

Sommaire

<i>Résumé</i>	1
<i>Abstract</i>	2
<i>Sommaire</i>	3
<i>Liste Des Figures</i>	6
<i>Liste Des Tableaux</i>	8
<i>Introduction Générale</i>	9
Chapitre 1: Les Écosystèmes	11
1. Introduction	11
2. Les écosystèmes naturels	11
2.1 L'écologie.....	11
2.2 L'écosystème et ces composants.....	13
2.3 Quelques principes écologiques	14
2.4 Caractéristiques des populations	15
2.5 Diversité des écosystèmes	15
2.6 La complexité des écosystèmes.....	16
2.7 Co-évolution.....	17
2.8 Les interactions entre les populations	18
2.8.1 Compétition.....	18
2.8.2 Prédation.....	18
2.8.3 Parasitisme	19
2.8.4 Symbiose ou Mutualisme	20
2.8.5 Autres	20
3. Les écosystèmes artificiels	21
3.1 Un peu d'histoire	22
3.2 Simulation SMA et écologie	23
3.3 Quelques plates-formes de simulation des écosystèmes	24
3.4 Exemple d'écosystème artificiel : le modèle "Prédateur/Proie"	24
3.4.1 Les équations du modèle	25
3.4.2 Réponse fonctionnelle et Réponse numérique	28
3.4.3 Exemples d'application du modèle Prédateur/Proie	29
4. Conclusion.....	31
Chapitre 2 : Les Algorithmes Évolutionnaires Et Co-Évolutionnaires	32
1. Introduction	32
2. La résolution des problèmes complexes.....	32

2.1	Définition d'un problème complexe	32
2.2	Les propriétés d'un problème complexe	33
2.3	L'apport de la biologie en informatique.....	34
2.4	Les stratégies de résolution des problèmes complexes	35
2.4.1	Les stratégies Hill-climbing	36
2.4.2	Le Recuit Simulé.....	36
2.4.3	La recherche Tabou.....	37
2.4.4	Les algorithmes évolutionnaires.....	37
2.4.5	Les systèmes immunitaires artificiels	37
2.4.6	Optimisation par colonies de fourmis	38
2.4.7	Optimisation par essaim de particules.....	39
3.	La vie artificielle	39
4.	Les algorithmes évolutionnaires.....	40
4.1	Principe de base.....	40
4.1.1	L'initialisation	42
4.1.2	Evolution	42
4.1.3	Sélection	42
4.1.4	L'évaluation	43
4.1.5	Recombinaison.....	43
4.1.6	Mutation	44
4.1.7	L'arrêt.....	44
4.2	Les algorithmes génétiques	44
4.3	La programmation évolutionnaire.....	45
4.4	Les stratégies d'évolution.....	46
4.5	La programmation génétique	46
5.	Les algorithmes co-évolutionnaires	47
5.1	Pourquoi les ACEs ?	47
5.2	Principe de fonctionnement.....	48
5.3	Les algorithmes co-évolutionnaires compétitifs	48
5.4	Les algorithmes co-évolutionnaires coopératifs.....	49
5.5	L'application des ACEs pour la résolution des problèmes complexes	51
6.	L'algorithme PSO et l'extension PPO	53
6.1	L'algorithme PSO	53
6.1.1	Principe de fonctionnement.....	53
6.1.2	Comparaison entre PSO et les algorithmes génétiques.....	55
6.1.3	Les extensions de PSO	55
6.1.4	La topologie de voisinage	56
6.2	L'algorithme PPO	58
6.2.1	Motivations de cet algorithme.....	58
6.2.2	Principe de fonctionnement.....	59
7.	Conclusion.....	60
Chapitre 3: Introduction Au Traitement D'images		61
1.	Introduction	61
2.	Notions de base	62
2.1	Définition d'une image.....	62
2.2	Les images numériques	62
2.3	Codage des images numérique.....	63
2.3.1	L'image matricielle (ou image bitmap)	63

2.3.2	L'image vectorielle	63
2.4	Définition et résolution.....	64
2.5	Avantages et inconvénients de l'image numérique.....	65
2.6	Format d'image en mémoire	65
2.6.1	Images Noir et blanc (monochromes)	65
2.6.2	Images couleur	66
2.7	Les formats d'image.....	67
2.8	L'histogramme d'une image	67
2.9	Connexité dans les images	68
2.9.1	Voisinage d'un pixel	68
2.9.2	Connexité	69
2.9.3	Distances entre pixels.....	69
2.10	Le bruit dans une image	69
3.	Traitement d'images.....	70
3.1	Restauration.....	71
3.2	Binarisation d'une image	72
3.3	Segmentation.....	73
3.4	Analyse.....	74
3.5	Compression.....	75
3.6	Recalage	75
3.7	Filtres de convolution.....	75
3.7.1	Définition d'un filtre	75
3.7.2	Les différents types des filtres.....	76
3.8	Amélioration d'images.....	78
3.8.1	Modification d'histogramme.....	78
3.8.2	Réhaussement de contraste.....	78
4.	Le traitement d'images et les algorithmes évolutionnaires.....	79
5.	Le traitement d'images et les algorithmes co-évolutionnaires.....	80
6.	Conclusion.....	81
Chapitre 4 : Application Du PSO Et PPO Au Réhaussement D'images		82
1.	Introduction	82
2.	Formulation du problème	82
2.1	La fonction de transformation	83
2.2	La fonction de fitness	85
2.2.1	Le détecteur de bords 'Sobel'	85
2.2.2	Calcul de l'entropie	86
3.	L'application de PSO et PPO pour le réhaussement de contraste.....	86
3.1	Choix des paramètres	87
3.2	Les étapes de l'algorithme PSO	88
3.3	Les étapes de l'algorithme PPO	90
4.	Résultats expérimentaux et discussion	92
Conclusion Et Perspectives.....		97
Bibliographie		98



Liste Des Figures

Figure 1.1. Les branches de l'écologie.....	12
Figure 1.2. Les composants d'un écosystème.	13
Figure 1.3. Quelques exemples des écosystèmes prédateur-proie.	19
Figure 1.4. Le champignon et l'arbre vivent en association symbiotique.....	20
Figure 1.5. Relation de symbiose entre l'Oxpecker et le zèbre.....	20
Figure 1.6. Les étapes du processus de simulation.	21
Figure 1.7. Une représentation en perspective de PENGI, du point de vue du pingouin.....	24
Figure 1.8. L'interaction entre la population des Prédateurs et celle des Proies.....	25
Figure 1.9. La dynamique du système proie-prédateur	26
Figure 1.10. Une famille des orbites fermées dans le plan x-y.	27
Figure 1.11. Une solution avec les conditions initiales $(x, y) = (4, 1)$	27
Figure 1.12. Résultat du rapport historique de la compagnie Bay de Hudson.....	27
Figure 1.13. Visualisation de l'état de modèle de prédateur-proie.....	29
Figure 1.14. Distribution aléatoire des prédateurs et des proies à travers une grille de 2D.....	30
Figure 2.1. Emergence d'une propriété globale à partir d'interactions locales.....	33
Figure 2.2. Intervention de la théorie dans la construction d'une simulation.	35
Figure 2.3. L'arbre de différentes techniques adaptatives de recherche.	36
Figure 2.4. Exemