9	Px20y1
W	18.483516449	-15.913696289	-1.8	Px21y1
W	18.483516449	-15.906829834	-2.1	Px22y1
W	18.483516449	-15.899963379	-0.8	Px23y1
W	18.483516449	-15.893096924	-1.9	Px24y1
W	18.483516449	-15.886230469	-0.3	Px25y1
W	18.483516449	-15.879364014	-2.5	Px26y1
W	18.483516449	-15.872497559	-2.8	Px27y1
W	18.483516449	-15.865631104	-1.9	Px28y1
W	18.483516449	-15.858764648	-3.5	Px29y1
W	18.483516449	-15.851898193	-1.2	Px30y1
W	18.483516449	-15.845031738	-2.5	Px31y1
W	18.483516449	-15.838165283	0.6	Px32y1
W	18.483516449	-15.831298828	-1.0	Px33y1
W	18.483516449	-15.824432373	-1.2	Px34y1
W	18.483516449	-15.817565918	-2.1	Px35y1
W	18.483516449	-15.810699463	-2.0	Px36y1
W	18.483516449	-15.803833008	-0.0	Px37y1
W	18.483516449	-15.796966553	-1.7	Px38y1
W	18.483516449	-15.790100098	0.2	Px39y1
W	18.483516449	-15.783233643	-1.1	Px40y1
W	18.483516449	-15.776367188	-1.6	Px41y1

FIGURE 4.3: Edition de coordonnées géographiques d'un système cellulaire

Le tableau 4.1 montre quelques niveaux de granularité utilisés dans le découpage spatial. Plusieurs programmes de simulation ont été produits par le générateur de code *Occam*, dont le compilateur permet difficilement de dépasser un nombre de processus supérieur à 800. L'alternative GPU permet toutefois d'atteindre des grains de $2@2$.

La production de ces systèmes est très rapide et ils peuvent partager le même comportement cellulaire. En même temps que le système cellulaire maillé, le générateur produit une spécification des données afférentes à chaque cellule. La synthèse de simulateurs peut donc être très rapide.

Le code du grain 25×25 est proposé en annexe A. L'ensemble de données correspondant avec leur libellé abstrait généré en *Occam* est proposé en annexe B.

4.4 Les modèles cellulaires d'évolution locale et de migration

Deux modèles sont présentés dans ce travail, le premier est relatif à l'évolution locale des individus dans une cellule et le second à la représentation migratoire des individus matures issus de cette évolution entre cellules.

Grain	Cellules	Aire correspondante	Nombre de canaux	MinFanOut	MaxFanOut
10@10	5053	764 X 764	38504	2	8
15@15	2191	1164 X 1164	16074	2	8
20@20	1287	1529 X 1529	9868	3	8
25@25	837	1911 X 1911	6352	3	8
30@30	572	2293 X 2293	4292	3	8
35@35	418	2675 X 2675	3102	3	8
40@40	304	3057 X 3057	2226	3	8
45@45	255	3440 X 3440	1852	3	8
50@50	195	3822 X 3822	1396	3	8

TABLE 4.1: Les éléments de segmentation spatiale de la vue graphique. Les paramètres MinFanOut et MaxFanOut représentent respectivement le nombre minimum de liens sortant d'une cellule et le nombre maximum de liens sortant d'une cellule. MaxFanOut affichera toujours la valeur 8 à cause de la topologie de voisinage utilisée, qui est Moore de rayon 1.

Le modèle d'évolution locale exprime le processus de développement d'une population en termes de transitions locales sur la base d'états et de micro états du cycle de vie du criquet pèlerin. Il permet de déterminer l'état et la densité des individus dans les différents stades de développement.

Le modèle de représentation migratoire, quant à lui, exprime les interactions qui existent entre cellules voisines selon des conditions éco-météorologiques bien définies. La migration des criquets pèlerins à la recherche de conditions éco-météorologiques favorables à leur développement est un cycle perpétuel. Ce modèle permet d'évaluer le nombre d'individus matures pouvant migrer vers d'autres cellules. Il permet aussi de prédire la direction de déplacement des individus dans les cellules en se basant sur les données météorologiques.

Rappelons que la simulation physique obéit au modèle synchrone, et qu'elle progresse par *tours* dont la vitesse d'évolution a été fixée à trois jours sur la base de constats effectués sur la vie des insectes.

4.4.1 Les variables d'état et la représentation physique paramétrique

Les variables d'état sont très importantes dans une modélisation cellulaire. Elles fournissent à chaque pas de temps t une description suffisante du système pour savoir comment celui-ci évoluera. La description du système est contingentée par l'ensemble des valeurs de l'ensemble des variables d'état.

Les fonctions de transition utilisées ici ont un caractère empirique, mais paramétrique. Elles devront être ajustées en fonction de constats établis en vraie grandeur dans un déploiement réel. Des scénarii tels que les attaques des criquets par les ennemis naturels que sont les parasites et les prédateurs ne sont pas considérés dans ces travaux. La raison est liée au fait que peu d'études quantitatives ont été effectuées dans ce sens [70] et les

soustraction des individus entrant et sortant permet de préparer l'étape suivante $t + 1$

Les processus suivants sont exécutés au niveau de chaque cellule, à chaque étape :

- déterminer la température locale et la comparer à celle des voisines ;
- déterminer le niveau de pluie et comparer avec celui des voisines ;
- déterminer la direction du vent ;
- évaluer la densité de population locale et la comparer à une valeur *seuil*, égale à la moitié des individus présents dans la cellule.

Une migration a lieu si la densité de population locale est plus grande que la moitié des individus y résidant, si le niveau de pluie est 2, si la température est 2. Sinon les criquets continuent leur développement dans la cellule si le niveau de pluie est 3 et la température est 3, autrement ils meurent.

4.4.3 Le modèle d'évolution locale

Le modèle d'évolution locale est relatif au développement d'une population de criquets pèlerins en termes de transitions locales sur la base d'états et de micro états de leur cycle de vie. On rappelle que la simulation obéit au modèle synchrone, qu'il progresse par *tours*, dont la vitesse d'évolution a été fixée à trois jours.

Nous décomposons ici le cycle de vie des criquets, comme cela pourrait être fait pour beaucoup d'autres espèces.

Une première transition a lieu lorsqu'un individu atteint la durée de vie maximale (elle est de douze jours pour les œufs, soit quatre tours synchrones). Par exemple, l'état œuf, défini par quatre micro états, évolue d'un micro état à un autre. Après quatre tours synchrones, tous les œufs deviennent des larves de stade L1.

Le premier stade larvaire dure 3 jours, c'est à dire un tour synchrone. Ainsi le cycle de vie des criquets de la ponte à l'état adulte est modélisé.

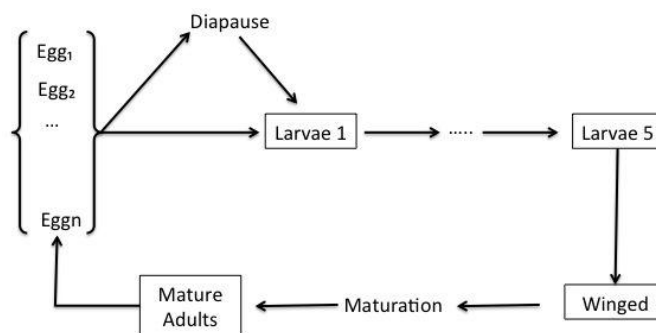


FIGURE 4.5: Trois principales étapes peuvent être distinguées dans le cycle de vie du criquet pèlerin : œuf, larve et adulte. La phase larvaire est composée de cinq étapes (L1 à L5). À la dernière étape larvaire, la mue imaginale se produit produisant des ailés immatures qui deviennent matures au bout de quelques semaines et dont les femelles peuvent pondre si l'humidité est suffisante dans une cellule.

Scénario1 : interaction locale dans une cellule

Dans le premier scénario, nous avons placé arbitrairement quelques œufs dans le premier micro état d'une cellule localisée à la position P_{x14y11} du système cellulaire, tous les autres micro états sont initialisés à 0 individus. Après quelques tours synchrones, certaines cellules voisines reçoivent quelques criquets adultes dont les femelles peuvent pondre des œufs. Le simulateur permet de retracer l'évolution des individus depuis l'état œuf, jusqu'à celui d'ailés immatures. Ce modèle peut être utilisé dans la prédiction de l'évolution de la population, dans le temps et l'espace. Il peut également être utilisé dans le comptage du nombre d'individus à chaque micro état.

```

Terminal
110 Eggs[-1] = 5000 + 0;
110 Larvae[-1] = 0 + 0 + 0 + 0 + 0;
110 Winged[-1] = 0 + 0 + 0;
110 Solitarious[-1] = 0 + 0 + 0 + 0 + 0 ;
110 Gregarious[-1] = 0 + 0 + 0 + 0 + 0;
110 Eggs[0] = 0 + 5000;
110 Larvae[0] = 0 + 0 + 0 + 0 + 0;
110 Winged[0] = 0 + 0 + 0;
110 Solitarious[0] = 0 + 0 + 0 + 0 + 0 ;
110 Gregarious[0] = 0 + 0 + 0 + 0 + 0;
110 Eggs[1] = 0 + 0;
110 Larvae[1] = 4673 + 0 + 0 + 0 + 0;
110 Winged[1] = 0 + 0 + 0;
110 Solitarious[1] = 0 + 0 + 0 + 0 + 0 ;
110 Gregarious[1] = 0 + 0 + 0 + 0 + 0;
110 Eggs[2] = 0 + 0;
110 Larvae[2] = 0 + 4673 + 0 + 0 + 0;
110 Winged[2] = 0 + 0 + 0;
110 Solitarious[2] = 0 + 0 + 0 + 0 + 0 ;
110 Gregarious[2] = 0 + 0 + 0 + 0 + 0;
:

```

FIGURE 4.6: Trace de simulation : evolution locale des individus

Les règles de transition qui régissent l'évolution entre les différents micro états est présenté dans le fragment de code transcrit en Occam ci-dessous. On peut remarquer qu'à chaque micro état, le programme calcule un certain nombre d'individus qui doivent quitter en fonction du taux de mortalité.

```

PROC Transfer(INT current , previous , delta )
  INT newPopInPrevious :
  SEQ
  IF
    delta >= previous
      delta := previous
  TRUE
  SKIP
  newPopInPrevious := previous - delta
  current := current + delta
  previous := newPopInPrevious
:

```

Cette procédure représente la fonction de transfert d'individus.

La variable `newPopInPrevious` est la population d'insectes dans l'état précédent. Elle est obtenue par soustraction d'un delta qui est fonction du taux de mortalité des individus à un stade donné.

La variable `current` est la population courante de criquets dans une cellule.

Le fragment de code suivant montre une application des règles de transition à la population œuf. Dans le modèle, il s'applique à tous les individus :

```
PROC EggEvolution(INT current , previous , delta)
  INT effective :
  REAL32 outgoing , outgoingRate :
  outgoingRate := 0,9660
  SEQ i=0 FOR MaxEggsLifePeriod
    SEQ
      outgoing := outgoingRate *
                  cP[eggsPeriod][i-1]
      eff := INT ROUND outgoing
      Transfer(cP[eggsPeriod][i] ,
              cP[eggsPeriod][i-1], eff)
  :
```

4.4.4 Le modèle de migration intercellulaire

Le modèle de migration considère les adultes immatures issus du modèle de développement local. Ces adultes peuvent migrer vers les cellules voisines si certaines conditions sont vérifiées.

L'automate local et général sont exposés sous forme de scénarii accompagnés de fragment de code en langage de programmation Occam.

Scénario2 : interaction entre les cellules

Dans ce scénario, l'objectif est de mettre en évidence les migrations inter cellulaires par la prédiction de la direction de migration des insectes et l'évaluation de la densité de population. Les résultats de simulation obtenus montrent des migrations d'individus. Les criquets migrants femelles peuvent pondre des œufs dans les cellules d'accueil.

La première génération de criquets pèlerins a été générée aléatoirement dans une cellule choisie arbitrairement, située à la position P_{x13y26} du système cellulaire. On peut observer l'évolution de la population de cellule à cellule. Ce simulateur permet déterminer la densité de population dans l'espace et le temps et la direction des migrations.

En se conformant au modèle synchrone, chaque système dispose de canaux d'entrée et de sortie pour la communication. Les cellules possèdent aussi des canaux d'entrée et de sortie à travers lesquels les migrations vont être modélisées. L'objectif est de déterminer à terme la taille de la population à chaque étape et de prévoir période de ponte et les phénomènes de changement de phase afin de représenter l'évolution des criquets.

```

Terminal
100 Eggs[-1] = 5070 + 4332;
100 Larvae[-1] = 3594 + 3387 + 3135 + 2816 + 2666;
100 Winged[-1] = 2516 + 2472 + 2238;
100 Solitarious[-1] = 2004 + 1957 + 1878 + 1781 + 1720 ;
100 Gregarious[-1] = 2238 + 2054 + 1933 + 1839 + 1697;
100 Eggs[0] = 0 + 5070;
100 Larvae[0] = 3753 + 3594 + 3387 + 3135 + 2816;
100 Winged[0] = 2496 + 2516 + 2472;
100 Solitarious[0] = 2157 + 2004 + 1957 + 1878 + 1781 ;
100 Gregarious[0] = 2076 + 2238 + 2054 + 1933 + 1839;
100 Eggs[1] = 0 + 0;
100 Larvae[1] = 4550 + 3753 + 3594 + 3387 + 3135;
100 Winged[1] = 2751 + 2496 + 2516;
100 Solitarious[1] = 2446 + 2157 + 2004 + 1957 + 1878 ;
100 Gregarious[1] = 2420 + 2076 + 2238 + 2054 + 1933;
100 Eggs[2] = 0 + 0;
100 Larvae[2] = 0 + 4550 + 3753 + 3594 + 3387;
100 Winged[2] = 3098 + 2751 + 2496;
100 Solitarious[2] = 2489 + 2446 + 2157 + 2004 + 1957 ;
100 Gregarious[2] = 2462 + 2420 + 2076 + 2238 + 2054;
:

```

FIGURE 4.7: Trace de simulation : evolution de la densité des individus sur chaque état et micro-état.

Les mécanismes simulés sont déduits d'études bibliographiques, le modèle attend d'être validé en utilisant les données de terrain. Nous avons noué un contact avec le centre national de lutte antiacridienne (CNLA) de la Mauritanie qui dispose d'un ensemble de données acridiennes que nous espérons utiliser.

Le fragment de code suivant montre les transitions spatiales ayant lieu entre les cellules. Chaque cellule dispose de huit canaux d'entrée et huit canaux de sortie. Les migrants passent par ces canaux conformément aux conditions éco-météorologiques.

```

PROC ShiftTabRand ([ ] INT tab , VAL INT percent )
  INT indexCopy , previous , lost , perte :
  INT iPercent :
  SEQ
    iPercent := percent
    perte := percent
    indexCopy := (SIZE tab) - 1
    lost := 0
    perte := 0
  WHILE indexCopy > 0
    previous := tab[indexCopy - 1]
    lost , Seed := random(iPercent , Seed)
    perte := (previous * perte) / 100
    previous := previous - perte
    tab[indexCopy] := previous
    indexCopy := indexCopy - 1
  :
PROC LifeEggs (INT DebugLevel)

```

```

INT lastIndex , number :
REAL32 outgoing , outgoingRate :
SEQ
  lastIndex := (MaxEggsLifePeriod) - 1
  number := cP[eggsPeriod][lastIndex]
  outgoingRate := 0.966
  outgoing := outgoingRate * number
  Limit := INT ROUND outgoing
  TransferToFromClearing (cP[larvae1Period][0] ,
                          cP[eggsPeriod][lastIndex] , Limit)
  ShiftTabRand (cP[eggsPeriod] , 96)
  cP[eggsPeriod][0] := 0
:

```

4.5 Les résultats de simulation

Les simulations ont porté sur deux scénarii, dans le premier, une cellule choisie arbitrairement, a été initialisée avec un certain nombre d'individus matures afin d'étudier les interactions entre cellules et le second a consisté à initialisé un micro-état dans le but de simuler les différents stades de développement du criquet pèlerin.

La table ci-dessous résume les paramètres de simulation utilisés dans les deux modèles. Elle montre les taux de mortalités relatifs à chaque stade de développement du criquet pèlerin, les durées de vie des différents stades et le nombre de cellules, de canaux entrant et sortant du système cellulaire. Les données de mortalités sont fournies dans le rapport de Roffey sur les criquets pèlerins [202].

Durée de vie maximale (jours)				
œuf	Larve	Ailé	Solitaire	Grégaire
12	30	18	30	30
Taux de mortalité (%)				
15	12	10	9	9
Système cellulaire				
Cellules	minFanout	maxFanout	Canaux	Topologie
798	2	8	6014	Moore 1

TABLE 4.2: Les paramètres de simulation

4.5.1 Illustration de la migration inter cellulaire

Dans ce scénario, une cellule est initialisée avec une fonction aléatoire qui a généré un certain nombre d'œufs. Durant les 15 premiers tours synchrones, nous remarquons

que la population a évolué exponentiellement avec des périodes de pointe, depuis l'état œuf, jusqu'à l'état ailé immature. Après 16 tours synchrones, des ailés matures émergent. Les nouveaux œufs qui apparaissent sont dus à la ponte des femelles matures.

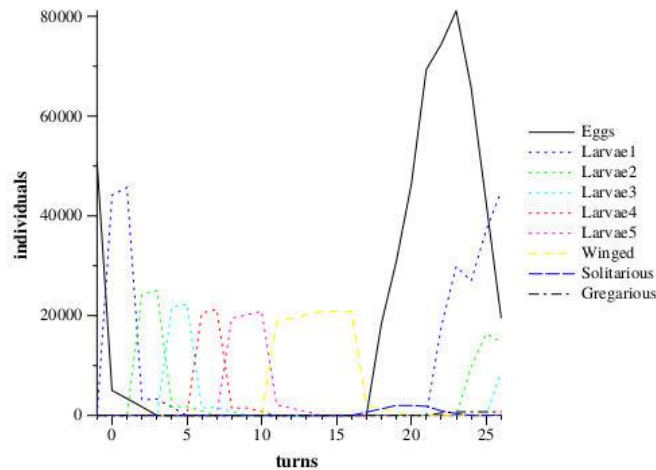


FIGURE 4.8: Transfert de migrants dans une cellule

Par rapport aux études précédentes basées sur des modèles numériques, nous constatons que nous avons les mêmes allures de courbe dans l'évolution des populations. La différence fondamentale réside dans le fait que nos cellules sont capables d'embarquer quantité d'autres informations circonstancielle. La direction et la vitesse du vent, la température et la pluviométrie ont ainsi été extraits de fichiers GRIB publics. D'autres données concernant les traitements chimiques pourraient y être ajoutés, par exemple.

4.5.2 Illustration du développement dans une cellule

Dans ce deuxième scénario, une cellule est initialisée avec 100 000 œufs de premier état, c'est à dire placés dans le premier micro état, $cP[eggsPeriod][0]$.

Les premiers 9 tours synchrones montrent l'évolution des œufs, des différents stades larvaires et des ailés immatures. Ensuite, ces ailés deviennent matures et les femelles pondent des œufs dans les cellules d'accueil. On remarque que chaque étape est faite de micro états dans lesquels le nombre d'individus décroît. Après 86 tours synchrones, c'est à dire 258 jours, de nouveaux œufs sont pondus par les premiers individus ailés matures provenant d'autres cellules.

4.5.3 Illustration de la propagation spatiale

Les deux figures (Voir Figures 4.10 et 4.11) illustrent la propagation spatiale qui a lieu entre les cellules après 30 et 60 tours synchrones.

Après 30 tours synchrones, la densité, l'état et la position des insectes peuvent être déterminés. Les cellules sont en plus géolocalisées ce qui permet de garder par exemple

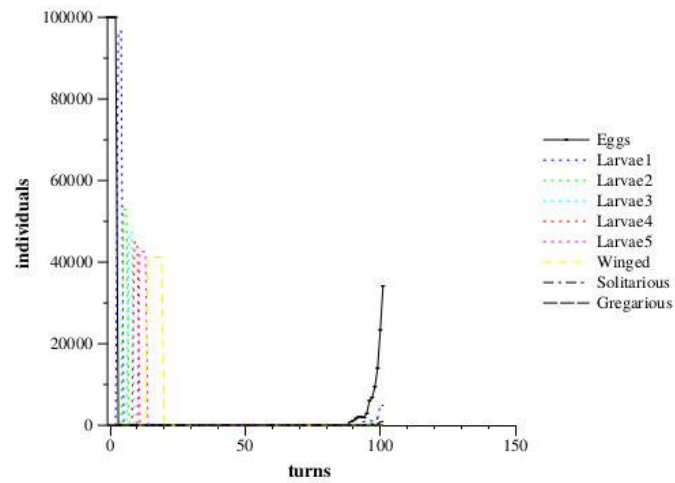
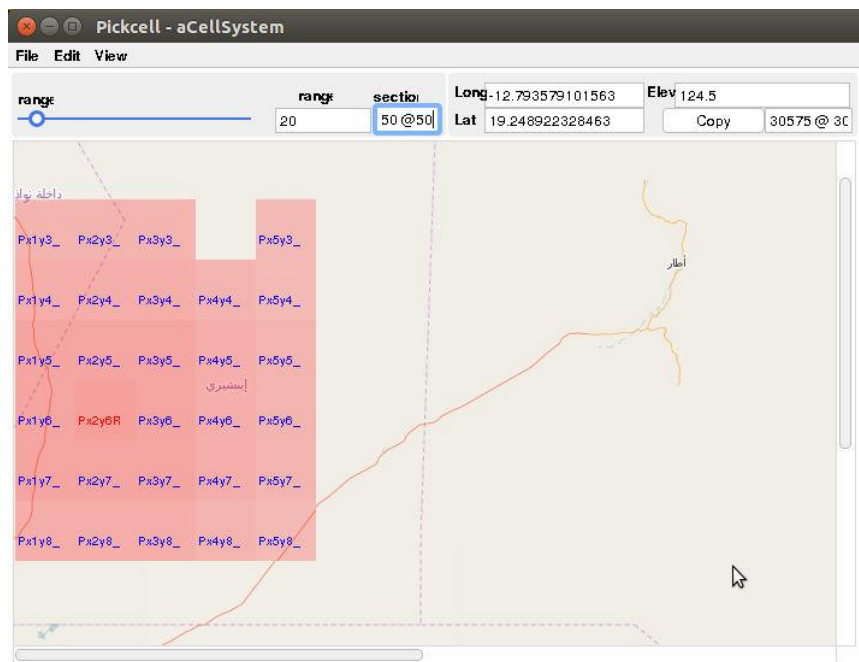


FIGURE 4.9: Evolution des individus dans une cellule

FIGURE 4.10: Propagation spatiale au tour synchrone 30 autour de la cellule P_{x2y6R}

une historique du mouvement des insectes dans leur évolution.

Les cellules sont progressivement occupées de manière radiale les unes à la suite des autres. Cette progression radiale s'explique par le fait qu'avec cet exemple les conditions relatives aux paramètres éco-météorologiques n'étaient pas implémentées à ce stade.

Sur la seconde figure 4.11, on remarque que le nombre de cellules occupées augmentent en fonction du nombre de tours synchrones.



FIGURE 4.11: Propagation spatiale au tour synchrone 60 autour de la cellule P_{x2y6R}

4.6 Conclusion

Ce chapitre a porté d'abord sur la représentation cellulaire des populations de criquets qui a permis à un découpage spatial et temporel d'une zone de reproduction de ces insectes. Ce découpage a permis de générer un système cellulaire en utilisant Pickcell.

Ces systèmes cellulaires représentent les modèles d'évolution locale et de migration. Le premier a permis de représenter le cycle de vie du cricquet pèlerin dans les cellules et le second les phénomènes de migration de ces insectes. Pour cela, nous avons utilisé des AC en utilisant des règles de transition calées sur la durée de vie des différents stades de développement des individus. Cette modélisation a abouti à un simulateur de dynamique de population formé de cellules communicantes. Ce travail a porté sur les criquets pèlerins, mais il peut être étendu à d'autres types d'insectes.

Ce sont des modèles cellulaires paramétriques dans lesquels nous avons pu intégrer des informations relatives aux paramètres climatiques tels que la température, le vent et la pluviométrie. L'intégration des paramètres climatiques dans les cellules du modèle permet d'améliorer la précision des simulateurs. On aura plus besoin de simuler conjointement plusieurs modèles parallèles, car ces paramètres apparaissent directement dans la description abstraite des réseaux.

Enfin, les résultats de simulations relatifs à deux scénarios le premier portant sur l'évolution locale des criquets dans une cellule et le second sur la migration des individus matures. Les résultats obtenus sont encourageants comparés aux travaux antérieurs relatifs à la dynamique de population du cricquet pèlerin. Nous avons obtenus des courbes sous forme de pipeline avec des périodes d'apparition et de disparition de certains états du cricquet pèlerin.

5

Simulations parallèle et distribuée

5.1 Introduction

Le chapitre 4 a porté sur la représentation cellulaire de la dynamique de population de criquets et à la présentation de résultats de simulation des deux modèles, le présent chapitre sera consacré à la montée en puissance de calcul dans deux directions : l'usage de parallélisme massif tel qu'il peut apparaître sur les accélérateurs graphiques GPU, la composition de plusieurs simulateurs communicants, avec *High Level Architecture* (HLA). Ces deux techniques complémentaires permettent de répondre à plusieurs besoins apparus dans la simulation des locustes, si nous l'envisageons à grande échelle. Elles ont été pratiquées.

5.2 La simulation distribuée

La simulation séquentielle peut se révéler inadaptée pour les modèles complexes comportant plusieurs composants. La simulation distribuée permet une répartition des simulations sur plusieurs ordinateurs ou processeurs facilitant la réutilisation des composants. La norme IEEE HLA permet de subdiviser la simulation en systèmes fédérés qui sont des simulateurs individuels interconnectés via un bus RTI. Son principal objectif est de permettre l'interopérabilité et la réutilisabilité des composants de simulations.

Une architecture distribuée est relative à un ensemble d'ordinateurs autonomes interconnectés via un réseau et partageant des ressources réparties sur ce dernier. Cet ensemble apparaît comme un seul système homogène. Le format de représentation des données et leur emplacement importent peu. L'utilisateur n'a pas à se soucier du format de représentation des données et comment il peut accéder aux ressources. Il ne connaît

pas non plus l'emplacement des ressources. Les ressources peuvent être déplacées sans modifier leur nom. Plusieurs utilisateurs peuvent partager en même temps les mêmes ressources. Des tâches peuvent s'exécuter en parallèle à l'insu des utilisateurs. Ceci permet d'augmenter les ressources. Les ordinateurs fonctionnent indépendamment l'un de l'autre. Ce qui pose le problème de la coordination des activités, elle est critique dans ce type d'architecture.

5.2.1 Le couplage par HLA et la puissance par CUDA

Le couplage de CUDA et HLA permet de simuler de manière parallèle et distribuée plusieurs modèles. Ce couplage nous a permis d'exécuter nos modèles de population, de climat, de réseaux de surveillance et de système de visualisation.

Plusieurs travaux ont porté sur la co-simulation distribuée, parmi lesquels, on peut citer les travaux de Lasnier et Al [142]. Dans cette publication les auteurs ont décrit un cadre basé sur le standard IEEE HLA, pour la simulation distribuée de CPS avec un grand nombre de composants hétérogènes. Ces composants hétérogènes sont généralement conçus avec différents outils qui ne sont pas facilement intégrables et qui peuvent influencer négativement sur les temps de simulation globaux.

Dans une deuxième publication [117], une approche de fédération de simulations dans le cadre de modèles de systèmes physiques programmés par automates cellulaires, est présentée. Le phénomène physique concernait la diffusion de polluants sur une rivière. Nous avons suivi une démarche similaire pour le cas le plus complexe et plus étendu des criquets pèlerins dans leur environnement physique.

Nos modèles représentent une population de criquets dans leur milieu de reproduction représenté sous forme de cellules. Les insectes peuvent se déplacer de cellule en cellule à la recherche de nourriture et de conditions éco-météorologiques favorables. Les paramètres éco-météorologiques influent fortement sur les modèles de migration et le développement des criquets pèlerins. Ces interdépendances en font des modèles complexes à large échelle. La simulation des modèles complexes à large échelle est souvent confrontée à des problèmes de performance car elle requiert de longs temps de simulation. De plus, les interopérabilités entre les composants sont souvent mal intégrées.

Le traitement parallèle par CUDA peut aider à décupler les performances de calcul du système en exploitant la puissance des processeurs graphiques (GPU) [175]. Cette technique repose sur une évolution du traitement exclusif par CPU vers les capacités de co-traitement offertes par l'association du CPU et du GPU qui sont chargés d'exécuter les calculs à la place du CPU.

Le calcul parallèle est utilisé dans plusieurs domaines, comme notamment dans le traitement des images et des vidéos, la chimie et la biologie, dans la modélisation numérique, la mécanique des fluides et bien plus encore. Toutefois, un problème subsiste lorsqu'il s'agit de faire communiquer les modèles entre-eux, c'est le manque d'interopérabilité entre les différentes simulations à exécuter. C'est dans ce contexte que le standard IEEE HLA, se positionne comme une solution à ce problème [106, 89].

Nous avons procédé à une subdivision de ce modèle complexe en sous modèles.

Ces sous modèles ont été exécutés de manière parallèle en s'appuyant sur la norme IEEE HLA. Cette dernière permet de faire de la simulation distribuée avec des échanges d'informations entre les différents simulateurs.

Afin de garantir une meilleure performance pour la simulation des modèles complexes, une ferme de processeurs graphiques, a été utilisée afin de fournir des simulations indépendantes dans un contexte de systèmes distribués synchrones et d'accélérer l'exécution de centaine ou de milliers de processus physiques qui peuvent composer ces modèles.

La co-simulation parallèle et distribuée a été utilisée pour minimiser le temps de simulation. Elle repose sur l'utilisation de fermes de processeurs graphiques et du standard HLA. La norme HLA permet de gérer les interopérabilités entre les différents composants. Cette approche est bien adaptée à nos systèmes physiques car elle a permis l'exécution simultanée de plusieurs processus physiques.

Ces simulations requièrent beaucoup de puissance de calculs car le nombre de cellules peut être très important pouvant atteindre des milliers de processus à exécuter.

5.2.2 Fonctionnement du HLA

HLA est un standard IEEE normalisant la multi-simulation éventuellement distribuée. Elle permet donc une subdivision de grands modèles complexes en sous-modèles en spécifiant le séquençement et les interactions entre eux [119].

HLA a été déployé dans plusieurs scénarios de simulation au cours de ces dernières années, parmi lesquels, nous pouvons citer les cas du domaine de transport et de l'industrie [43]. À notre connaissance, il n' a jamais été utilisé dans la simulation des phénomènes physiques naturels tels que le changement climatique ou la dynamique de population. Nous utilisons HLA pour faciliter la composition de simulations distribuées et concurrentes.

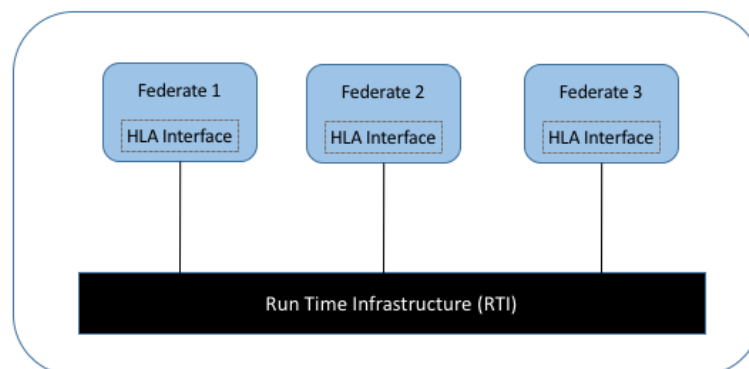


FIGURE 5.1: Architecture HLA : Une fédération de trois simulateurs reliés à un bus "Run Time Infrastructure" (RTI) (Source [229]).

Sur la figure 5.1, Federate 1, Federate 2 et Federate 3 représentent trois simulations à exécuter parallèlement. Ces simulations peuvent tourner sur des systèmes d'exploitation

et machines différents. Elles échangent des données par l'intermédiaire du bus RTI. Le composant *HLA interface* représente les points d'échange entre les simulations et le bus RTI.

Les spécifications du standard HLA sont un ensemble de règles qui définissent les comportements des simulateurs fédérés et leur relation avec le bus RTI [119]. Elles caractérisent aussi les interfaces qui représentent les points d'accès aux services par lesquels les simulateurs communiquent avec le bus [118]. C'est un modèle objet basé sur le standard "Object Model Template" (OMT) [120] qui définit comment les informations sont échangées entre les simulateurs fédérés, et comment les simulateurs et la fédération sont documentés par le biais du "Federation Object Model" (FOM). Ce dernier définit toutes les classes d'un objet, les attributs et les interactions de la fédération.

Les services proposés sont les opérations de création, de maintenance et de suppression de simulateurs, d'une part et d'autre part de publication, de souscription à des attributs de classe servant à stocker des données et à gérer les interactions avec le bus RTI.

Les simulateurs fédérés peuvent s'échanger deux types d'éléments, l'objet et l'interaction. Un *objet* représente une entité de la simulation qui contient des données partagées par les simulateurs fédérés. Ces données persistent jusqu'à la suppression des simulateurs. Une *interaction* est un message de diffusion que tout simulateur fédéré peut émettre ou recevoir. Un simulateur publié envoie une interaction aux autres simulateurs fédérés déjà souscrit au bus RTI. Si aucun simulateur fédéré souscrit ne reçoit l'interaction, les données transportées sont perdues (voir Figure 5.2). Si un simulateur fédéré veut publier ou souscrire à une interaction, il doit d'abord définir cette interaction dans son FOM.

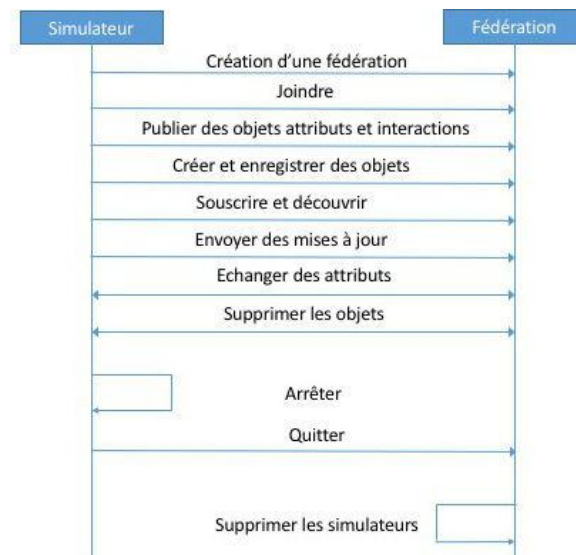


FIGURE 5.2: Interactions entre un simulateur et une fédération (Source [229]).

Un simulateur peut créer ou joindre une fédération via le bus RTI.

Les interactions et objets publiés sont disponibles pour tous les simulateurs de la fédération. Un simulateur fédéré, avant d'envoyer des données, doit d'abord s'enregistrer

et invoquer un service de mise à jour. Ces données sont automatiquement transmises par le bus RTI et les ressources allouées sont libérées à la fin de la transaction. Ce fonctionnement est orienté connexion avec une reservation de ressources entre un simulateur et le bus.

5.2.3 La gestion des horloges avec HLA

Le bus RTI offre une variété de services de gestion temporelle. Ces services permettent de coordonner les échanges d'événements entre les simulateurs fédérés. Chaque simulateur fédéré génère une horloge locale qui est communiquée au bus RTI. Ce dernier est chargé d'assurer leur coordination selon une politique temporelle. Chaque simulateur fédéré est donc impliqué dans la politique d'évolution temporelle.

La synchronisation des différents simulateurs fédérés est nécessaire car plusieurs simulateurs se partagent un bus unique. Cela permet aux simulateurs fédérés d'échanger des données dans un ordre défini par le bus RTI.

Les services d'évolution temporelle Le gestionnaire temporel est chargé de faire progresser le temps de simulation dans chaque simulateur. Pour cela, il fournit trois services que sont :

- "*timeAdvanceRequest*", chargé d'implémenter les bases de temps ;
- "*nextEventRequest*", chargé d'implémenter les évènements ;
- "*timeAdvanceGrant*", chargé d'accorder du temps aux simulateurs.

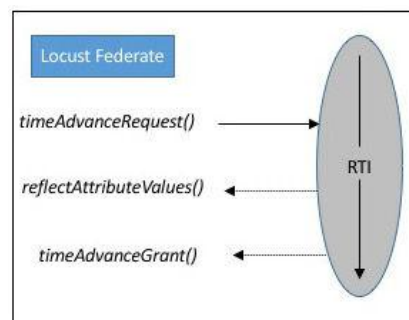


FIGURE 5.3: Progression temporelle dans le simulateur (Source [229]).

Un cycle de gestion du temps consiste en trois étapes. D'abord, un simulateur fédéré envoie une demande de progression temporaire. Ensuite, les simulateurs fédérés peuvent recevoir des appels comme "ReflectAttributeValues". Enfin le bus RTI complète le cycle en invoquant la procédure "timeAdvanceGrant" pour indiquer qu'il y a eu progression temporaire

La synchronisation des simulations Il est nécessaire de synchroniser les simulations pour une communication efficace avec le bus RTI. Pour cela, deux paramètres sont activés pour toutes les simulations. Ce sont les paramètres de contrainte et de régulation

temporelle. Le premier permet aux simulateurs d'envoyer des mises à jour et des interactions et le second leur permet de recevoir les mises à jour et les interactions du bus RTI.

Il existe un simulateur fédéré particulier, appelé, *Visualisation*, qui n'a aucun impact sur la progression temporelle d'une fédération. Son paramètre de régulation temporelle peut être désactivé comme il reçoit aucune mise à jour (voir Tableau 5.1).

Simulateurs	Contrainte	Régulation	Evolution temporelle
Locust	OUI	OUI	Par étape
Climat	OUI	OUI	Par étape
Capteurs	OUI	OUI	Par étape
Visualisation	OUI	NON	Par étape

TABLE 5.1: Politique temporelle dans notre cas d'étude

Le bus RTI fournit des mécanismes de synchronisation des simulateurs lors des échanges de données, en associant des temps aux communications qui ont lieu sur des points de synchronisation bien définis.

La figure suivante (voir Figure 5.4) illustre un processus de synchronisation entre deux simulateurs fédérés : *Locust* et *Climat*.

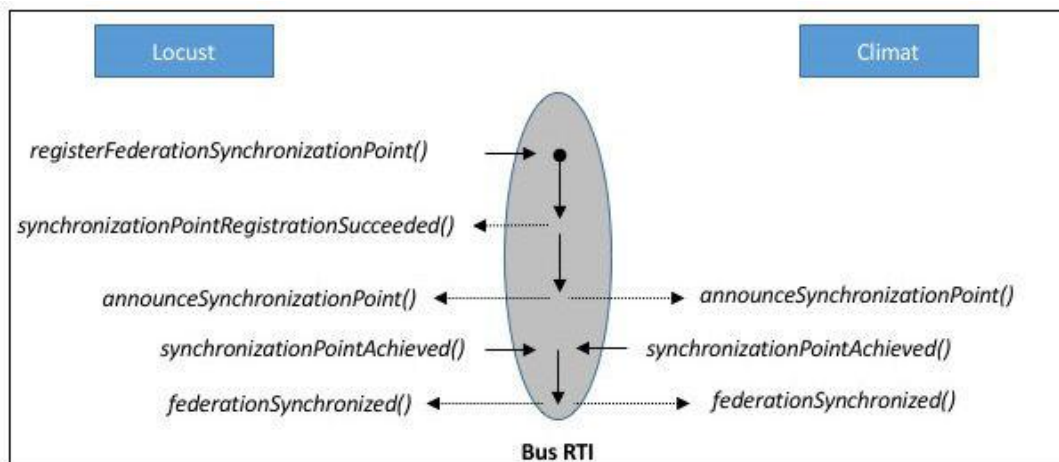


FIGURE 5.4: Communication entre simulateurs et le bus RTI

Tout d'abord, le simulateur *Locust* envoie une requête de synchronisation au bus RTI. Ensuite, le bus lui répond en annonçant des points de synchronisation. Un service sera alors, utilisé par les simulateurs fédérés pour confirmer le choix du point synchronisé.

5.2.4 La communication entre simulateurs

La communication est une caractéristique importante des systèmes distribués. Elle utilise des types de données particuliers déterminés par les caractéristiques des systèmes

réels et se produit à des moments bien définis.

Dans notre cas, quatre simulateurs sont fédérés, il s'agit des simulateurs *Locust*, *Climat*, *RCSF* et *Visualisation*. Le simulateur *Locust* transmet son statut (états et positions) aux simulateurs *Climat* et *RCSF*, par le biais de la classe d'objets *LocustNode* à laquelle ils doivent souscrire. Tous les trois simulateurs transmettent leurs données au simulateur *Visualisation* pour l'analyse et la visualisation des résultats.

Les simulateurs *Climat* et *RCSF* publient également leurs classes d'objets *ClimatNode* et *RCSFNode*, respectivement. Ces classes doivent d'abord être déclarées dans le FOM de la fédération, leur structure est présentée dans le tableau 5.2 ci-dessous.

Classes	Attributs	Publicateurs	Souscripteurs
<i>LocustNode</i>	Etat et positions	<i>LocustNode</i>	Climat, RCSF, Visualiseur
<i>ClimatNode</i>	Temperature, pluviométrie	<i>ClimatNode</i>	Visualiseur
<i>RCSFNode</i>	Etat et positions	<i>RCSFNode</i>	Visualiseur

TABLE 5.2: Les objets et attributs publiés par les simulateurs fédérés

Dans certains cas, il est important de préciser le lieu d'échange des données entre des simulateurs fédérés, en particulier dans le cas de systèmes physiques de grande dimension. En effet, il est souvent inefficace d'envoyer l'intégralité des données via le bus RTI en raison de problèmes de performance du réseau et de rendement de calcul local. Ainsi, nous proposons une nouvelle approche d'échange de données entre deux systèmes adjacents. Les systèmes adjacents sont caractérisés par l'existence de frontière commune ou de certains points communs. Avec cette approche, il n'est pas nécessaire d'envoyer certaines informations ayant par exemple, un sens local, aux autres simulateurs.

Par exemple, dans notre cas d'étude, nous avons supposé que la migration des criquets peut être causée par des conditions climatiques défavorables apparaissant à la frontière des systèmes *Locust* et *Climat*. Le simulateur *Climat* envoie régulièrement ses états au simulateur *Locust* qui ne considère que les états des points qui lui sont proches.

5.2.5 La simulation physique distribuée : cas des criquets

Le standard HLA est utilisé pour unifier des simulations parallèles. La simulation physique du modèle regroupe quatre simulateurs que sont *Locust*, *Climat*, *RCSF* et *Visualisation*. Ces simulations sont fédérées en accord avec le modèle de programmation CUDA et le standard HLA. Ensuite elles sont exécutées en parallèles comme quatre différentes simulations.

La migration des criquets d'une cellule à une autre est gouvernée par des règles de transition basées sur les conditions climatiques dans des cellules. Lorsqu'une cellule est occupée à l'instant t , le modèle climatique inclut ses nouvelles données à l'instant $t + 1$. En ce qui concerne la communication entre les simulateurs *RCSF* et *Locust*, les capteurs collectent régulièrement l'état d'une cellule. De nouvelles informations seront envoyées au simulateur *Visualisation* et si une densité de population seuil est atteinte,

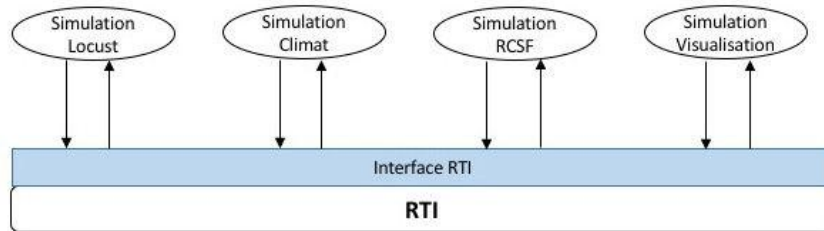


FIGURE 5.5: Structure de la fédération de simulateurs

Visualisation émet des signaux d'urgence. Pour cela, la synchronisation doit être réalisée comme indiqué dans le tableau 5.1 et les données partagées doivent être déclarées comme indiqué dans le tableau 5.2.

Le fichier FOM pour la fédération est présentée dans le fichier cyber.fed qui suit :

```
;; Cyber physical simulation
```

```
(Fed
  (Federation Cyber)
  (FedVersion v1.0)
  (Federate "Locust" "Public")
  (Federate "Climat" "Public")
  (Federate "RCSF" "Public")
  (Federate "visualisation" "Public")
  (Objects
    (Class ObjectRoot
      (Attribute privilegeToDelete reliable
        timestamp)
      (Class RTIprivate )
    ( Class LocustNode
      ( Attribute PositionX RELIABLE TIMESTAMP)
      ( Attribute PositionY RELIABLE TIMESTAMP)
      ( Attribute State RELIABLE TIMESTAMP)
    )
    (Class ClimatNode
      ( Attribute Temperature RELIABLE TIMESTAMP)
      ( Attribute Pluviometrie RELIABLE TIMESTAMP)
      ( Attribute Vent RELIABLE TIMESTAMP)
    )
    ( Class RCSFNode
      ( Attribute PositionX RELIABLE TIMESTAMP)
      ( Attribute PositionY RELIABLE TIMESTAMP)
      (Attribute State RELIABLE TIMESTAMP)
    )
  )
)
```

)

Le simulateur Locust Il est basé sur un modèle de migration de groupes de criquets pèlerins. Chaque cellule est initialisée aléatoirement avec un certain nombre d'individus et à chaque étape un nombre aléatoire de criquets se déplace vers les cellules voisines ayant reçu une plus grande quantité de pluie, par conséquent ayant un couvert végétal plus fourni. Cette migration dépend des conditions éco-météorologiques dans les cellules. Nous nous sommes intéressés aux paramètres vent, température, pluviométrie. Le dernier détermine la végétation et le taux d'humidité relative dans les cellules.



FIGURE 5.6: Simulation de migration cellulaire avec un voisinage de Von Neumann. La couleur rouge représente la propagation des criquets

Sur la Figure 5.6, la couleur verte foncée de la première image représente le nombre initial de criquets, choisi aléatoirement entre 0 et 150. La couleur rouge sur la deuxième image représente le nombre de criquets compris entre 300 et 350. Cet état est obtenu après dix tours synchrones. La couleur rouge sera plus vive au fur et à mesure de l'évolution de la simulation (voir Figure 5.7) caractérisant une densité de population plus élevée.

Les règles de transition La règle de transition tient compte du nombre d'insectes des cellules voisines, et choisit d'envoyer plus ou moins d'insectes aux voisins. Par exemple si on suppose qu'une cellule centrale reliée à ses voisins par Von Neumann 1, dispose de 100 criquets, sa voisine Nord, de 200, celle du Sud de 10 et celle de l'Est de 50 criquets. La cellule centrale calcule de façon aléatoire une quantité de sa population (exemple 20%) et les envoie avec une proportion qui est fonction du nombre de criquets reçus des voisins au tour précédent (exemple $200 + 10 + 120 + 50 = 380$). Ainsi, la voisine Nord recevra $200/380$, la voisine Sud $10/380$, celle de l'Ouest $120/380$ et celle de l'Est $50/380$.

Les messages représentent le nombre d'insectes d'une cellule à envoyer aux voisins. Ensuite, la cellule soustrait les insectes perdus de son effectif, et s'occupe des messages entrants. Puis, le nuage d'insectes bouge dans une direction choisie aléatoirement.

Après vingt tours synchrones, on obtient la Figure 5.7 ci-dessus :

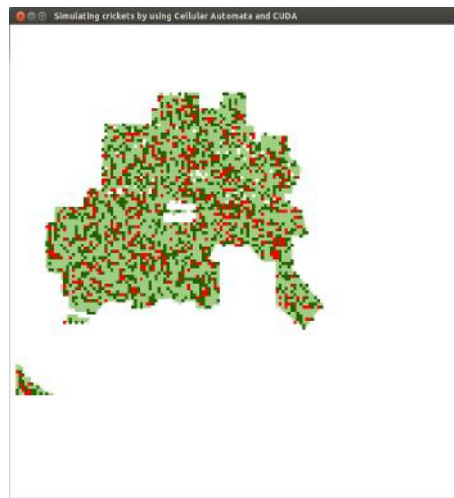


FIGURE 5.7: Etat des cellules après vingt tours synchrones

Ces figures (Figures 5.6 et 5.7) montrent la migration des criquets en rouge au début, milieu et fin de simulation. Les pixels dans les cellules représentent l'espace de données (la végétation) et une densité de population dans une cellule. La dégradation de la couleur verte signifie que la densité de population de la cellule devient de plus en plus faible. Ceci est dû à des conditions météorologiques défavorables.

Le simulateur Climat Initialement les cellules disposent d'un bon couvert végétal matérialisé par la couleur verte initiale des cellules. Durant la migration des criquets, ce simulateur reçoit les états des criquets du simulateur *Locust* par l'intermédiaire du bus RTI.

Il évalue le couvert végétal, comme les criquets se nourrissent de la végétation, la densité de végétation décroît proportionnellement avec la densité de criquets dans une cellule. Le bus RTI envoie les données au simulateur *Climat* dès qu'il reçoit des mises à jour du simulateur *Locust*. Ces mises à jour ont lieu lorsque les criquets atteignent les frontières d'une cellule.

Le simulateur RCSF À chaque étape synchrone, les nœuds capteurs recevront des données du simulateur *Locust* par l'intermédiaire du bus RTI. Seules les cellules dont la portée est comprise dans la plage de détection sont considérées. Lorsque la densité de criquets dépasse un certain seuil, le comportement du *RCSF* peut consister à expédier une alerte à un centre de traitement. Ainsi un réseau de surveillance synchrone basé sur les réseaux de capteurs sans fil permet de contrôler le mouvement des insectes. L'objectif de ces simulations est de prospecter les fonctions possibles des *RCSF*, en tenant compte de circonstances pratiques telles que les connectivités, ou de besoins, tels que la collecte synthétique des états réels. En première approche, le signalement d'un danger est toutefois un minimum significatif.

Le simulateur Visualisation Ce simulateur est basé sur un système de visualisation 2D présenté sur X11. Tout d'abord il s'enregistre au niveau du bus RTI et recueille toutes les données nécessaires qui ont été publiées par les autres simulateurs. Son rôle est de fournir une vue d'ensemble des résultats de simulation. Pendant l'exécution de la co-simulation, il reçoit des données des autres simulateurs et met à jour la présentation des résultats pour chaque étape.

5.3 Conclusion

Ce chapitre a porté sur les modélisations sur des architectures distribuée et parallèle. Il a permis de présenter les différentes architectures des calculateurs. La co-simulation distribuée avec couplage de HLA et CUDA a été évoqué avec une étude de cas sur la modélisation de la dynamique de criquets pèlerins dans leur milieu de reproduction.

L'intégration de données spécialisées requiert toutefois des efforts de programmation non triviaux. On conçoit bien que quantité de considérations conduisent à séparer les simulateurs : par exemple, l'effet des traitements, ou des accidents climatiques peuvent justifier des compositions de simulations.

Ce chapitre a permis également de montrer comment les paramètres climatiques ont été intégrés dans les cellules. Ce dernier aspect diffère nos modèles cellulaires de ceux existants car il permet de disposer de données de vent et pluie directement dans les cellules, évitant de disposer de modèles parallèles qui peuvent nécessiter des temps de simulation importants et d'utiliser des techniques de co-simulation distribuée.

6

Calcul de couverture radio et déploiement de capteurs

6.1 Introduction

Ce chapitre portera sur l'étude de couverture radio afin d'établir une carte de visibilité entre un ensemble d'émetteurs radio et un récepteur placé sur le point le plus haut de l'espace de l'université Gaston Berger (UGB) de Saint-Louis. Cet espace s'étend sur une superficie de 240 hectares et est caractérisée par la présence d'obstacles (arbres, bâtiments) qui peuvent provoquer des perturbations pouvant atténuer la puissance des signaux émis.

QuickMap, un navigateur de cartes ou vues graphiques compatibles avec NetGen, a permis de quadriller cette zone en choisissant des cellules avec différents niveaux de granularité. Les niveaux de granularité permettent de déterminer les dimensions des cellules. ils correspondent à la résolution spatiale.

Plus la résolution spatiale est grande, moins nous disposons de cellules dans nos systèmes cellulaires. Ceci permet d'accélérer le temps d'exécution des processus, mais avec moins de précision sur les positions des obstacles et des zones d'élévation. Nous savons que pour avoir une bonne précision des positions des objets, il faut segmenter la carte avec une résolution fine. Aussi, si la résolution est très fine, cela peut causer le ralentissement du délai d'exécution des programmes générés, dû à un nombre important de cellules dans les systèmes cellulaires.

En procédant à des simulations parallèles avec le GPU, on peut optimiser le délai d'exécution des programmes générés en CUDA. Donc il nous fallait trouver un compromis entre la résolution et le temps d'exécution.

En insérant les coordonnées géographiques de la tour de l'UGB qui est le point

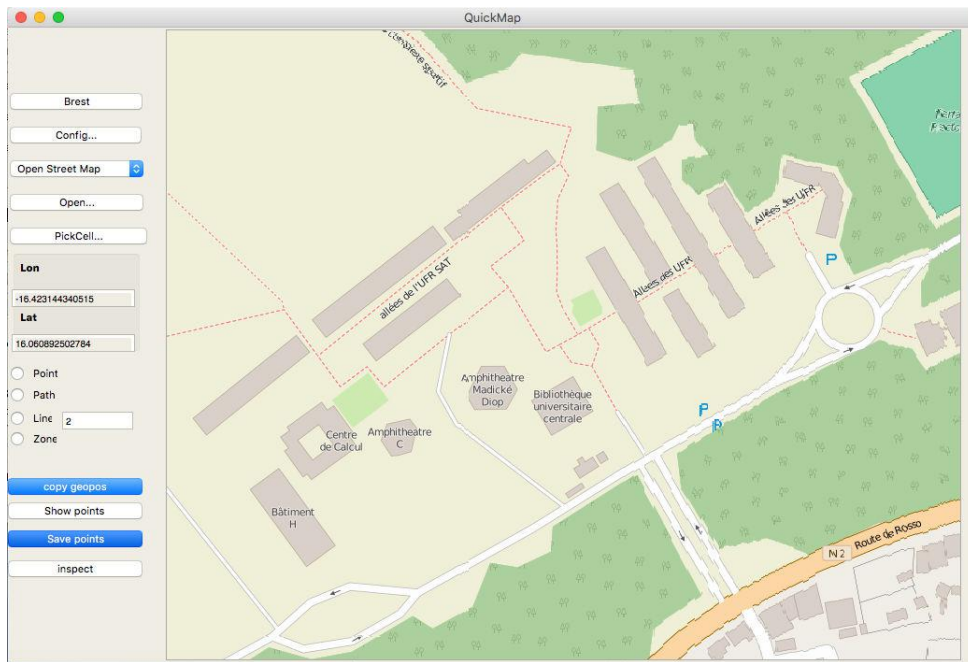


FIGURE 6.1: Une vue de l'UGB produite avec QuickMap. La tuile OpenStreetMap nous permet d'obtenir cette représentation. D'autres modèles de tuile existent.

culminant de la zone dans le fichier contenant les coordonnées géographiques des points d'élévation, nous avons pu établir une connectivité entre les points visibles et le récepteur. La tour de l'UGB est un immeuble de neuf étages et chaque étage fait trois mètres de haut, donc notre point culmine à une hauteur de vingt sept mètres.

L'intérêt d'une telle étude est de pouvoir planifier efficacement le déploiement d'un réseau de capteurs sans fil. C'est la même méthodologie utilisée par les opérateurs de télécommunication pour la planification de déploiement de réseaux cellulaires. Dans notre cas d'espèce, elle permet de déployer efficacement un réseau de capteur dans une zone donnée afin de récolter des informations sur le nuage d'insectes volants.

Nous avons étudié en fonction de la connectivité et des résolutions des cellules, les variations d'altitude dans cette zone, afin de déterminer les grains permettant d'avoir des valeurs d'altitude fiables corrélées à un temps d'exécution optimale des programmes. La fiabilité de ces valeurs est vérifiée, en injectant dans Googlemap des valeurs de longitude et de latitude prélevées dans le fichier.

La suite de ce chapitre sera organisée comme suit : la section 6.2 sera consacrée à la synthèse des systèmes cellulaires qui a permis la production de programmes transcrits en Occam et CUDA, dans la section 6.3, nous présenterons la méthode utilisée pour renseigner les valeurs d'élévations dans les cellules et les faire apparaître dans les programmes, la section 6.4 sera consacrée à la présentation des algorithmes classiques qui ont servi à la conception des comportements des modèles cellulaires, nous présenterons une étude de cas dans la section 6.5 avant de conclure en section 6.6.

6.2 La méthodologie de synthèse de systèmes cellulaires

La méthodologie de synthèse de système cellulaire suit la même procédure que celle décrite dans le cas des criquets pèlerins. Elle permet d'obtenir un réseau de cellules interconnectées selon une topologie donnée. Ici, nous avons considérés la topologie de Von Neumann 1, chaque cellule est liée à ses quatre voisins cardinales du Nord, Sud, Est et Ouest.

La génération de systèmes cellulaires avec les outils maison, peut être résumée en trois étapes. Tout d'abord, nous avons commencé par la sélection avec le navigateur QuickMap, d'une carte de la zone d'intérêt. Les données géographiques sont déjà inclus dans nos outils grâce à un utilitaire en ligne, GPSVisualizer. Ainsi les positions géographiques des cellules sont repérées. L'invocation de la commande "PickCell" du menu de QuickMap permet de lancer l'outil de segmentation cellulaire, présenté sur la figure (Voir Figure 6.2). Puis avec la commande "GetGridCell" de PickCell, nous avons procédé à la segmentation de l'espace en cellules dont les dimensions sont réglables dans la rubrique section du navigateur. Les cellules ayant des caractéristiques similaires sont regroupées en classes selon des critères basés sur les composantes rouge vert bleu de la couleur.

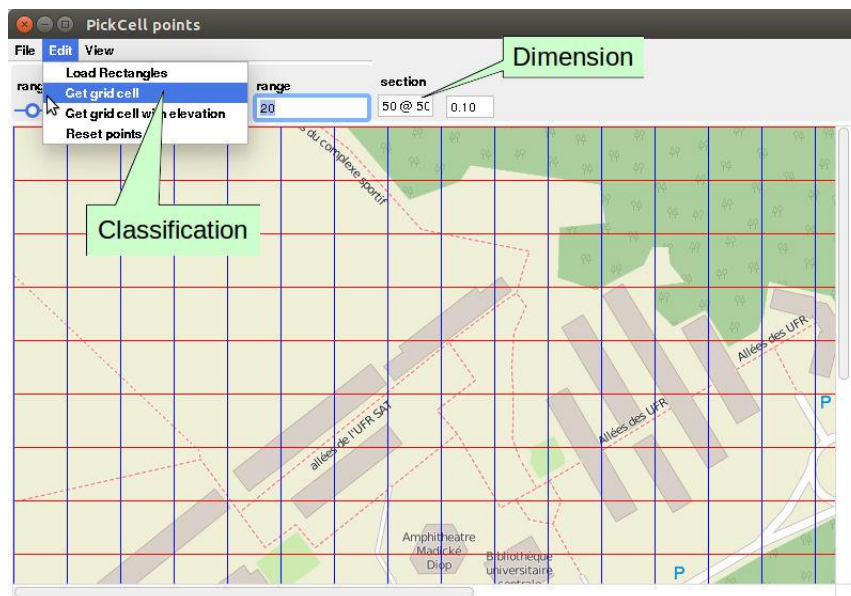


FIGURE 6.2: Segmentation spatiale. Nous utilisons une grille de dimension 50@50 qui permet de segmenter la vue graphique en 50×50 pixels.

Enfin, nous avons fait varier les voisinages et les dimensions des cellules pour produire des architectures abstraites représentant des descriptions du réseau en Occam. Notons aussi que ces descriptions peuvent être transcrites en CUDA pour des besoins de simulations parallèles.

L'évolution des processus physiques est définie dans des fichiers de comportement. Ce sont généralement des règles de transition standard relatives aux AC, qui sont beaucoup

utilisés en systèmes distribués, quelques exemples sont :

- le calcul de diamètre, qui représente la plus grande distance entre deux cellules ;
- la détermination de leader, pour nommer et différencier les réseaux non connexes ;
- le routage de données entre réseaux connexes ;
- et l'acheminement de données.

Si nous supposons un système cellulaire produit à partir d'une carte. Il existe plusieurs manières de déterminer les cellules voisines. Cette tâche est simplifiée par l'utilisation de modèles de connectivité définis dans les AC. En effet ces derniers utilisent différents modèles de connectivité pour définir le voisinage.

La classification des cellules aboutit à la création de collections de cellules annotées par des coordonnées géométriques et géographiques et des données d'élévation permettant de retrouver des cellules voisines.

6.3 Acquisition de données d'élévation

Nos modèles intègrent des données d'élévation qui permettent de définir les positions des objets sur la carte par rapport au niveau de la mer. Aussi ces données peuvent servir de critères de classification pouvant être utilisées pour extraire des groupes de cellules qui seront, par exemple, à zéro, dix mètres au dessus du niveau de la mer.

Avec GPS Visualizer, nous avons pu acquérir les données d'élévation de la zone d'étude grâce aux coordonnées géographiques, ainsi, des cartes et profils de la zone ont pu être créés. Les informations d'entrée sont des points relevés au GPS, des itinéraires de routes, des adresses ou coordonnées géographiques.

Il existe deux méthodes pour accéder aux données d'élévation, la première consiste à utiliser des bases de données de modèles d'élévation numérique et la seconde à utiliser des API google dont la contrainte principale est le nombre limité de requêtes qui nous est imposé lors des tests.

6.3.1 Les modèles numériques d'élévation, de terrain et de surface

Il n'existe pas encore de nomenclature standard relative aux concepts de modèle numérique d'élévation (MNE), modèle numérique de terrain (MNT) et modèle numérique de surface (MNS).

Généralement, un MNS représente la surface terrestre avec tous les objets tels que les bâtiments, les arbres, le relief, alors qu'un MNT, contrairement au MNS, représente la surface terrestre sans ces objets [151]. Un MNE est un terme générique pour désigner soit un MNS soit un MNT avec seulement des données d'élévations [114].

Dans la suite, nous considérerons le MNE comme la somme des deux autres, comme le font la plupart des fournisseurs de données.

En effet les fournisseurs de données comme "United States Geological Survey" (USGS) , " Earth Remote Sensing Data Analysis Center" (ERSDAC), par exemple, utilisent le terme MNE comme étant un terme générique désignant un MNS et un MNT.

type	latitude	longitude	altitude (m)	name
W	18.483516449	-16.051025391	2.5	Px1y1
W	18.483516449	-16.044158936	3.5	Px2y1
W	18.483516449	-16.037292480	3.7	Px3y1
W	18.483516449	-16.030426025	3.6	Px4y1
W	18.483516449	-16.023559570	3.2	Px5y1
W	18.483516449	-16.016693115	3.1	Px6y1
W	18.483516449	-16.009826660	1.4	Px7y1
W	18.483516449	-16.002960205	3.2	Px8y1
W	18.483516449	-15.996093750	2.5	Px9y1
W	18.483516449	-15.989227295	-0.4	Px10y1
W	18.483516449	-15.982360840	-1.4	Px11y1
W	18.483516449	-15.975494385	1.9	Px12y1
W	18.483516449	-15.968627930	0.0	Px13y1
W	18.483516449	-15.961761475	-1.7	Px14y1
W	18.483516449	-15.954895020	-2.7	Px15y1
W	18.483516449	-15.948028564	-1.0	Px16y1
W	18.483516449	-15.941162109	-2.0	Px17y1
W	18.483516449	-15.934295654	-0.8	Px18y1
W	18.483516449	-15.927429199	-2.8	Px19y1
W	18.483516449	-15.920562744	-1.9	Px20y1
W	18.483516449	-15.913696289	-1.8	Px21y1
W	18.483516449	-15.906829834	-2.1	Px22y1
W	18.483516449	-15.899963379	-0.8	Px23y1
W	18.483516449	-15.893096924	-1.9	Px24y1
W	18.483516449	-15.886230469	-0.3	Px25y1
W	18.483516449	-15.879364014	-2.5	Px26y1
W	18.483516449	-15.872497559	-2.8	Px27y1
W	18.483516449	-15.865631104	-1.9	Px28y1
W	18.483516449	-15.858764648	-3.5	Px29y1
W	18.483516449	-15.851898193	-1.2	Px30y1
W	18.483516449	-15.845031738	-2.5	Px31y1
W	18.483516449	-15.838165283	0.6	Px32y1
W	18.483516449	-15.831298828	-1.0	Px33y1
W	18.483516449	-15.824432373	-1.2	Px34y1
W	18.483516449	-15.817565918	-2.1	Px35y1
W	18.483516449	-15.810699463	-2.0	Px36y1
W	18.483516449	-15.803833008	-0.0	Px37y1
W	18.483516449	-15.796966553	-1.7	Px38y1
W	18.483516449	-15.790100098	0.2	Px39y1
W	18.483516449	-15.783233643	-1.1	Px40y1
W	18.483516449	-15.776367188	-1.6	Px41y1

FIGURE 6.3: Sur la figure, apparaissent les valeurs d'élévations par rapport au niveau de la mer, résolues par GPS Visualizer. Cette résolution se fait sur la base des positions géographiques des points.

Les MNS sont obtenus par des techniques de télédétection via des satellites ou avions. Comme exemple, on peut citer le cas du "Shuttle Radar Topography Mission" (SRTM) et du "Advanced Spaceborne Thermal Emission and Reflection Radiometer, Global Digital Elevation Model" (ASTER GDEM).

Il est possible d'obtenir des MNT de haute résolution à partir de MNS par des traitements algorithmiques complexes [151].

6.3.2 Exemple d'utilisation d'un MNE

Un MNE est un ensemble de cellules sous forme carré appelées "raster". Il est souvent utilisé dans les systèmes d'information géographique, et permet la production numérique de cartes en relief, alors qu'un MNS est utilisé dans les modélisations du paysage, de la ville. Un MNT est utilisé dans la modélisation des phénomènes d'inondations, des études de paysage ou d'applications géologiques [27]. C'est un bon candidat pour la représentation cellulaire des phénomènes d'inondations que les outils du laboratoire peuvent offrir.

SRTM dans Pickcell

Il existe une variété de fichiers MNE qui couvrent tout ou une partie du globe terrestre. Ces fichiers se différencient par leurs degrés de résolutions. Par exemple le "GTOPO30" est un MNE libre qui couvre la terre entière avec une résolution de 30 arcsecondes, correspondant à environ 1 km. La qualité fournie est variable et reste très pauvre dans certains continents comme en Afrique. L'ASTER est un autre type de MNE de meilleure qualité que le premier, disponible gratuitement pour 99% du globe terrestre, et qui représente l'élévation avec une résolution d'un arcseconde (environ 30 mètres). Seuls les Etats Unis disposent d'un MNE de haute résolution dans le cadre du "Shuttle Radar Topography Mission" (SRTM), pendant que le reste de la planète est seulement couvert avec une résolution de 3 arcsecondes (environ 90 mètres). Les modèles "GTOPO30" et SRTM couvrent seulement des domaines continentaux. Le dernier exclut aussi les régions polaires, les montagnes et les déserts.

GPS Visualizer utilise du SRTM pour produire des valeurs d'élévation avec une résolution de 3 arcsecondes. PickCell utilise du SRTM comme MNE.

6.4 Couverture et algorithmes distribués

Les algorithmes utilisés dans la définition des comportements des modèles cellulaires sont des scénarios basiques qu'on retrouve dans la plupart des ouvrages de systèmes distribués [158, 62].

Par exemple, l'algorithme de détermination du leader d'un réseau, basé sur la comparaison des identités des nœuds, consiste à faire prendre connaissance à un nœud mobile, l'identité des nœuds traversés lors du parcours du réseau. Le mobile les compare à son identité et si cette dernière est plus grande, il est élu leader.

Un autre algorithme portant sur le calcul du diamètre d'un réseau, qui est la distance maximale entre deux nœuds quelconques, permet de connaître le nombre de tours nécessaires à la diffusion d'une information sur le réseau. Ceci est important car si nous propageons un message sur le réseau, nous sommes sûrs qu'au bout d'un diamètre tour, tous les nœuds auront reçu ce message.

Un troisième algorithme relatif à propagation de route est basé sur du "Breadth First Search" (BFS), un algorithme de parcours en largeur d'un graphe. BFS permet de parcourir un graphe de manière itérative, en utilisant une file. Il peut par exemple servir à déterminer la connexité d'un graphe.

D'autres algorithmes sont également utilisés pour le routage et le repérage de la position des cellules via leurs coordonnées géométriques.

6.4.1 Algorithme du maximum distribué

En systèmes distribués, cet algorithme permet la désignation d'un nœud unique comme organisateur d'une tâche répartie entre plusieurs nœuds d'un réseau. Par exemple

la génération d'un jeton permettant d'attribuer du temps de parole aux nœuds désirant transmettre.

Le principe consiste à faire disposer chaque nœud du réseau d'une variable locale "MaxLoc", son identité. Chaque nœud transmet un message, le maximum global "Mx-Global" qui est défini initialement à son "MaxLoc". A la réception, d'un message, si "MaxGlob" est inférieure à "MaxLoc", le message est ignoré, sinon le nœud enregistre la nouvelle valeur reçue comme le "MaxGlob". Si ce dernier est égal à "MaxLoc", alors ce nœud est le leader.

La complexité de cet algorithme est N (nombre de nœuds) \times M (nombre de messages par nœud).

En exécutant n tours et en plaçant la variable "MaxGlob" dans les tampons sortants, avec n , le diamètre du réseau, tous les nœuds connaissent le "MaxGlob" en $\text{MaxNodes} - 1$ étapes. Plusieurs leaders peuvent exister si le réseau n'est pas connexe.

Listing 6.1: Communications, changement d'état sur MaxGlob et production de messages sortants

```

1  PROTOCOL diam.proto IS INT:
2  PROCNode([ ] CHANOF diam.proto in , out , VAL INT id ,
           CHANOFBYTE toMux)
3    INT MaxGlob:
4    INT inBuf , outBuf : BYTE isNull:
5    SEQ
6      MaxGlob := id
7      SEQ i=0 FOR SIZE in
8        outBuf [ i ] := MaxGlob
9      SEQ tour=0 FOR MaxNodes 1
10     SEQ
11     PAR
12       out.number(MaxGlob,4 ,toMux)
13       toMux ! '*n'
14 :
```

6.4.2 Algorithme de détermination du leader d'un réseau

Les nœuds identifiés de façon unique, communiquent entre eux afin de décider lequel sera leader. L'algorithme est basé sur une comparaison des identifiants. Le nœud ayant l'identifiant le plus élevé, est désigné leader du réseau. Au début, le leader est inconnu, de même que le nombre de processus. L'algorithme finit en n étapes et un et un seul processus s'élit leader.

Listing 6.2: Election d'un leader

```

1  PROTOCOL diam.proto
2  CASE
3    tab ; TableIdVal null ; BYTE
```

```

4     int; INT
5 :
6
7 PROC Node([N] CHAN OF diam.proto in ,out)
8     [N] INT tabInt ,tabOut: BYTE isNull:
9     SEQ
10    SEQ i=0 FOR N
11        tabOut [i] := i
12
13    PAR i=0 FOR N
14        out[i] ! int ; tabOut[ i]
15
16    PAR i=0 FOR N
17        in [i] ? CASE
18            int ; tabIn[i]
19            SKIP
20        null ; isNull
21        SKIP
22 :
```

6.4.3 Calcul du diamètre d'un réseau

Dans ce scénario, chaque nœud dispose d'une table où sont enregistrés les identifiant et distance des autres nœuds. Les messages sont envoyés à des files d'enregistrement contenant identifiant et distance.

Au début, les nœuds placent leur enregistrement dans la table, et on fait tourner l'algorithme N fois. A chaque tour, les nœuds regardent s'il y a un nouveau nœud qui s'est fait connaître. Si c'est le cas, il est enregistré localement dans la table en incrémentant la distance, puis il est rajouté dans une file d'enregistrement pour le tour suivant. Après N étapes, tous les nœuds se connaissent et connaissent la plus grande distance relative. Un maxglobal permet de connaître le diamètre.

Listing 6.3: Détermination du diamètre

```

1 DATA TYPE IdentifiedValue
2     RECORD
3         INT Id:
4         INT Val:
5 :
6
7 DATA TYPE TableIdVal
8     RECORD
9         INT limit:
10        [MaxNodes] IdentifiedValue tabNodes
```

```
11 :
12
13 PROTOCOL diam.proto
14   CASE
15     tab ; TableIdVal null ; BYTE
16     int ; INT
17 :
18
19 PROC AddInTable( TableIdVal thisTable , IdentifiedValue
    idVal)
20   INT limit:
21   SEQ
22     limit:= thisTable[ limit]
23     thisTable[ tabNodes][ limit] :=idVal
24     thisTable [ limit] := limit+1
25 :
26
27 PROC InitMyVal(IdentifiedValue myVal,INT aValue)
28   SEQ
29     myVal[Id] :=identity
30     myVal[Val] :=aValue
31 :
32 PROC InitMyTable(TableIdVal thisTable)
33   SEQ
34     thisTable[ limit] :=0
35     AddInTable(thisTable ,myVal)
36 :
37
38 PROC LookInTable(TableIdVal thisTable ,IdentifiedValue
    idVal,BOOL found)
39   INT limit:
40   SEQ
41     found:=FALSE
42     SEQ index= 0 FOR thisTable[ limit]
43     IF
44       thisTable[tabNodes][index][Id] =idVal[Id]
45       SEQ
46         idVal:= thisTable[tabNodes][index]
47         found:=TRUE
48     TRUE
49     SKIP
50 :
51
```



```

52 PROC MergeTable(TableIdVal localTable , TableIdVal
    inputTable , TableIdVal deltaTable)
53   INT limit:
54   BOOL found:
55   IdentifiedValue idVal:
56   SEQ
57     SEQ i=0 FOR inputTable[limit]
58       SEQ
59         idVal := inputTable[tabNodes][i]
60         LookInTable(localTable , idVal , found)
61         IF
62           NOT found
63             SEQ
64               idVal [Val ] := idVal [Val ] + 1
65               AddInTable(localTable , idVal)
66               AddInTable(deltaTable , idVal)
67           TRUE
68           SKIP
69 :
70
71 PROC Circulate(INT ivar)
72   TableIdVal deltaTable:
73   SEQ
74     InitMyVal(myVal , ivar)
75     InitMyTable(myTable)
76     SEQ i=0 FOR SIZE out
77       outMessages[i] := myTable
78     SEQ tours =0 FOR MaxNodes ?1
79     SEQ
80     PAR
81     PAR i=0 FOR SIZE in
82       in [ i ] ? CASE
83         tab ; inMessages[i]
84         SKIP
85     PAR i=0 FOR SIZE out
86       out [ i ] ! tab ; outMessages[i]
87 :
```

Deux structures IdentifiedValue et TableIdVal sont définies. La première a deux éléments entiers que sont l'identité (Id) et la valeur de l'identité identifiée. La deuxième structure contient une valeur entière "limit" et un tableau de type IdentifiedValue avec MaxNodes éléments. Le protocole de communication est diam.proto, on peut remarquer que les nœuds peuvent s'échanger un tableau tab de type "TableIdVal", qui contient les identités relevées. De même un message "null" de type "BYTE" et un entier "int"

peuvent être échangés.

Dans la procédure "Circulate", les deux tableaux sont initialisés et une boucle synchrone est réalisée pour permettre la communication entre les nœuds. Les nœuds communiquent via des canaux entrants et sortants définis sous forme de tableaux. Des processus représentant les nœuds lisent et écrivent de manière parallèle sur les canaux.

6.4.4 Algorithme du Bread First Search (BFS)

BFS [158] est un algorithme qui permet de construire un arbre permettant de joindre tous les nœuds en un temps le plus court possible. Les routes sont fixées dans des variables d'état des nœuds intermédiaires.

Etat de noeuds :

```

Variable booléenne marked,
    Vrai pour le noeud root,
    Faux ailleurs.
Variable entière parent : nil (-1) partout
messages initiaux sortants
    send = search pour root
    send = null ailleurs
(Les messages entrants sont stockés dans received[i])

```

Exemple de BFS synchrone en boucle.

Répéter N fois :

```

Communiquer avec les voisins
changement d'état et messages sortants:
    si marked alors send:= null
    si not marked
        si received[i] = search alors
            parent :=i, marked := Vrai, send := search
        sinon send:= null

```

Après N tours, avec ($N = \text{MaxNodes} - 1$, ou $N = \text{diamètre}$), tous les nœuds ont basculé marked à Vrai et tous les nœuds ont un index dans parent. Cet algorithme permet la construction d'un arbre qui permet de faire remonter les informations vers le nœud racine, par exemple.

Dans un réseau connexe, chaque nœud connaît son parent. Cet index peut être utilisé pour des calculs transformant le réseau en arbre remontant. Des calculs descendants peuvent être menés en ajoutant des attributs aux messages search : tels que :

- la somme, le maximum ou le minimum ;
- la distance relative des P_i au nœud root.

Le processus P_i ayant reçu un message search sur un canal parent peut renvoyer le message "beParent" au tour suivant afin de notifier son choix. Un nœud qui reçoit ce message gardera la position de son fils dans une file.

Ceci permet une évaluation en profondeur dans la propagation des distances. Aussi plusieurs algorithmes BFS peuvent tourner en parallèle, permettant d'obtenir les profondeurs des arbres de recouvrement en tout point.

6.4.5 Algorithme de calcul de pente

Cet algorithme permet de déterminer une carte de visibilité du point culminant du système cellulaire. Pour cela un BFS est effectué pour propager les messages et atteindre les autres nœuds.

Tous les nœuds du système cellulaire sont parcourus au bout d'un temps connu au delà duquel, il est inutile de faire tourner l'algorithme. Ce temps est le diamètre du réseau dont la procédure de détermination est présentée plus haut.

Quand un nœud reçoit une route, il calcule sa pente vers le sommet et compare sa pente avec la dernière plus faible pente. A partir de là il sait s'il se trouve en dessous ou au dessus du nœud précédent. S'il est au dessus, il est visible par le point culminant.

La procédure ci-dessous permet de calculer la pente d'un nœud par rapport au point culminant.

Listing 6.4: Code de calcul de pente

```

1  PROC PenteRoute(TraceRoute thisRoute , REAL64 pente , BOOL
      visible )
2  INT limit:
3  CellPosition origin , current:
4  REAL64 xo , yo , zo , xc , yc , zc , deltaPlan , deltaZ , newpente :
5  INT index:
6  SEQ
7  limit := thisRoute[limit]
8  pente := 100000.0
9  visible := TRUE
10 index:=2
11 IF
12     limit >=1
13     SEQ
14         origin:= thisRoute[cellTrace][0]
15         xo := origin[longitude]
16         yo := origin[latitude]
17         zo := origin[elevation]
18         current:= thisRoute[cellTrace][1]
19         xc := current[longitude]
20         yc := current[latitude]
21         zc := current[elevation]

```

```

22         deltaPlan := DSQRT(((xc-xo)*(xc-xo)) + ((yc-
                yo)*(yc-yo)))
23         deltaZ := DABS(zc-zo)
24         pente := deltaZ / deltaPlan
25         WHILE index <= limit
26             SEQ
27                 current := thisRoute[cellTrace][index]
28                 index := index + 1
29                 xc := current[longitude]
30                 yc := current[latitude]
31                 zc := current[elevation]
32                 deltaPlan := DSQRT(((xc-xo)*(xc-xo)) + ((
                        yc-yo)*(yc-yo)))
33                 deltaZ := DABS(zc-zo)
34                 newpente := deltaZ / deltaPlan
35                 IF
36                     newpente <= pente
37                     SEQ
38                         pente := newpente
39                         visible := TRUE
40                     TRUE
41                         visible := FALSE
42                 TRUE
43                 SKIP
44         -- thisRoute[cellTrace][limit] := thisPosition
45     :
```

6.5 Etude de cas : calcul de couverture radio à l'UGB

Elle porte sur l'étude de la couverture radio à l'université Gaston Berger de Saint-Louis. Pour cela, nous avons procédé à un quadrillage de la zone afin de générer nos systèmes cellulaires sur plusieurs résolutions et en utilisant les différentes connectivités. Ceci dans le but d'établir une carte de visibilité sur l'ensemble des points élevés de la zone. Ces points élevés visibles peuvent servir à abriter des émetteurs ou récepteurs radio.

A cette fin, les coordonnées géographiques du point culminant ont été injectées dans le modèle, parce que les élévations de plus de vingt mètres ne sont pas visibles par nos outils de simulation à cause des résolutions disponibles.

Le calcul des pentes, le rapport de $\frac{\Delta Z}{\Delta P}$, avec $\Delta P = \sqrt{X^2 + Y^2}$, permet d'établir des cartes de visibilité des points autour du point culminant.

La méthodologie consiste à établir un ensemble de routes parmi un ensemble de nœuds. Ce calcul est fait avec BFS qui consiste à un balayage du réseau au cours duquel

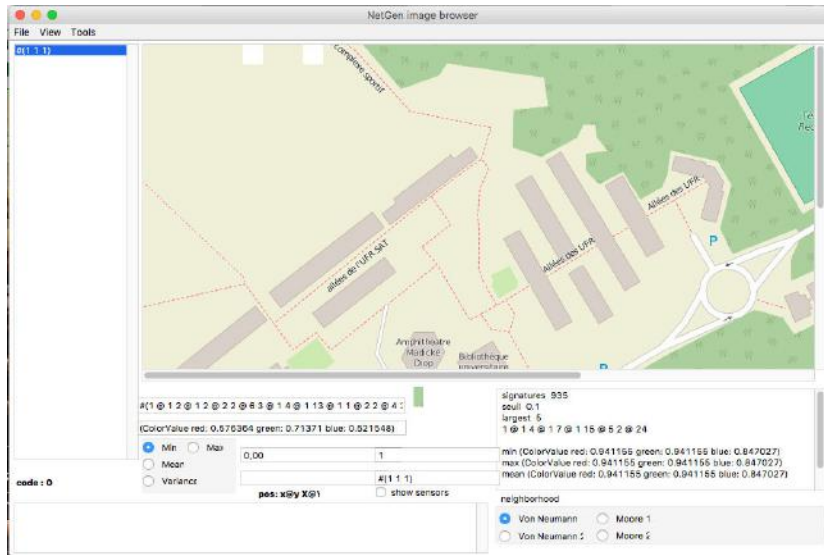


FIGURE 6.4: Une vue graphique de la zone d'étude

chaque nœud calcule ses coordonnées x , y et z et sa pente relativement au point culminant. Si la pente d'un point est supérieure à la pente précédente, cela signifie que le nœud est au dessus. Les nœuds visibles affichent leurs coordonnées et positions géographiques. Nous obtenons ainsi un simulateur qui permet de planifier le déploiement de réseau de capteurs sans fil.

Carte de visibilité du réseau Cette carte montre un ensemble de points visibles (marque rouge) à partir du point culminant. Nous remarquons aussi quelques points égarés.



FIGURE 6.5: Carte de visibilité du réseau

6.5.1 Analyse des résultats

Après avoir injecté les coordonnées géographiques du point culminant et après soumission à GPSVISUALIZER, nous avons obtenu le profil d'élévation qui suit :

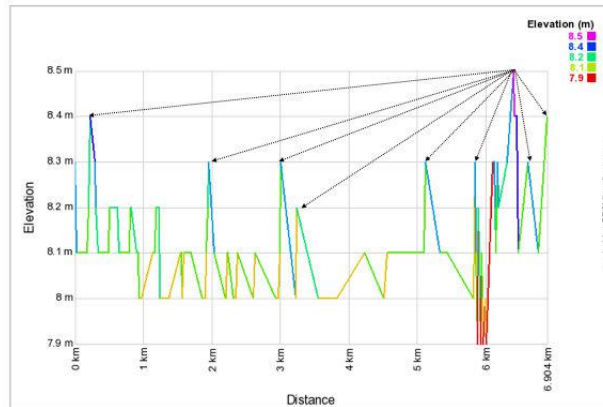


FIGURE 6.6: Profil d'élévation avec Von Neumann 1 comme topologie de voisinage

Le profil montre toutes les variations d'élévation et permet de mettre en exergue les points visibles.

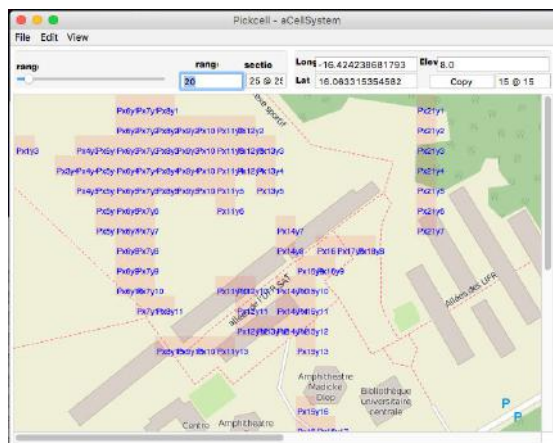


FIGURE 6.7: La carte de couverture du réseau avec un voisinage de Moore 1

On remarque qu'il y a plus de connectivité avec les topologies de Von Neumann et Moore à la distance 2. Entre les deux dernières, celle de Moore 2 présente plus de connectivité, donc on dispose de plus de points visibles. Ceci est dû au nombre de voisins que la topologie de Moore 2 présente. Dans un projet de déploiement de réseaux de capteurs, seuls les points fléchés seront intéressants dans la mise en place d'un noeud.

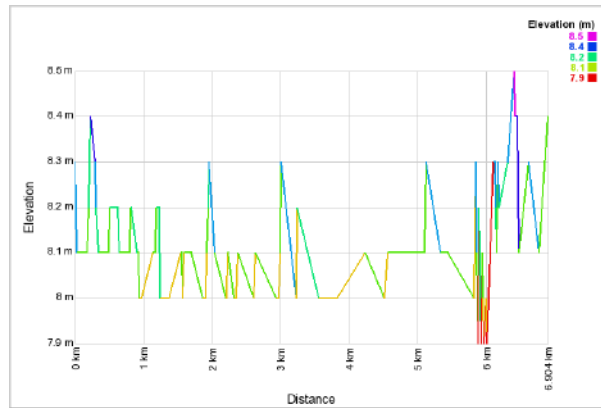


FIGURE 6.8: Profil de connectivité avec le voisinage de Von Neumann 2

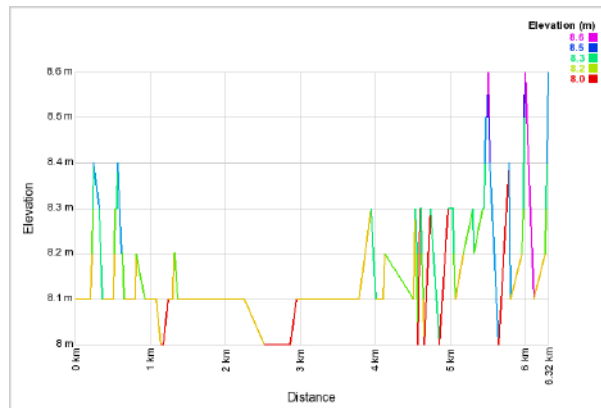


FIGURE 6.9: Profil de connectivité avec le voisinage de Moore2

6.6 Conclusion

Ce chapitre a porté sur le calcul de couverture radio afin de déterminer la carte de visibilité d'une zone donnée. La carte de visibilité représente un ensemble d'émetteurs et de récepteurs en visibilité directe, c'est à dire sans obstacles entre eux. Ceci permet d'éviter les phénomènes de réflexion qui peuvent avoir un impact sur la qualité des signaux transmis.

Nous nous sommes intéressés à l'université Gaston Berger de Saint-Louis qui couvre une superficie de 240 hectares et disposant d'une tour de cinquante mètres de haut sur laquelle, nous envisageons de placer notre émetteur radio.

Pour cela un ensemble d'algorithmes ont été présentés permettant de tracer une route, de calculer le diamètre d'un réseau ou de parcourir un arbre en profondeur. Une route est une série d'étapes avec comme point initial, le point culminant. Le calcul de route est fait par BFS qui consiste à balayer les routes à partir du point culminant et à comparer les valeurs de pente obtenues. Si la pente d'un nœud est supérieure à la pente précédente,

il devient visible et imprime ses coordonnées géométriques.

Un filtre Linux permet d'avoir les points visibles. Le point culminant est connu au départ en faisant un grep. Si R est le point correspondant au point le plus haut, nous pouvons déterminer toutes les routes qui mènent à ce point.

En plaçant la trace des routes dans un fichier et en chargeant le fichier de couverture qui est un sous ensemble de réseaux, le point culminant apparaît en rouge. La couleur représente la dégradation de pixels (distance ()). Chacune des cellules visibles calcule sa distance. Par exemple, le déploiement du système Lora nécessite peu d'émetteurs, donc une planification basée sur le calcul de couvertures peut s'avérer utile pour disposer le réseau.

7

Conclusions et perspectives

Cette thèse a porté sur la modélisation et simulation des phénomènes de migration et du cycle de vie du criquet pèlerin. Ceci a abouti à la production de simulateurs basés sur des automates cellulaires permettant, d'une part, de retracer la dynamique de population de ces insectes ravageurs et d'autre part d'établir une carte de visibilité d'un ensemble d'émetteurs et de récepteurs déployés dans une zone d'intérêt. Le premier permet de faire des prédictions sur l'évolution des populations en local dans une cellule et entre cellules. En fonction des régimes de vent et des paramètres climatiques, une estimation du sens de déplacement des essaims de criquets est faite. Des prévisions d'éventuelles infiltrations provenant des cellules voisines sont aussi établies. Le second permet d'effectuer des calculs de couverture radio dans une région donnée. Ces calculs peuvent guider pour une bonne planification du déploiement de réseaux de capteurs sans fil.

Les insectes ravageurs dans leur milieu de reproduction et les paramètres climatiques sont considérés comme un phénomène biophysique sur lequel peut rétro-agir un réseau de surveillance synchrone basé sur les réseaux de capteurs sans fil. Un phénomène physique est caractérisé par la complexité des composants et des échanges pouvant exister entre-eux.

Nous avons procédé à un découpage spatial de cet ensemble en utilisant des résolutions différentes pour les cellules afin d'étudier les mécanismes de migrations de population entre les cellules.

Le découpage temporel a été fait en calant les étapes synchrones aux différentes durées de vie des différents stades de l'évolution des criquets. Ce découpage a permis de déterminer la densité de population des individus aux différentes phases de développement.

Les simulateurs qui ont été proposés jusqu'ici, n'intégraient pas directement les données relatives au vent, à la température ou à la pluviométrie directement dans les

cellules. Ils proposaient plutôt des modèles parallèles à simuler conjointement.

Les modèles cellulaires de migration et de développement local développés dans ce travail, intègrent directement les données climatiques dans les cellules à l'aide de paramètres collectés dans des fichiers au format Grib fournis par les organismes météo.

Les applications pratiques qu'on peut en faire sont multiples et variées, par exemple simuler le cycle biologique du criquet pèlerin, ou le processus de migrations saisonnières des insectes. Aussi planifier le déploiement de réseaux de capteurs dans une zone caractérisée par la présence d'obstacles localisables à l'aide de leurs coordonnées géographiques.

Les contributions scientifiques de cette thèse se trouvent dans l'intégration de paramètres météorologiques dans les cellules. Ceci permet de réduire le nombre de sous-modèles qui composent les phénomènes physiques.

Ces modèles restent à valider en utilisant des données recueillies sur le terrain par les centres nationaux de lutte anti-acridienne. Il s'agira de comparer les résultats des simulations obtenus aux données d'observations collectées par les équipes de prospection. Les comparaisons avec des modèles théoriques ont été faites, des similitudes ont été trouvées dans les résultats mais les approches de modélisation sont différentes.



Annexe A

A.1 Extrait de fichier Occam du réseau cellulaire

```
-- processus : 5053
-- min fanout : 2
-- max fanout : 8
-- channels   : 38504

--generated at 22 septembre 2016 16:12:34
--generic synchronous network model
-- include data file for physical simulation

#INCLUDE    "/Users/traore/VisualWorks Projects/Generated/sl1/sl1Data0.occ"

VAL [5053][7]BYTE NetProcess IS [    "Px10y1 ", -- id: 1
    "Px10y10", -- id: 2
    "Px10y11", -- id: 3
    "Px10y12", -- id: 4
    "Px10y13", -- id: 5
    -----
    "Px20y63", -- id: 796
    "Px20y64", -- id: 797
    -----
    "Px9y7  ", -- id: 5051
    "Px9y8  ", -- id: 5052
    "Px9y9  "]:
```

```

VAL [5053][8]BYTE NetProcedure IS [ "CellNode", -- id: 1
  "CellNode", -- id: 2
  "CellNode", -- id: 3
  "CellNode", -- id: 4
  "CellNode", -- id: 5
  -----
  "CellNode", -- id: 796
  "CellNode", -- id: 797
  -----
  "CellNode", -- id: 5051
  "CellNode", -- id: 5052
  "CellNode"]:

DATA TYPE Location
RECORD
  INT xLoc:
  INT yLoc:
  INT elevation:
  INT range:
:
VAL [5053] Location NetLocation IS [ [10 , 1 , 0 , 20 ] , -- id: 1
  [10 , 10 , 0 , 20 ] , -- id: 2
  [10 , 11 , 0 , 20 ] , -- id: 3
  [10 , 12 , 0 , 20 ] , -- id: 4
  [10 , 13 , 0 , 20 ] , -- id: 5
  -----
  [20 , 63 , 0 , 20 ] , -- id: 796
  [20 , 64 , 0 , 20 ] , -- id: 797
  -----
  [9 , 7 , 0 , 20 ] , -- id: 5051
  [9 , 8 , 0 , 20 ] , -- id: 5052
  [9 , 9 , 0 , 20 ] ]:

#USE "course.lib"

VAL INT MaxFanOut IS 8:

VAL INT MaxNodes IS 5053:

#INCLUDE "nodes-test-include.occ"

PROC sl10(CHAN OF BYTE stdin, stdout, stderr)

```

```

--- Channel declarations
CHAN OF diam.proto Px59y20.Px59y19,Px59y20.Px59y21,Px59y20.Px60y21,...:
CHAN OF diam.proto Px28y47.Px29y47,Px28y47.Px27y48,Px28y47.Px29y48,...:
CHAN OF diam.proto Px60y39.Px60y38,Px60y39.Px59y39,Px60y39.Px61y38,...:
-----
CHAN OF diam.proto Px27y9.Px27y10,Px27y9.Px26y8,Px27y9.Px27y8,...:
CHAN OF diam.proto Px57y58.Px57y59,Px57y58.Px56y58,Px57y58.Px57y57,...:
CHAN OF diam.proto Px71y10.Px72y10,Px71y10.Px70y11,Px71y10.Px72y11,...:

--- Channel table declaration for nodes
Px59y20.out IS [ Px59y20.Px59y19,Px59y20.Px59y21,Px59y20.Px60y21,... ]:
Px59y20.in IS [ Px60y21.Px59y20,Px58y21.Px59y20,Px58y19.Px59y20,... ]:
-----
Px71y10.out IS [ Px71y10.Px72y10,Px71y10.Px70y11,Px71y10.Px72y11... ]:
Px71y10.in IS [ Px72y11.Px71y10,Px70y9.Px71y10,Px72y10.Px71y10,... ]:

-- Program Body

[MaxNodes]CHAN OF BYTE toMux:
PAR
  CellNode(Px10y1.in, Px10y1.out,0, toMux [0])
  CellNode(Px10y10.in, Px10y10.out,1, toMux [1])
  CellNode(Px10y11.in, Px10y11.out,2, toMux [2])
  CellNode(Px10y12.in, Px10y12.out,3, toMux [3])
  CellNode(Px10y13.in, Px10y13.out,4, toMux [4])
  CellNode(Px10y14.in, Px10y14.out,5, toMux [5])
  -----
  CellNode(Px20y64.in, Px20y64.out,796, toMux [796])
  CellNode(Px20y65.in, Px20y65.out,797, toMux [797])
  -----
  CellNode(Px9y8.in, Px9y8.out,5051, toMux [5051])
  CellNode(Px9y9.in, Px9y9.out,5052, toMux [5052])
Mux(toMux,stdout)
  -- End of program body
:

```


B

Annexe B

B.1 Extrait du fichier Occam des données cellulaires

```
DATA TYPE ImageExtent
```

```
RECORD
```

```
  INT width:
```

```
  INT height:
```

```
:
```

```
DATA TYPE CellPosition
```

```
RECORD
```

```
  INT identity: -- self reference
```

```
  INT x,y: -- cell position
```

```
  REAL64 longitude,latitude,elevation: -- geo loc
```

```
:
```

```
DATA TYPE Wind
```

```
RECORD
```

```
  REAL32 ugrd: --composante u du vent
```

```
  REAL32 vgrd: --composante v du vent
```

```
:
```

```
DATA TYPE Weather
```

```
RECORD
```

```
  INT temperature:
```

```
  INT pluie:
```



```

    INT humidity:
:
DATA TYPE RGBPixel
  RECORD
    BYTE red, green, blue:
:
DATA TYPE Depth24ByteArray IS [ 100] RGBPixel:
DATA TYPE CellImage
  RECORD
    ImageExtent extent:
    Depth24ByteArray pixelArray:
:
DATA TYPE CellArray
  RECORD
    CellPosition position:
    Wind windArray:
    CellImage image:
:
VAL [ 5053] CellArray Cells IS [ [ [ 1,10,1, -15.989227294922,18.483516449358, -0.400
  [ [ 2, 10, 10, -15.989227294922, 18.424896202842, 0.40000000596046], [ 0.0, 0.0] , [
  [ [ 3, 10, 11, -15.989227294922, 18.418381607361, 5.3000001907349], [ 0.0, 0.0] , [
  [ [ 4, 10, 12, -15.989227294922, 18.411866765203, -1.2000000476837], [ 0.0, 0.0] , [
  [ [ 5, 10, 13, -15.989227294922, 18.405351676442, 3.2000000476837], [ 0.0, 0.0] , [
-----
  [ [ 5053, 9, 9, -15.99609375, 18.431410551572, 5.3000001907349], [ 0.0, 0.0] , [ [

```

C

Annexe C

C.1 Algorithme de regroupement de cellules en classes

L' algorithme suivant est appliqué à l'ensemble du réseau cellulaire, pour obtenir un ensemble de minimum, maximum et moyennes de chaque couleur.

Etape 1

```
Pour chaque cellule de la grille
  Pour chaque pixel dans la cellule
    Pour chaque composante de couleur dans {R,V,B}
      calcul (minimum (couleur))
      calcul (maximum (couleur))
      calcul (somme(couleur))
    Pour chaque composante de couleur dans {R,V,B}
      calcul (moyenne(couleur))
```

Après cette étape, nous obtenons 3×3 paramètres définis dans chaque cellule, permettant ainsi de calculer les caractéristiques globales de l'image.

Etape 2

```
Pour chaque cellule de la grille
  Pour chaque composante de couleur dans {R,V,B}
    Pour chaque composante de couleur dans {R,V,B}
      calcul (minGlobal)
      calcul (maxGlobal)
      calcul (minMoyenneGlobal)
      calcul (maxMoyenneGlobal)
```

Après cette étape et pour chaque composante dans l'espace RVB, nous obtenons $[min, max]$ mesures pour

- la valeur prise sur la vue graphique ;
- la moyenne des valeurs dans chaque cellule.

C.2 Algorithme de regroupement de cellules par affinité

Nous pouvons utiliser les valeurs calculées globalement dans chaque cellule pour regrouper les cellules par affinité. Pour le minimum, le maximum, et la moyenne dans chaque cellule, l'algorithme est le suivant :

Etape 3

decide sur une partition dans $N > 1$

pour chaque composante de couleur dans (R, V, B)

supprimer toute valeur trouvée en dehors de l'intervalle $[min, max]$ pour chaque c

produire N sous-intervalles adjacents $[min, max]$

Dans le cas de $N = 2$, les valeurs minimum, maximum et moyennes RVB dans chaque cellule permettront de classer chaque cellule d'une manière unique dans un espace de dimension 2^3 , $(R_0, R_1) \times (V_0, V_1) \times (B_0, B_1)$. Le cube de coordonnées va de $0, 0, 0 = 0$ à $1, 1, 1 = 7$.

La sélection de N permet d'assigner à chaque cellule un intervalle de valeurs. En choisissant des codes tels que définis ci-dessus, les régions correspondantes apparaissent comme une partie de l'image originale.

Etape 4

pour chaque interval allouer une collection pour enregistrer des cellules rela
pour chaque cellule de la grille enregistre la cellule dans son intervalle ave

D

Annexe D

D.1 Exemple de fichiers GRIB

```
MsgNum, Byte, GRIB-Version, elem, level, reference(UTC), valid(UTC), Proj(hr)
1.0, 0, 1, RH="Relative humidity [%]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 06:00
2.0, 1177, 1, RH="Relative humidity [%]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 09:00
3.0, 2354, 1, RH="Relative humidity [%]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 12:00
4.0, 3531, 1, RH="Relative humidity [%]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 15:00
5.0, 4708, 1, RH="Relative humidity [%]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 18:00
6.0, 5885, 1, RH="Relative humidity [%]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 21:00
7.0, 7062, 1, RH="Relative humidity [%]", 2-HTGL, 03/22/2017 06:00, 03/23/2017 00:00
8.0, 8239, 1, RH="Relative humidity [%]", 2-HTGL, 03/22/2017 06:00, 03/23/2017 03:00
9.0, 9416, 1, RH="Relative humidity [%]", 2-HTGL, 03/22/2017 06:00, 03/23/2017 06:00

66.0, 76505, 1, TMAX="Maximum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 06:00
67.0, 78120, 1, TMAX="Maximum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 09:00
68.0, 79297, 1, TMAX="Maximum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 12:00
69.0, 80584, 1, TMAX="Maximum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 15:00
70.0, 81871, 1, TMAX="Maximum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 18:00
71.0, 83048, 1, TMAX="Maximum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/23/2017 00:00
72.0, 84225, 1, TMAX="Maximum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/23/2017 03:00
73.0, 85840, 1, TMAX="Maximum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/23/2017 06:00

130.0, 158867, 1, TMIN="Minimum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 06:00
131.0, 160482, 1, TMIN="Minimum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 09:00
132.0, 162097, 1, TMIN="Minimum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 12:00
```

133.0, 163384, 1, TMIN="Minimum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017
134.0, 164999, 1, TMIN="Minimum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017
135.0, 166614, 1, TMIN="Minimum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/23/2017
136.0, 168229, 1, TMIN="Minimum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/23/2017
137.0, 169844, 1, TMIN="Minimum temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/23/2017

194.0, 245161, 1, APCP="Total precipitation [kg/m^2]", 0-SFC, 03/22/2017 06:00, 03/22/
195.0, 246666, 1, APCP="Total precipitation [kg/m^2]", 0-SFC, 03/22/2017 06:00, 03/22/
196.0, 247843, 1, APCP="Total precipitation [kg/m^2]", 0-SFC, 03/22/2017 06:00, 03/22/
197.0, 249348, 1, APCP="Total precipitation [kg/m^2]", 0-SFC, 03/22/2017 06:00, 03/22/
198.0, 250635, 1, APCP="Total precipitation [kg/m^2]", 0-SFC, 03/22/2017 06:00, 03/22/
199.0, 252250, 1, APCP="Total precipitation [kg/m^2]", 0-SFC, 03/22/2017 06:00, 03/23/
200.0, 253537, 1, APCP="Total precipitation [kg/m^2]", 0-SFC, 03/22/2017 06:00, 03/23/
201.0, 255152, 1, APCP="Total precipitation [kg/m^2]", 0-SFC, 03/22/2017 06:00, 03/23/

258.0, 336143, 1, TMP="Temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 06:00, 0
259.0, 337758, 1, TMP="Temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 09:00, 3
260.0, 339373, 1, TMP="Temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 12:00, 6
261.0, 340660, 1, TMP="Temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 15:00, 9
262.0, 341947, 1, TMP="Temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 18:00, 1
263.0, 343124, 1, TMP="Temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/22/2017 21:00, 1
264.0, 344301, 1, TMP="Temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/23/2017 00:00, 1
265.0, 345916, 1, TMP="Temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/23/2017 03:00, 2
266.0, 347531, 1, TMP="Temperature [K]", 2-HTGL, 03/22/2017 06:00, 03/23/2017 06:00, 2



Annexe E

E.1 Taxonomie des architectures

Une architecture distribuée permet de répartir des ressources informatiques sur plusieurs calculateurs pouvant se trouver à des endroits différents. Internet est un exemple de réseau basé sur cette architecture car aucun ordinateur du réseau ne centralise toutes les ressources disponibles.

À ce concept s'oppose celui d'architecture centralisée dont une machine, disposant d'une importante puissance de calcul, concentre toutes les ressources. Cette architecture est basée sur la possibilité d'exécuter des programmes sur des machines réparties sur le réseau et qui communiquent par messages.

Une architecture parallèle permet de traiter les données et d'exécuter les instructions de manière concurrente en des temps raisonnables. Elle est utilisée aujourd'hui dans de nombreux domaines parmi lesquels on peut citer :

- la dynamique des fluides ;
- la modélisation et la simulation de phénomènes physiques complexes ;
- le traitement de signal ;
- la météorologie,
- etc...

À côté, on peut lui opposer l'architecture séquentielle de Von Neumann où le traitement et les exécutions sont faits en série. L'architecture de von Neumann utilise une structure de mémoire unique pour le stockage des instructions et données. Elle est composée de quatre parties distinctes que sont :

- l'unité de calcul arithmétique et logique (UAL) pour effectuer les opérations de base ;
- l'unité de contrôle (UC) pour la coordination des opérations ;

- la mémoire qui contient à la fois les données et le programme,
- les périphériques d'entrée-sortie.

Il existe d'autres architectures comme celle de Harvard qui dissocie la mémoire de données de celle des instructions. L'accès à chacune de ses mémoires se fait avec des bus distincts.

Les modèles cellulaires développés dans cette thèse sont constitués de milliers de processus que seul un traitement parallèle permet de faire garantir des temps d'exécution acceptables. En effet le phénomène de migration des criquets pèlerin se fait sur de grandes échelles et le découpage en cellule de la zone de migration utilise un nombre important de cellules.

E.2 Les différentes architectures des calculateurs

Un ordinateur traite les données stockées dans sa mémoire via son CPU. Le processeur dispose de registres qui sont ses propres unités de stockage plus rapides que la mémoire principale, mais moins nombreuses à cause de leur coût plus élevé.

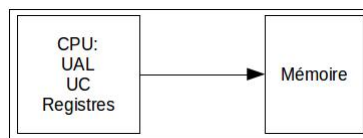


FIGURE E.1: Schéma fonctionnel simplifié d'un ordinateur

E.2.1 Les architectures parallèles

Le besoin de plus en plus important de disposer de plus de puissance de calcul pour les simulations a impulsé la recherche sur le développement d'architectures parallèles. L'avènement des semi-conducteurs et la technologie d'intégration à très grande échelle, VLSI (Very Large Scale Integration) ont contribué à améliorer les performances des calculateurs à processeur unique en terme de puissance de calcul pendant une bonne période, mais ils n'ont pas suffi à faire face aux nouveaux défis de simulations simultanées intégrant des milliers de processus.

Ces limites ont motivé le développement d'architectures parallèles pouvant traiter des volumes importants de données en un délai raisonnable. Ces architectures s'appuient sur l'association de plusieurs processeurs sur un ordinateur ou répartis entre différents ordinateurs dans le but de travailler sur un même problème de façon simultanée, ainsi la barrière séquentielle imposée par l'architecture de von Neumann est surmontée.

En plus de surmonter la barrière séquentielle, les architectures parallèles présentent plus de fiabilité et de disponibilité que les systèmes mono-processeur. Par exemple les systèmes parallèles peuvent résister à la panne d'un des processeurs, avec une certaine perte de capacité, alors qu'un système mono-processeur est placé hors service.

En plus de l'augmentation du nombre de processeurs interconnectés, l'utilisation de micro-processeurs performants disposant de canaux de communication rapides permet d'améliorer la vitesse de calcul de ces machines appelées supercalculateurs.

D'autres types de supercalculateurs existent comme ceux utilisant des technologies de pipeline. Cette technologie consiste à découper une tâche en plusieurs de manière à accélérer le traitement. Cependant, elle nécessite des composants très rapides, des machines coûteuses et un long temps de conception.

Il existe plusieurs architectures parallèles, nous décrivons les deux les plus couramment utilisées, les systèmes multi-processeurs et multi-ordinateurs.

Les systèmes multi-processeurs

Les systèmes multi-processeur reposent sur un système de mémoire composé d'un ou plusieurs modules. L'espace d'adressage de ce système de mémoire est partagé entre les processeurs. Les systèmes multi-processeur peuvent être classés en deux catégories selon l'organisation du système de mémoire qui peut être fortement ou faiblement couplée. Dans le premier cas, tous les processeurs disposent du même temps d'accès à la mémoire centrale. Cette architecture peut être implémentée sous la forme d'une grande unité ou d'un ensemble d'unités mémoire accessibles en parallèle par le biais de processeurs différents. Cette dernière implémentation comporte des avantages car elle permet de réduire le nombre de collisions en accès mémoire et rend ainsi le système plus efficace. La collision est relative à des situations où plusieurs processeurs demandent un accès à la mémoire sur une durée très courte, ceci a des conséquences sur les délais d'accès à la mémoire.

Dans le second cas, le système de mémoire est partitionné entre les différents processeurs. C'est-à-dire qu'une mémoire locale est attachée à chaque processeur. Ainsi, chacun d'entre eux peut accéder directement à sa mémoire locale et à celle des autres. Cependant, le temps d'accès à une mémoire locale est beaucoup plus faible que celui d'une mémoire distante. Cette organisation permet aussi de réduire considérablement le nombre de collisions.

Les systèmes multi-ordinateurs

Les systèmes multi-ordinateur ont une mémoire locale associée à chaque processeur, en plus d'une unité de traitement individuelle et des périphériques d'entrée/sortie. Par conséquent, chaque processeur peut agir de manière autonome en utilisant ses données locales. Les processeurs peuvent accéder directement à leur mémoire locale et non aux mémoires distantes. Pour un accès à une mémoire distante, un mécanisme de transmission de messages permet de réaliser cette tâche. Ce mécanisme est basé sur l'envoi ou la réception de bloc d'informations vers ou depuis les autres processeurs via des canaux de communication. Ces derniers sont des connexions physiques entre processeurs et sont organisés suivant une topologie de réseau donnée.

Chaque processeur est relié à un canal de communication via une interface. Ce dernier effectue des transactions pour s'assurer de la bonne transmission des données.

Un bloc d'informations est encapsulé dans un message avec au début un champ d'en-tête et à la fin un champ de contrôle. Le champ d'en-tête permet d'identifier les nœuds source et destination et contient la longueur du message et le champ de contrôle permet la détection d'erreurs de transmission. L'interface de communication, dans certaines implémentations, est capable de créer et de décoder ces champs d'en-tête et de contrôle.

L'architecture parallèle repose sur deux piliers, le premier consiste à juxtaposer plusieurs processeurs séquentiels et le second à exécuter de manière concurrente plusieurs instructions indépendantes. Sur la figure de droite ci-dessous (voir Figure E.2) un programme est subdivisé en quatre parties à exécuter parallèlement. Chaque partie est subdivisée en un ensemble d'instructions. Les instructions sont exécutées simultanément sur différents processeurs. Un mécanisme de coordination des activités est nécessaire dans ce cas. La figure à gauche est une représentation du traitement séquentiel.

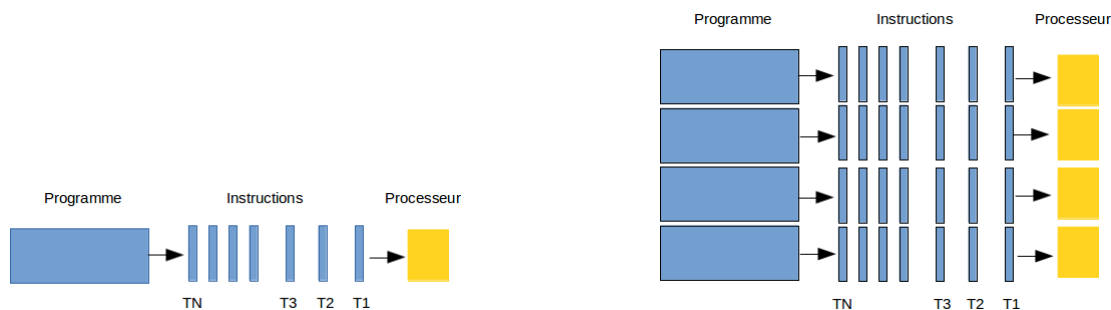


FIGURE E.2: Traitement séquentiel et parallèle

E.2.2 Les modèles architecturaux classiques

La taxonomie de Flynn [90] est le plus utilisée dans la classification des modèles architecturaux. Cette classification, basée sur la manipulation de flux de données et d'instructions, a abouti à quatre architectures principales :

- SISD (Single Instruction, Single Data) ;
- SIMD (Single Instruction, Multiple Data) ;
- MISD (Multiple Instruction, Single Data) ;
- MIMD (Multiple Instructions, Multiple Data).

Le modèle SISD est le plus simple. Il permet le traitement de données comme dans les systèmes séquentiels. Il est retrouvé dans les systèmes classiques mono-processeur pouvant accueillir un flux d'instructions exécuté en série. Aujourd'hui, presque tous les ordinateurs sont multi-processeur avec des flux d'instructions indépendants exécutés par chacun d'entre eux. Par conséquent, de tels systèmes doivent être considérés comme un ensemble de machines SISD agissant sur des espaces de données différents.

Le modèle SIMD est plus orienté dans le traitement uniforme de grands volumes de données. L'exemple des processeurs vectoriels avec des unités de calcul traitant des images et sons peut être donné. Les processeurs vectoriels agissent sur des tableaux de

données similaires plutôt que sur des données simples en utilisant des CPU spécialement structurés pour cela.

Le modèle MISD a été rarement utilisé, il semble néanmoins adapté à certains problèmes comme la simulation des réseaux de neurones.

Le modèle MIMD est adapté aux systèmes multi-processeurs ou mono-processeur multi-cœur.

Cette classification met en exergue deux types de parallélismes : le parallélisme par flot d'instructions également connu sous le nom parallélisme de traitement, où plusieurs instructions différentes sont exécutées simultanément, qui correspond au MIMD et le parallélisme de données, où les mêmes opérations sont répétées sur des données différentes, le SIMD.

Selon le modèle d'accès à la mémoire, deux types d'architectures à mémoires partagées et distribuées peuvent être distinguées dans la classe MIMD.

Les systèmes à mémoire partagée Les systèmes à mémoire partagée disposent de plusieurs CPU. Les CPU partagent un accès équitable à une mémoire commune, généralement via un bus. Par conséquent l'utilisateur n'a pas à s'occuper du lieu de stockage des données.

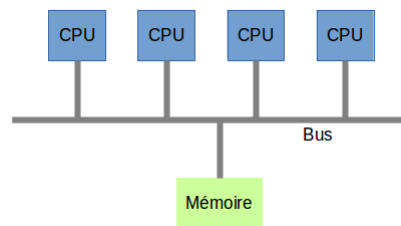


FIGURE E.3: Architecture à mémoire partagée

Le passage à l'échelle dans cette architecture introduit habituellement une certaine forme de hiérarchie de mémoire. En particulier, la fréquence à laquelle la mémoire partagée est accédée, peut être réduite en stockant des copies d'éléments de données fréquemment utilisés dans une mémoire cache associée à chaque processeur. L'accès à ce cache est beaucoup plus rapide que l'accès à la mémoire partagée. Par conséquent, la localité est généralement importante, et les différences entre les multi-ordinateurs et multi-processeurs sont juste des questions de degré. Les programmes développés pour les systèmes multi-ordinateurs peuvent s'exécuter efficacement sur les systèmes multi-processeurs, car la mémoire partagée permet un passage efficace des messages.

Les systèmes à mémoire distribuée Dans cette architecture chaque ordinateur est autonome. Les ordinateurs communiquent par envoi de message (messages passing)

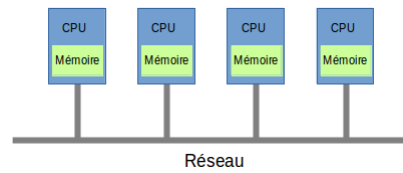


FIGURE E.4: Architecture à mémoire distribuée

Modèles de programmation parallèle

Ce modèle de machines parallèle comprend un certain nombre d'ordinateurs de von Neumann interconnectés en réseau. Chaque ordinateur exécute son propre programme qui peut accéder à la mémoire locale et peut envoyer ou recevoir des messages sur le réseau. Les messages sont utilisés pour communiquer avec d'autres ordinateurs pour lire et écrire sur des mémoires distantes de manière équivalente. Idéalement, le coût d'envoi d'un message entre deux nœuds est indépendant de leur emplacement et du trafic réseau, mais dépend de la longueur du message. Ce modèle se rapproche plus du modèle MIMD à mémoire distribuée avec comme différence la dépendance du coût d'envoi d'un message de l'emplacement des nœuds et du trafic réseau.

L'architecture de von Neumann utilise des processeurs capables d'exécuter des séquences d'instructions. Les instructions sont des opérations arithmétiques et logiques, des adresses de lecture et d'écriture d'une donnée en mémoire et l'adresse de la prochaine instruction à exécuter. Bien qu'il soit possible de programmer en fonction de ce modèle en écrivant le langage machine, cette méthode est pour la plupart très complexe car l'historique de millions de cases mémoires doit être gardée et l'exécution de milliers d'instructions doit être structurée.

Par conséquent, des techniques de conception modulaire sont utilisées où des programmes complexes sont construits à partir de composants simples structurés en termes d'abstractions haut niveau telles que les structures de données, les boucles itératives et les procédures. Par exemple les dernières rendent la conception modulaire plus facile en permettant la manipulation d'objets sans se soucier de leur structure interne. Ainsi, les langages de programmation haut niveau tels que Pascal, Occam, C et Ada faciliteront la transcription de ces abstractions en code exécutable.

Acronymes et définitions

- API** - Application Programming Interface
AllEnvi - Alliance nationale de la recherche pour l'Environnement
ASTER - Advanced Spaceborne Thermal Emission and Reflection Radiometer
BBEES - Bases de données biodiversité, écologie, environnements et sociétés
BFS - Breadth First Search
CAN - Convertisseur Analogique Numérique
CIRAD - Centre de coopération Internationale en Recherche Agronomique pour le développement
CLCPRO - Commission de Lutte contre le Criquet Pèlerin dans la Région Occidentale
CNA - Convertisseur Numérique Analogique
CSV - Comma-Separated Values
CPS - Cyber-Physical Systems
CPU - Central Processing Unit
CUDA - Compute Unified Device Architecture
CRC - Commission de lutte contre le criquet pèlerin dans la Région Centrale
DDL - Data Definition Language
DEM - Digital Elevation Model
DLCO-EA - Organisation de lutte contre le criquet pèlerin en Afrique de l'Est
DLIS - Desert Locust Information Service
EXDI - External Database Interface
FAO - Food and Agriculture Organization of the United Nations
FOM - Federation Object Model
GCM - General Circulation Model
GDEM - Global Digital Elevation Model
GeoSciML - GeoScience Markup Language
Glorp - Generalized Lightweight Object-Relational Persistence
GPS - Global Positioning System
GPU - Graphical Process Unit
GRIB - Gridded Binary Data
HLA - High Level Architecture
IAGMV - Initiative Africaine de la Grande Muraille Verte
IEEE - Institute of Electrical and Electronics Engineers
InEE - Institut Ecologie et Environnement

- IR** - Infrastructures de Recherche nationales
IRD - Institut pour la Recherche et le Développement
IRI - International Research Institute
I2C - Inter-Integrated Circuit
JTAG - Joint Test Action Group
LabSTICC - Laboratoire en sciences et techniques de l'information, de la communication et de la connaissance
MEMS - MicroelectroMechanical Systems
MIMD - Multiple Instructions, Multiple Data
MISD - Multiple Instruction, Single Data
MNHN - Muséum National d'Histoire Naturelle
MVC - Model View Controller
NetCDF - NetworkCommonDataForm
NOAA - National Oceanic and Atmospheric Administration
OCLALAV - Organisation commune de lutte anti-acridienne et de lutte anti-aviaire pour les pays d'Afrique de l'Ouest au sud du Sahara
OMT - Object Model Template
PC - Personal Computer
PSI - Public Sector Information
PWM - Pulse Width Modulation
RTI - Run Time Environment
SAMES - Stic Asia Modeling for Environment and Simulation
SDMX - Statistical Data and Metadata eXchange
SIMD - Single Instruction, Multiple Data
SISD - Single Instruction, Single Data
SOERE - Système d'Observation et d'Expérimentation pour la Recherche en Environnement
SRTM - Shuttle Radar Topography Mission
SPI - Serial Peripheral Interface
SWAC - Commission de lutte contre le criquet pèlerin en Asie du Sud-Ouest
TSV - Tab-Separated Values
UAL - Unité Arithmétique et Logique
UART - Universal Asynchronous Receiver-Transmitter
UBO - Université de Bretagne Occidentale
UGB - Université Gaston Berger

Liste des publications

1. O. N. P. Goubier, H. X. Huynh, T. P. Truong, M. Traore. Wireless Sensor Network-based Monitoring, Cellular Modelling and Simulations for the Environment. *ASM Sc. J.*, Special Issue 2017(1) ICT-Bio, 56 - 45
2. M. Traore, L. Escalade, O. Thiare and B. Pottier. "Life dynamics cellular modelling and simulation", In Proceedings of the 6th IASTED International Conference on Modelling, Simulation and Identification (MSI 2016), pp. 134-140, August 2016.
3. M. Traore, T. V. Hoang, O. Thiare and B. Pottier. "A novel design flow for physical systems simulation with sensor networks : a case study on insect's distribution", In Proceedings of CNRIA 2015, Thies Senegal
4. Lam, B.H., Huynh, H.X., Traoré, M., Lucas, P.Y., Pottier, B. : Monitoring environmental factors in Mekong Delta of Vietnam using Wireless Sensor Network approach, In : 8th International conference on Simulation and Modelling in the Food and Bio-Industry FoodSim 2014, pp. 71-78. In Book "Nature of Computation and Communication (Springer)", ISBN 978-90-77381-84- 7 (2014)
5. M. Traore, B. Pottier and O. Thiare. "Introduction à la Programmation des capteurs sans fil", In Proceedings of CNRIA 2013, Ziguinchor Senegal, pp. 144-146, April 2013.
6. Pierre-Yves Lucas, Eloi Keita, Ahcene Bounceur, Mahamadou Traore, and Bernard Pottier. A Test Case for a Wireless Sensor Network Design Flow, In 12th Embed With Linux Workshop (EWILI'12), 7 June 2012.

Bibliographie

- [1] Bbees-ums 3468-bases de données biodiversité, écologie, environnement et sociétés.
- [2] Site web de la fondation géospatiale open source.
- [3] World databank.
- [4] Données environnementales états unis, Mai 2009.
- [5] Données environnementales nouvelle zélande, Novembre 2009.
- [6] Données environnementales royaume uni, Janvier 2010.
- [7] Données environnementales australiennes, Mars 2011.
- [8] Ecoscope, 2015.
- [9] HE Abdel Rahman. *Studies on the effect of different food plant species on the life system of the desert locust, Schistocerca gregaria Forskal (Orthoptera : Acrididae)*. PhD thesis, PhD thesis. University of Khartoum, Sudan, 1999.
- [10] DIA Abdoulaye and Docteur Robin DUPONNOIS. de la grande muraille verte.
- [11] AHMED Ahmed. Simulation et modélisation de diffusions physiques rapport de ter. 2013.
- [12] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks : a survey. *Computer networks*, 38(4) :393–422, 2002.
- [13] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensors networks. *IEEE Communications Magazine*, 40(8) :102–114, 2002.
- [14] Ahmad T Al-Hammouri, Michael S Branicky, and Vincenzo Liberatore. Co-simulation tools for networked control systems. In *Hybrid Systems : Computation and Control*, pages 16–29. Springer, 2008.
- [15] Fillou Alain. Home automation. *Techniques de l'ingénieur Construction*, CD3 N°C3781 :1–17, 1997.
- [16] Cesare Alippi, Romolo Camplani, Cristian Galperti, and Manuel Roveri. A robust, adaptive, solar-powered wsn framework for aquatic environmental monitoring. *Sensors Journal, IEEE*, 11(1) :45–55, 2011.
- [17] Frédéric Amblard, Juliette Rouchier, and Pierre Bommel. Evaluation et validation de modèles multi-agents. *Modélisation et simulation multi-agents. Applications pour les Sciences de l'Homme et de la Société*, pages 103–140, 2006.

- [18] Cécile Appert and Ludger Santen. Modélisation du trafic routier par des automates cellulaires. *Actes INRETS*, Actes INRETS 100, 2002.
- [19] Florent Arrignon. *HOVER-WINTER : un modèle multi-agent pour simuler la dynamique hivernale d'un insecte auxiliaire des cultures (Episyrphus balteatus, Diptera : Syrphidae) dans un paysage hétérogène*. PhD thesis, 2006.
- [20] Florent Arrignon, Marc Deconchat, Jean-Pierre Sarthou, Gérard Balent, and Claude Monteil. Modelling the overwintering strategy of a beneficial insect in a heterogeneous landscape using a multi-agent system. *Ecological Modelling*, 205(3) :423–436, 2007.
- [21] C Ashall, Peggy E Ellis, et al. Studies on numbers and mortality in field populations of the desert locust (*Schistocerca gregaria* forskål). *Anti-Locust Bull.*, (38), 1962.
- [22] J. Aurelien and B. Pottier. Les réseaux de capteurs sans fil au service des applications agri-environnementales. In *Séminaire Réseaux de capteurs sans fil, LabSTICC, Faculté de sciences, Brest*, September 2010.
- [23] A Ayali, MP Pener, and J Girardie. Comparative study of neuropeptides from the corpora cardiaca of solitary and gregarious locusta. *Archives of insect biochemistry and physiology*, 31(4) :439–450, 1996.
- [24] Nicolas Bacaër. *Histoires de mathématiques et de populations*. Cassini, 2008.
- [25] Jacques Bair and Jean Mawhin. Tangente hors-série. num. 40. p. 46-48. autour de l'équation de verhulst.
- [26] Jacques BAIR and Jean MAWHIN. Modèles de type proie-prédateur, 2011.
- [27] Ivan Balenović, Hrvoje Marjanović, Dijana Vuletić, Elvis Paladinić, Ostrogović Sever, Maša Zorana, and Krunoslav Indir. Quality assessment of high density digital surface model over different land cover classes. *Periodicum biologorum*, 117(4) :459–470, 2016.
- [28] Edward Ballard, AM Mistikawi, Mohamed Soliman el Zoheiry, et al. The desert locust, *Schistocerca gregaria* forsk., in egypt. *Bull. Minist. Agric. Egypt*, (110), 1932.
- [29] H Baltzer, P Braun, and W Köhler. Modeling population dynamics with cellular automata. *UNITED STATES DEPARTMENT OF AGRICULTURE FOREST SERVICE GENERAL TECHNICAL REPORT RM*, pages 703–712, 1996.
- [30] Adama Bamba, Bastien Dieppois, Abdourahamane Konaré, Thierry Pellarin, Ahmed Balogun, Nadine Dessay, Bamory Kamagaté, Issiaka Savané, and Arona Diédhiou. Changes in vegetation and rainfall over west africa during the last three decades (1981-2010). *Atmospheric and Climate Sciences*, 5(04) :367, 2015.
- [31] G.E.A.P.A. Batista, Yuan Hao, E. Keogh, and A. Mafra-Neto. Towards automatic classification on flying insects using inexpensive sensors. In *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, volume 1, pages 364–369, dec. 2011.

- [32] M Bell. The 2004 desert locust outbreak. *BULLETIN OF THE AMERICAN METEOROLOGICAL SOCIETY*, 86(6) :S60–S60, 2005.
- [33] Ahmed Salem SIDI BENAHI. Délimitation spatiale et temporelle des biotopes du criquet pèlerin schistocerca gregaria (zone pilote de grarat levras). 2008.
- [34] Atika Guendouz Benrima, Jean-François Duranton, Mohamed El Habib Benhammouda, and Bahia Doumandji Mitiche. Distribution des populations de criquet pèlerin schistocerca gregaria forsk 1775 (insecta orthoptera) en période de rémission en algérie de 1983 à 2000. *Science et changements planétaires/Sécheresse*, 18(4) :246–253, 2007.
- [35] Pottier Bernard. <http://wsn.univ-brest.fr/recapa>. B. Pottier, TF et N.Mel - recherche : "des capteurs à la connaissance".
- [36] Thibault Failler et Nicolas Melot. Bernard Pottier. Simulation de réseaux de capteurs et mobiles sur processeurs graphiques. *UMR LabSTICC : recherche des capteurs à la connaissance.*, 2012.
- [37] Andrew J. Bernoff and Chad M. Topaz. Nonlocal aggregation models : A primer of swarm equilibria. *SIAM Review*, 55(4) :709–747, 2013.
- [38] F Bertrand. La directive inspire. *GEOMATIK SCHWEIZ*, 105(8) :386, 2007.
- [39] Michael P Bishop, Radoslav Bonk, Ulrich Kamp Jr, and John F Shroder Jr. Terrain analysis and data modeling for alpine glacier mapping. *Polar Geography*, 25(3) :182–201, 2001.
- [40] Laura M Blackburn, Swidbert R Ott, Tom Matheson, Malcolm Burrows, and Stephen M Rogers. Motor neurone responses during a postural reflex in solitary and gregarious desert locusts. *Journal of insect physiology*, 56(8) :902–910, 2010.
- [41] Michael Blakemore and Max Craglia. Access to public-sector information in europe : Policy, rights, and obligations. *The Information Society*, 22(1) :13–24, 2006.
- [42] J. Blumenthal, M. Handy, F. Golatowski, M. Haase, and D. Timmermann. Wireless sensor networks - new challenges in software engineering. In *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, volume 1, pages 551–556 vol.1, Sept.
- [43] Csaba A Boer, Arie de Bruin, and Alexander Verbraeck. Distributed simulation in industry-a survey part 3-the hla standard in industry. In *2008 Winter Simulation Conference*, pages 1094–1102. IEEE, 2008.
- [44] PA Boroughs and RA McDonald. Principles of geographic information systems, 1998.
- [45] Joumana Boustany and Gabriella Salzano. La réutilisation des données publiques : quels dispositifs ? 8e Colloque international de l'ISKO-France Lille, 2011.
- [46] O. Boyinbode, H. Le, A. Mbogho, M. Takizawa, and R. Poliah. A survey on clustering algorithms for wireless sensor networks. In *Network-Based Information Systems (NBIS), 2010 13th International Conference on*, pages 358–364, Sept 2010.

- [47] L Brader, H Djibo, FG Faye, S Ghaout, M Lazar, PN Luzietoso, and MA Ould Baba. Towards a more effective response to desert locusts and their impacts on food security, livelihoods and poverty : Multilateral evaluation of the 2003–05 desert locust campaign. *evaluation*, 2006.
- [48] Anti-Locust Research Centre (Great Britain) and Maud J Norris. *Reproduction in the desert locust (Schistocerca gregaria Forsk.) in relation to density and phase*. 1952.
- [49] Jerome Buhl, David JT Sumpter, Iain D Couzin, Joe J Hale, Emma Despland, ER Miller, and Steve J Simpson. From disorder to order in marching locusts. *Science*, 312(5778) :1402–1406, 2006.
- [50] Chiara Buratti, Marco Martalò, Gianluigi Ferrari, and Roberto Verdone. *Sensor Networks with IEEE 802.15.4 Systems : Distributed Processing, MAC, and Connectivity*. Berlin : Springer Berlin, 2010.
- [51] Arthur W Burks and John Von Neumann. *Theory of self-reproducing automata*. University of Illinois Press, 1966.
- [52] Alan Burns. *Programming in occam 2*. Addison-Wesley Longman Publishing Co., Inc., 1987.
- [53] Li Jing-jing. C. Jian. Monitoring the oriental migratory locust plague based on the lai retrieved from remotely sensed data. *IEEE conference publications*, 2 :312–315, 2008.
- [54] Fraser Cain. What percentage of the earth’s land surface is desert ? <http://www.universetoday.com/65639/what-percentage-of-the-earths-land-surface-is-desert/>, juin 2010.
- [55] Tyrone D Cannon, Jaakko Kaprio, Jouko Lönnqvist, Matti Huttunen, and Markku Koskenvuo. The genetic epidemiology of schizophrenia in a finnish twin cohort : a population-based modeling study. *Archives of general psychiatry*, 55(1) :67–74, 1998.
- [56] Carmena and Jose M. Towards a bionic bat : A biomimetic investigation of active sensing, doppler-shift estimation, and ear morphology design for mobile robots. *mobile robots*, 2002.
- [57] Climate Change. *Climate of change*, 2001.
- [58] JW Chapman and VA Drake. *Insect migration*. 2010.
- [59] Yves Cherruault. *Modèles et méthodes mathématiques pour les sciences du vivant*. Presses Universitaires de France-PUF, 1998.
- [60] Sory Cisse, Saïd Ghaout, Ahmed Mazih, Babah Ebbe, Mohamed Abdallahi Ould, Ahmed Salem Benahi, and Cyril Piou. Effect of vegetation on density thresholds of adult desert locust gregarization from survey data in mauritania. *Entomologia Experimentalis et Applicata*, 149(2) :159–165, 2013.
- [61] Edward A Codling, Michael J Plank, and Simon Benhamou. Random walk models in biology. *Journal of the Royal Society Interface*, 5(25) :813–834, 2008.

- [62] George F Coulouris, Jean Dollimore, and Tim Kindberg. *Distributed systems : concepts and design*. pearson education, 2005.
- [63] K. CRESSMAN. Current methods of desert locust forecasting at fao 1. *EPPO Bulletin*, 26(3-4) :577–585, 1996.
- [64] Keith Cressman. The use of new technologies in desert locust early warning. *Outlooks on Pest Management*, 19(2) :55–59, 2008.
- [65] Judith S Dahmann and Katherine L Morse. High level architecture for simulation : An update. In *Distributed Interactive Simulation and Real-Time Applications, 1998. Proceedings. 2nd International Workshop on*, pages 32–40. IEEE, 1998.
- [66] Paul J Darwen and David G Green. Viability of populations in a landscape. *Ecological Modelling*, 85(2) :165–171, 1996.
- [67] Atmel ATmega32 Datasheet. Atmel corporation. *Technical Support*, 2009.
- [68] John H Davies. *MSP430 microcontroller basics*. Elsevier, 2008.
- [69] Agence Nationale de l'Aménagement du Territoire. Basegéo : Base de données géographiques du sénégal. [http ://www.basegeo.gouv.sn](http://www.basegeo.gouv.sn).
- [70] J. P. DEMPSTER. The population dynamics of grasshoppers and locusts. *Biological Reviews*, 38(4) :490–529, 1963.
- [71] Organisation des nations unies pour l'agriculture et l'alimentation. L'observatoire acridien : criquet pèlerin. [http ://www.fao.org/ag/locusts/fr/archives/archive/1823/1935/index.html](http://www.fao.org/ag/locusts/fr/archives/archive/1823/1935/index.html).
- [72] Organisation des nations unies pour l'alimentation et l'agriculture. Les statistiques à la fao. [http ://www.fao.org/statistics/fr/](http://www.fao.org/statistics/fr/), 2016.
- [73] Organisation des nations unies pour l'alimentation et l'agriculture (FAO). Données statistiques de la fao : Faostat. [http ://faostat3.fao.org/home/](http://faostat3.fao.org/home/).
- [74] Organisation des nations unies pour l'alimentation et l'agriculture (FAO). Geonetwork. [http ://www.fao.org/geonetwork/srv/en/main.home](http://www.fao.org/geonetwork/srv/en/main.home).
- [75] Organisation des nations unies pour l'alimentation et l'agriculture (FAO). Base de données principale d'aquastat, organisation des nations unies pour l'alimentation et l'agriculture (fao), 2016.
- [76] John B Dunning Jr, David J Stewart, Brent J Danielson, Barry R Noon, Terry L Root, Roland H Lamberson, and Ernest E Stevens. Spatially explicit population models : current forms and future uses. *Ecological Applications*, 5(1) :3–11, 1995.
- [77] J.-F. Duranton and M.Lecoq. *LE CRIQUET PELERIN AU SAHEL*, volume 6 of *Collection Acridologie Opérationnelle*. CIRAD/PRIFAS, 1990.
- [78] Hritam Dutta, Thibault Failler, Nicolas Melot, Bernard Pottier, and Serge Stinckwich. An execution flow for dynamic concurrent systems : simulation of wsn on a smalltalk/cuda environment. DYROS, in SIMPAR'10, Darmstadt, November 2010.

- [79] Hritam Dutta, Thibault Failler, Nicolas Melot, and Serge Stinckwich. An execution flow for dynamic concurrent systems :simulation of wsn on a smalltalk/cuda environment. in *SIMPAR'10, Darmstadt*, 2010.
- [80] Peggy E Ellis. Marching in locust hoppers of the solitary phase. *Nature*, 166(4212) :151–151, 1950.
- [81] International energy data and analysis. <http://www.eia.gov/beta/international/>.
- [82] Thibault Failler et B. Pottier. Netgen : un générateur de code pour cuda, principes, implémentation et performances. *Lab-STICC*, 2010.
- [83] Le Gall Johann et Herry Romain. Les réseaux de capteurs en ville : Calculs de couverture. *Thème : Systèmes Pervasifs (équipe CACS) - LabSTICC, UMR CNRS 3192*, Master 1 Informatique, juin 2011.
- [84] Leonhard Euler. *Introductio in analysin infinitorum*, volume 2. MM Bousquet, 1748.
- [85] Brian Evans. *Beginning Arduino Programming*. Springer, 2011.
- [86] MS Evans. Cellular automata-brian’s brain (2002).
- [87] FAO. L’observatoire acridien. <http://www.fao.org/ag/locusts/fr/activ/index.html>. Accessed : 19/10/2012.
- [88] N Ferry. Assimilation et surface libre dans les modeles océaniques mercator. rapport interne projet mercator. Technical report, Référence MOO-ST-410-231-MER, 2003.
- [89] Massimo Ficco, Giovanni Avolio, Francesco Palmieri, and Aniello Castiglione. An hla-based framework for simulation of large-scale critical systems. *Concurrency and Computation : Practice and Experience*, 28(2) :400–419, 2016.
- [90] Michael J Flynn. Some computer organizations and their effectiveness. *IEEE transactions on computers*, 100(9) :948–960, 1972.
- [91] Food and Agriculture Organization of the United Nations. *Report of the Fifth Session of the FAO Desert Locust Control Committee : Held in Rome, Italy 16-21 June, 1958*. Meeting report. Food and Agriculture Organization of the United Nations, 1958.
- [92] Gerald Forkuor and Ben Maathuis. *Comparison of SRTM and ASTER Derived Digital Elevation Models Over Two Regions in Ghana-Implications for Hydrological and Environmental Modeling*. INTECH Open Access Publisher, 2012.
- [93] Petter Forskal et al. *Descriptiones animalium, avium, amphibiorum, piscium, insectorum, vermium, quae in itinere orientali observavit Petrus Forskal... edidit Carsten Niebuhr. Adjuncta est materia medica kahirina atque tabula Maris Rubri geographica*. ex-officina Mölleri, 1775.
- [94] Raffaella Franci. Il liber abaci di leonardo fibonacci 1202-2002. *Bollettino dell’Unione Matematica Italiana*, 5(2) :293–328, 2002.
- [95] Edward Fredkin. An informational process based on reversible universal cellular automata. *Physica D : Nonlinear Phenomena*, 45(1-3) :254–270, 1990.

- [96] Karen Frenken. *Irrigation in Africa in figures : AQUASTAT survey, 2005*, volume 29. Food & Agriculture Org., 2005.
- [97] Pr Nicolas Fressengeas and Hervé Frezza-Buet. Résolution automatique des équations différentielles aux dérivées partielles. *J. des Automates Cellulaires*, avril 2014.
- [98] Richard Fujimoto. Parallel and distributed simulation. In *Proceedings of the 2015 Winter Simulation Conference*, pages 45–59. IEEE Press, 2015.
- [99] Yohann Le Gall and Romain Herry. Calcul de couverture radio par lancer de rayon. Technical report, Master UBO, June 2011.
- [100] Jay Gao. Resolution and accuracy of terrain representation by grid dems at a micro-scale. *International Journal of Geographical Information Science*, 11(2) :199–212, 1997.
- [101] Martin Gardner. Mathematical games : The fantastic combinations of john conway’s new solitaire game “life”. *Scientific American*, 223(4) :120–123, 1970.
- [102] G.A.Sword, M.Lecoq, and S.J.Simpson. Phase polyphenism and preventative locust management. *Journal of Insect Physiology*, 56(8) :949–957, 2010.
- [103] Jean-Marc Ginoux. Le paradoxe du modèle prédateur-proie de vito volterra. *Laboratoire PROTEE, Université du Sud, IUT de Toulon, BP, 20132* :83957, 2006.
- [104] Volker Grimm. Ten years of individual-based modelling in ecology : what have we learned and what could we learn in the future ? *Ecological modelling*, 115(2) :129–148, 1999.
- [105] Volker Grimm and Steven F Railsback. *Individual-based modeling and ecology*. Princeton university press, 2013.
- [106] HLA Working Group et al. Ieee standard for modeling and simulation (m&s) high level architecture (hla)-framework and rules. *IEEE Standard*, pages 1516–2000, 2000.
- [107] Wei Guo, Xianhui Wang, Zongyuan Ma, Liang Xue, Jingyao Han, Dan Yu, and Le Kang. Csp and takeout genes modulate the switch between attraction and repulsion during behavioral phase change in the migratory locust. *PLoS Genet*, 7(2) :e1001291, 2011.
- [108] Ashley B Gurney. The upsurges and recessions of the desert locust plague ; an historical survey. *Bulletin of the Entomological Society of America*, 13(3) :244–244, 1967.
- [109] Tomislav Hengl and Ian S Evans. Mathematical and digital models of the land surface. *Developments in soil science*, 33 :31–63, 2009.
- [110] MI Hernández-Zul, JA Quijano-Carranza, R Yáñez-López, RV Ocampo-Velázquez, I Torres-Pacheco, RG Guevara-González, and AE Castro-Ramírez. Dynamic simulation model of central american locust schistocerca piceifrons (orthoptera : Acrididae). *Florida Entomologist*, 96(4) :1274–1283, 2013.

- [111] Claudia A Herok, Stephan Krall, et al. *Economics of desert locust control*. Deutsche Gesellschaft für Technische Zusammenarbeit (GTZ) GmbH, 1995.
- [112] Stefanie M Herrmann, Assaf Anyamba, and Compton J Tucker. Recent trends in vegetation dynamics in the african sahel and their relationship to climate. *Global Environmental Change*, 15(4) :394–404, 2005.
- [113] Robert J. Hijmans, Susan E. Cameron, Juan L. Parra, Peter G. Jones, and Andy Jarvis. Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology*, 25(15) :1965–1978, 2005.
- [114] Christian Hirt. *Digital Terrain Models*, pages 1–6. Springer International Publishing, Cham, 2014.
- [115] Timothy G Howard and Adele Foreword By-Goldberg. *The Smalltalk Developer's Guide to VisualWorks with Disk*. Cambridge University Press, 1995.
- [116] Philip Hunter-Jones et al. Laboratory studies on the inheritance of phase characters in locusts. *Anti-Locust Bull.*, (29), 1958.
- [117] Hiep Xuan Huynh and Bernard Pottier. A federated approach for simulations in cyber-physical systems. In *Context-Aware Systems and Applications : 4th International Conference, ICCASA 2015, Vung Tau, Vietnam, November 26-27, 2015, Revised Selected Papers*, volume 165, page 165. Springer, 2016.
- [118] IEEE. Ieee standard for modeling and simulation (m&s) high level architecture (hla)- federate interface specification. *IEEE Std 1516TM-2010*, (1-378), 2010.
- [119] IEEE. Ieee standard for modeling and simulation (m&s) high level architecture (hla)- framework and rules. *IEEE Std 1516TM-2010*, pages 1–38, 2010.
- [120] IEEE. Ieee standard for modeling and simulation (m&s) high level architecture (hla)- object model template specification. *IEEE Std 1516.2TM-2010*, pages 1–110, 2010.
- [121] ” "IEEE Standard 802.15.4-2011. Le standard ieee 802.15.4. 2011.
- [122] Inmos. *Occam 2.1 Reference Manual*. SGS-THOMSON Microelectronics Limited, 1995.
- [123] EARTH INSTITUTE and COLUMBIA UNIVERSITY. International research institute for climate and society. <http://iri.columbia.edu/?bbox=bb>
- [124] Adnan Iqbal and Bernard Pottier. Meta-simulation of large wsn on multi-core computers. *SIMSPRING, NUST/SEECS Islamabad UBO/LabSTICC Brest*, 2010.
- [125] Adnan Iqbal and Bernard Pottier. Meta-Simulation of Large WSN on Multi-core Computers. DEVS10, in SpringSim SCS Conference, Orlando, USA, April 2010.
- [126] Robert M Itami. Simulating spatial dynamics : cellular automata theory. *Landscape and urban planning*, 30(1) :27–47, 1994.
- [127] Agre. J and Clare. L. An integrated architecture for cooperative sensing networks. *computer*, 33(5) :106–108, 2000.

- [128] W OOLDRIDGE M. J ENNINGS N., S YCARA K. A roadmap of agent research and development :autonomous agents and multi-agent systems. 1998.
- [129] Andy Jarvis, Hannes Isaak Reuter, Andrew Nelson, and Edward Guevara. Hole-filled srtm for the globe version 4. *available from the CGIAR-CSI SRTM 90m Database (<http://srtm.csi.cgiar.org>)*, 2008.
- [130] Steen R. Joffe. Economic and policy issues in desert locust management : a preliminary analysis. DESERT LOCUST TECHNICAL SERIES No. AGP/DL/TS/27 AO/EMPRES Workshop on Economics in Desert Locust Management Cairo, September 1997.
- [131] Holger Karl and Andreas Willig. *Protocols and Architectures for Wireless Sensor Networks*. New York, NY : Wiley, J, 2007.
- [132] John Stodart Kennedy. The migration of the desert locust (*schistocerca gregaria* forsk.). i. the behaviour of swarms. ii. a theory of long-range migrations. *Philosophical Transactions of the Royal Society of London B : Biological Sciences*, 235(625) :163–290, 1951.
- [133] Kyoung-Dae Kim and P.R. Kumar. Cyber physical systems : A perspective at the centennial. *Proceedings of the IEEE*, 100(Special Centennial Issue) :1287 –1308, may 2012.
- [134] Toshihiko Komatsuzaki and Yoshio Iwata. Study on acoustic field with fractal boundary using cellular automata. *Cellular Automata*, pp :282–290, 2008.
- [135] S Krall and C Herok. Economics of desert locust control. In *New Strategies in Locust Control*, pages 401–413. Springer, 1997.
- [136] G. Kremer, J. Osmont, and B. Pottier. A process oriented development flow for wireless sensor networks. In *International Workshop on Smalltalk Technology (IWST'09), ESUG, ACM*, Sep 2009.
- [137] Bao Hoai Lam, Hoang Van Tran, Hiep Xuan Huynh, and Bernard Pottier. Synchronous networks for insects surveillance. In *Proceedings of the Sixth International Symposium on Information and Communication Technology*, page 27. ACM, 2015.
- [138] Hoai Bao Lam, Hiep Xuan Huynh, Pierre Yves Lucas, Mahamadou Traore, and Bernard Pottier. Monitoring environmental factors in mekong delta of vietnam using wireless sensor network approach.
- [139] Hoai Bao Lam, Tai Tan Phan, Long Huynh Vuong, Hiep Xuan Huynh, and Bernard Pottier. Designing a brown planthoppers surveillance network based on wireless sensor network approach. *arXiv preprint arXiv :1312.3692*, 2013.
- [140] SN Lane, KS Richards, and JH Chandler. Developments in monitoring and modelling small-scale river bed topography. *Earth Surface Processes and Landforms*, 19(4) :349–368, 1994.
- [141] Christopher G Langton. Studying artificial life with cellular automata. *Physica D : Nonlinear Phenomena*, 22(1) :120–149, 1986.

- [142] Gilles Lasnier, Janette Cardoso, Pierre Siron, Claire Pagetti, and Patricia Derler. Distributed simulation of heterogeneous and real-time systems. In *Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*, DS-RT '13, pages 55–62, Washington, DC, USA, 2013. IEEE Computer Society.
- [143] Gilles Lasnier, Janette Cardoso, Pierre Siron, Claire Pagetti, and Patricia Derler. Distributed simulation of heterogeneous and real-time systems. In *Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*, pages 55–62. IEEE Computer Society, 2013.
- [144] M Lecoq. Le criquet pèlerin : enseignements de la dernière invasion et perspectives offertes par la biomodélisation. *La lutte anti-acridienne. AUPELF-UREF, John Libbey Eurotext, Paris*, pages 71–98, 1991.
- [145] Edward Lee et al. Cyber physical systems : Design challenges. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, pages 363–369. IEEE, 2008.
- [146] Edward A Lee. Cyber-physical systems-are computing foundations adequate. In *Position Paper for NSF Workshop On Cyber-Physical Systems : Research Motivation, Techniques and Roadmap*, volume 2. Citeseer, 2006.
- [147] Edward A Lee. Computing foundations and practice for cyber-physical systems : A preliminary report. *University of California, Berkeley, Tech. Rep. UCB/EECS-2007-72*, 2007.
- [148] Edward A. Lee. Cyber physical systems : Design challenges. Technical Report UCB/EECS-2008-8, EECS Department, University of California, Berkeley, Jan 2008.
- [149] Edward A Lee. Cps foundations. In *Proceedings of the 47th Design Automation Conference*, pages 737–742. ACM, 2010.
- [150] Patrick H Leslie. On the use of matrices in certain population mathematics. *Biometrika*, 33(3) :183–212, 1945.
- [151] Zhilin Li, Christopher Zhu, and Chris Gold. *Digital terrain modeling : principles and methodology*. CRC press, 2004.
- [152] Buquan Liu, Yiping Yao, Jing Tao, and Huaimin Wang. Implementation of time management in a runtime infrastructure. In *Proceedings of the 2006 Winter Simulation Conference*, pages 1044–1052. IEEE, 2006.
- [153] Leslie M Loew and James C Schaff. The virtual cell : a software environment for computational cell biology. *TRENDS in Biotechnology*, 19(10) :401–406, 2001.
- [154] Alfred J Lotka. *Elements of physical biology*. 1925.
- [155] Timothy B Love* and Vadlamani Kumar. P5.4 a 20-year daily africa precipitation climatology using satellite and gauge data. In *14th Conference on Applied Climatology*, RS Information Systems, Inc. and NOAA/NWS/NCEP/CPC, January 2004.

- [156] P-Y Lucas, Nguyen Huu Van Long, Tuyen Phong Truong, and Bernard Pottier. Wireless sensor networks and satellite simulation. In *International Conference on Wireless and Satellite Systems*, pages 185–198. Springer, 2015.
- [157] M.H.L Luong, M. Launois, and T. Rachadi. *LA LUTTE CHIMIQUE CONTRE LES CRIQUETS AU SAHEL*. Collection Acridologie Opérationnelle no 3. CIRAD/PRIFAS, 1988.
- [158] Nancy A Lynch. *Distributed algorithms*. Morgan Kaufmann, 1996.
- [159] Launois. M. *Manuel pratique d'identification des principaux Acridiens du Sahel*. GERDAT/PRIFAS, Paris, 1979.
- [160] Ludovic Macaire, Nicolas Vandenbroucke, and Jack-Gérard Postaire. 04-segmentation d'images par classification spatio-colorimétrique des pixels. 2004.
- [161] Sk Sazid Mahammad and R Ramakrishnan. Geotiff-a standard image file format for gis applications. *Map India*, pages 28–31, 2003.
- [162] Jacek Malczewski. Gis-based multicriteria decision analysis : a survey of the literature. *International Journal of Geographical Information Science*, 20(7) :703–726, 2006.
- [163] Thomas Robert Malthus, Pierre Prévost, Guillaume Prevost, Pellegrino Rossi, Charles Comte, and Joseph Garnier. *Essai sur le principe de population*. Guillaumin, 1852.
- [164] John Markoff and New-York Times. Can't find a parking spot ? check smartphone. <http://www.nytimes.com/2008/07/12/business/12newpark.html>, july 2008.
- [165] M.Lecoq. *LES CRIQUETS DU SAHEL*. Collection Acridologie Opérationnelle no 1. CIRAD/PRIFAS, 1988.
- [166] M.Lecoq. *La lutte anti-acridienne*. AUPELF-UREF, 1991.
- [167] M.Lecoq. Desert locust threat to agricultural development and food security and fao/international role in its control. *Arab Journal of Plant Protection*, pages 188–193, 2003.
- [168] M.Lecoq. Vers une solution durable au probleme du criquet pelerin ? *Secheresse*, 15 :217–224, 2004.
- [169] M.Lecoq. Desert locust management :from ecology to anthropology. *Journal of Orthoptera Research*, 14 :179–186, 2005.
- [170] M.Lecoq and J.Mestre. *La Surveillance des Sauteriaux du Sahel*. Collection Acridologie Opérationnelle no 2. CIRAD/PRIFAS, 1988.
- [171] Gov Monitor. Victoria invests \$1.5 million to help farmers control locust. *Gov Monitor*, 2010.
- [172] Gordon E Moore. Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp. 114 ff. *IEEE Solid-State Circuits Newsletter*, 3(20) :33–35, 2006.

- [173] John von Neumann and Arthur W Burks. Theory of self-reproducing automata. 1966.
- [174] Silvia Nittel. A survey of geosensor networks : Advances in dynamic environmental monitoring. *Sensors*, 9(7) :5664–5678, 2009.
- [175] NVIDIA. *NVIDIA CUDA Programming Guide 2.0*. NVIDIA, 2008.
- [176] Luís M Oliveira and Joel J Rodrigues. Wireless sensor networks : a survey on environmental monitoring. *Journal of communications*, 6(2) :143–151, 2011.
- [177] Swidbert R Ott, Heleen Verlinden, Stephen M Rogers, Caroline H Brighton, Pei Shan Quah, Rut K Vleugels, Rik Verdonck, and Jozef Vanden Broeck. Critical role for protein kinase a in the acquisition of gregarious behavior in the desert locust. *Proceedings of the National Academy of Sciences*, 109(7) :E381–E387, 2012.
- [178] David Pedgley et al. *Desert locust forecasting manual. Volumes I and II*. Centre for Overseas Pest Research., 1981.
- [179] M Phipps and A Langlois. Automates cellulaires-application à la simulation urbaine. *Hermes Science*, 1997.
- [180] Clifford A Pickover. *The math book : from Pythagoras to the 57th dimension, 250 milestones in the history of mathematics*. Sterling Publishing Company, Inc., 2009.
- [181] Kris Pister. Berkeley sensor and actuator center. <http://www-bsac.eecs.berkeley.edu/>.
- [182] Jonathan Pletzke. *Advanced Smalltalk*. John Wiley & Sons, Inc., 1996.
- [183] George Basil Popov. *Les larves des criquets du Sahel*. Overseas Development Natural Resources Institute, 1989.
- [184] B. Pottier. Une approche descendante de la conception de systèmes distribués :netgen. In *LabSTICC, UMR CNRS 3192*, Apr 2010.
- [185] Bernard Pottier. "global warming, global sharing" - labsticc- université de bretagne occidentale (ubo)- france. RESSACS'14 at IRD/Bondy, Web site and program.
- [186] Bernard Pottier. <http://wsn.univ-brest.fr/pottier/pucescom.pdf>. Cours B. Pottier Labsticc -UBO.
- [187] Bernard Pottier and Pierre-Yves Lucas. Concevoir, simuler, exécuter une chaîne de développement pour réseau de capteurs. In *Ubimob'12*, pages 94–107. Cépaduès éditions, 2012.
- [188] Bernard Pottier and Pierre-Yves Lucas. Dynamic networks" netgen : objectives, installation, use, and programming. *Université de Bretagne Occidentale*, 2014.
- [189] Bernard Pottier and Serge Stinckwich. An execution flow for dynamic concurrent systems :simulation of wsn on a smalltalk/cuda environment. in *SIMPAR'10, Darmstadt*, 2010.

- [190] Nathalie Poupaert and Katleen Janssen. La directive 2003/98 du 17 novembre 2003 concernant la réutilisation des informations du secteur public.
- [191] Michel Prieur. La convention d'aarhus, instrument universel de la démocratie environnementale. *Revue juridique de l'environnement*, pages 9–29, 1999.
- [192] C.S Raghavendra, K.M Sivalingam, and T. Znati. *Wireless Sensors Networks*. Springer-verlag, 2004.
- [193] RC Rainey. Recent developments in the use of insecticides from aircraft against locusts. In *Rep 6th Commonw Entomol Conf London*, volume 6, pages 48–51, 1954.
- [194] Nicolas Reimen. *Contribution à l'étude des automates cellulaires*. PhD thesis, 1993.
- [195] Jean-Philippe Rennard. Implementation of logical functions in the game of life. In *Collision-based computing*, pages 491–512. Springer, 2002.
- [196] J.A.W.A. Reus and P.M. Symmons. A model to predict the incubation and nymphal development periods of the desert locust, *schistocerca gregaria* (orthoptera : Acrididae). *Bulletin of Entomological Research*, 82 :517–520, 12 1992.
- [197] Derek Riley, Emeka Eyisi, Jia Bai, Xenofon Koutsoukos, Yuan Xue, and Janos Sztipanovits. Networked control system wind tunnel (ncswt) : an evaluation tool for networked multi-agent systems. In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, pages 9–18. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.
- [198] N Ritter and M Ruth. The geotiff data interchange standard for raster geographic images. *International Journal of Remote Sensing*, 18(7) :1637–1647, 1997.
- [199] Niles Ritter, Mike Ruth, Brett Borup Grissom, George Galang, John Haller, Gary Stephenson, Steve Covington, Tim Nagy, Jamie Moyers, Jim Stickley, et al. Geotiff format specification geotiff revision 1.0. URL : <http://www.remotesensing.org/geotiff/spec/geotiffhome.html>, 2000.
- [200] Gianluca Mazzini Roberto Verdone, Davide Dardari and Andrea Conti. *Wireless Sensor and Actuator Networks Technologies, Analysis and Design*. Elsevier, 2008.
- [201] E Rodriguez and JM Martin. Theory and design of interferometric synthetic aperture radars. In *Radar and Signal Processing, IEE Proceedings F*, volume 139, pages 147–159. IET, 1992.
- [202] J Roffey. Desert locust population dynamics : a preliminary approach to a simulation model. Rome : *Food and Agriculture Organization*, 1990.
- [203] J Roffey and J I Magor. Desert locust population dynamics parameters. FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS, PLANT PRODUCTION AND PROTECTION DIVISION LOCUSTS AND OTHER MIGRATORY PESTS GROUP No. A G P/DL/TS/30 DESERT LOCUST TECHNICAL SERIES No. A G P/DL/TS/30, Rome 2003.

- [204] J Roffey, Z Waloff, and D Pedgley. The desert locust. *Desert Locust Forecasting Manual*, 1 :4–53, 1981.
- [205] Jeremy Roffey and George Popov. Environmental and behavioural processes in a desert locust outbreak. *Nature*, 219 :446–450, 1968.
- [206] Paul Rosen, Scott Hensley, Ian R Joughin, Fuk K Li, Søren N Madsen, Ernesto Rodriguez, Richard M Goldstein, et al. Synthetic aperture radar interferometry. *Proceedings of the IEEE*, 88(3) :333–382, 2000.
- [207] François-Xavier Roux. Programmation des super-calculateurs scientifiques vectoriels et parallèles. *Note technique- Office national d'études et de recherches aérospatiales*, 1991.
- [208] Teodora Sanislav and Liviu Miclea. Cyber-physical systems-concept, challenges and research areas. *Journal of Control Engineering and Applied Informatics*, 14(2) :28–33, 2012.
- [209] Robert R Schaller. Moore's law : past, present and future. *Spectrum, IEEE*, 34(6) :52–59, 1997.
- [210] HA Simon and Howard H Pattee. Hierarchy theory : The challenge of complex systems. *Hierarchy Theory : The Challenge of Complex Systems*, 1973.
- [211] Stephen J Simpson, ALAN McCaffery, and BERND F HAeGELE. A behavioural analysis of phase change in the desert locust. *Biological Reviews*, 74(4) :461–480, 1999.
- [212] Stephen J. Simpson and Gregory A. Sword. Locusts. *Current Biology*, 18(9) :R364 – R366, 2008.
- [213] Kenny Smith, Henry Brighton, and Simon Kirby. Complex systems in language evolution : the cultural emergence of compositional structure. *Advances in Complex Systems*, 6(04) :537–558, 2003.
- [214] BASMA SOLAIMAN. Energy optimization in wireless sensor networks using a hybrid k-means pso clustering algorithm. *Turkish Journal of Electrical Engineering & Computer Sciences*, 24(4) :2679–2695, 2016.
- [215] sparkfun. The mcp9700/01 datasheet. <https://www.sparkfun.com/datasheets/DevTools/LilyPad/MCP9700/>
- [216] Alison Steedman et al. *Locust handbook*. Number 3. ed. Natural Resources Institute, 1990.
- [217] Streetline. Streetline – connecting the real world. <http://www.streetline.com>.
- [218] PM Symmons and K Cressman. Directives sur le criquet pèlerin 1. biologie et comportement. *Organisation des nations unies pur l'alimentation et l'agriculture*, 2001.
- [219] P.M. Symmons, K. Cressman, H. van der Valk, and J.W. Everts. Criquet pèlerin directives. *FAO*, 1, 2003.

- [220] PM Symmons, SM Green, RA Robertson, and KG Wardhaugh. The production of distribution maps of the incubation and hopper development periods of the desert locust *schistocerca gregaria* (forsk.)(orthoptera, acrididae). *Bulletin of Entomological Research*, 64(03) :443–451, 1974.
- [221] Véronique Terrier. Two-dimensional cellular automata and their neighborhoods. *Theoretical computer science*, 312(2) :203–222, 2004.
- [222] R Tomassone and C Dervin. Masson jp-1993 : Biométrie : Modélisation de phénomènes biologiques.
- [223] Chad M Topaz, Andrew J Bernoff, Sheldon Logan, and Wyatt Toolson. A model for rolling swarms of locusts. *The European Physical Journal Special Topics*, 157(1) :93–109, 2008.
- [224] Chad M Topaz, Maria R D’Orsogna, Leah Edelstein-Keshet, and Andrew J Bernoff. Locust dynamics : behavioral phase change and swarming. *PLoS Comput Biol*, 8(8) :e1002642, 2012.
- [225] Alan Mathison Turing. *Morphogenesis*, volume 4. North Holland, 1992.
- [226] Stanislaw M Ulam. *Adventures of a Mathematician*. Univ of California Press, 1991.
- [227] Boris UVAROV. *Grasshoppers and locusts*. 1966.
- [228] Boris Uvarov et al. *Grasshoppers and locusts. A handbook of general acridology. Volume 2. Behaviour, ecology, biogeography, population dynamics*. Centre for Overseas Pest Research., 1977.
- [229] Tran Van Hoang. Cyber-physical systems and mixed simulations, 2015.
- [230] Jakob J Van Zyl. The shuttle radar topography mission (srtm) : a breakthrough in remote sensing of topography. *Acta Astronautica*, 48(5) :559–565, 2001.
- [231] John Von Neumann. The general and logical theory of automata. *Cerebral mechanisms in behavior*, 1(41) :1–2, 1951.
- [232] Z Waloff. upsurges and recessions of the desert locust plague ; an historical survey. 1966.
- [233] Zena Waloff et al. Some temporal characteristics of desert locust plagues. *Some temporal characteristics of desert locust plagues.*, (13), 1976.
- [234] K Wardhaugh, Y Ashour, AO Ibrahim, AM Khan, M Bassonbol, et al. Experiments on the incubation and hopper development periods of the desert locust (*schistocerca gregaria* forskál) in saudi arabia. *Anti-Locust Bull.*, (45), 1969.
- [235] Libelium Comunicaciones Distribuidas S.L. Waspnote Datasheet. <http://www.libelium.com/support/waspnote>. Accessed : 13/10/2012.
- [236] Mark Weiser, Rich Gold, and John Seely Brown. The origins of ubiquitous computing research at parc in the late 1980s. *IBM systems journal*, 38(4) :693, 1999.

-
- [237] Wikipédia. Projection de mercator — wikipédia, l'encyclopédie libre, 2016. [En ligne ; Page disponible le 16-janvier-2017].
- [238] Stephen Wolfram. Cellular automata as simple self-organizing systems. *Caltech preprint CALT-68-938*, 5, 1982.
- [239] Stephen Wolfram. *Cellular automata and complexity : collected papers*, volume 1. Addison-Wesley Reading, 1994.
- [240] Stephen Wolfram et al. *Theory and applications of cellular automata*, volume 1. World Scientific Singapore, 1986.
- [241] The world bank group. Climate change knowledge portal. Climate Change Knowledge Portal for development practitioners and policy makers, 2016.
- [242] S. Yazdani, J. Cambonie, and B. Pottier. Coordinated concurrent shared memory accesses on a reconfigurable multimedia accelerator. *Microprocessor and Microsystems, Embedded Hardware Design, Elsevier*, 33(1) :13–23, 2009.
- [243] J. Zheng and M. J. Lee. “a comprehensive performance study of ieee 802.15.4”. *Sensor Network Operations, IEEE Press, Wiley InterScience*, pages 218–237, 2006.

Cell Modeling and physical Simulation : contribution to the analysis of population dynamics of insect pests

Abstract :

Insect pests have long been a serious threat to agriculture in many parts of the world. In particular, desert locusts (*Schistocerca Gregaria*) are the most feared because of the colossal damage they can inflict on a wide variety of crops.

The latter, in their breeding environment and a set of climatic parameters, can be assimilated to a biophysical system. A biophysical system associates several components characterized by their own complexity and that of their interactions. For example, rain brings moisture and canopy development that is critical in Desert Locust population dynamics. These biophysical systems can be controlled by various means, including a monitoring network coupling wireless sensors. The integration of biophysical systems and observation networks can be one of the major events of this decade. It allows an exchange between physical cyclical processes and information systems. These can collect and process field data and can also act on the physical system through feedback loops. This integration has been made possible thanks to technological advances noted in the fields of microelectronics and wireless transmission. It is found in emerging fields such as fine agriculture. This thesis is devoted to cellular modeling and simulation of physical systems. For this, a spatial and temporal sampling of a Desert Locust breeding area was carried out using specific tools with different resolutions, in order to generate the cellular systems. These cellular systems contain information such as wind, temperature, relative humidity and rainfall. Spatial sampling allows a migration representation and temporal sampling can track the local evolution of individuals in a cell.

Cellular systems are transcribed into communicating process networks and evolve synchronously to reproduce and simulate the phenomenon of intercellular migration and the life cycle of the Desert Locust in a cell. The construction of these systems has been explored in depth, making vary their geographical context, the criteria for classification, the cells, connectivity between processes, collective behaviors. Exploration has focused on the generation of parallel codes and performance at the execution, mainly for the case of lite processes.

This work has led to the production of two parametric simulators, the first of which is aimed at Desert Locust population dynamics with a view to assessing the state of the locust situation through coupling between synchronous monitoring networks and biophysical systems. The second allows you to plan the deployment of wireless sensor networks in an area to determine the location of the sensors.

Keywords : Environmental modeling, physical simulation, Wireless sensor networks, cellular automata.

Modélisation cellulaire et simulation physique : contribution à l'analyse de la dynamique de population des insectes ravageurs

Résumé :

Les insectes ravageurs constituent depuis longtemps une menace sérieuse pour l'agriculture dans plusieurs régions du monde. En particulier, les criquets pèlerins (*Schistocerca Gregaria*) sont les plus redoutés à cause des dégâts colossaux qu'ils peuvent infliger à une large variété de cultures. Ces derniers, dans leur milieu de reproduction et un ensemble de paramètres climatiques, peuvent être assimilés à un système biophysique. Un système biophysique associe plusieurs composants caractérisés par leur complexité propre et celle de leurs interactions. Par exemple la pluie apporte de l'humidité et le développement du couvert végétal qui sont critiques dans la dynamique de population du criquet pèlerin. Ces systèmes biophysiques peuvent être contrôlés par divers moyens, dont un réseau de surveillance couplant des capteurs sans fil. L'intégration des systèmes biophysiques et des réseaux d'observation peut être un des faits majeurs de cette décennie. Elle permet un échange entre les processus cycliques physiques et les systèmes d'information. Ces derniers permettent de collecter et de traiter les données de terrain et peuvent aussi agir sur le système physique par le biais de boucles rétro-actives. Cette intégration a été possible grâce aux avancées technologiques notées dans les domaines de la micro électronique et de la transmission sans fil. On la trouve dans des domaines émergents tels que l'agriculture fine.

Cette thèse est consacrée à la modélisation cellulaire et à la simulation de systèmes physiques. Pour cela, un échantillonnage spatial et temporel d'une zone de reproduction des criquets pèlerins a été réalisé à l'aide d'outils spécifiques avec différentes résolutions, dans le but de générer les systèmes cellulaires. Ces systèmes cellulaires contiennent des informations telles que le vent, la température, l'humidité relative et la pluviométrie. L'échantillonnage spatial permet une représentation de migration et l'échantillonnage temporel permet de suivre l'évolution locale des individus dans une cellule.

Les systèmes cellulaires sont transcrits en réseaux de processus communicant et évoluent de manière synchrone afin de reproduire et de simuler le phénomène de migration intercellulaire et le cycle de vie du criquet pèlerin dans une cellule.

La construction de ces systèmes a été explorée en profondeur, en faisant varier leur contexte géographique, les critères de classification, la dimension des cellules, la connectivité entre les processus, les comportements collectifs. L'exploration a porté sur la génération de codes parallèles et les performances à l'exécution, essentiellement pour le cas des processus légers.

Ce travail a abouti à la production de deux simulateurs paramétriques, le premier est destiné à la dynamique de population des criquets pèlerins en vue d'évaluer l'état de la situation acridienne par un couplage entre les réseaux de surveillance synchrones et les systèmes biophysiques. Le second permet de planifier le déploiement de réseaux de capteurs sans fil dans une zone afin de déterminer l'emplacement des capteurs.

Mots-clé : Modélisation environnementale, simulation physique, Réseau de capteurs sans fil, automate cellulaire.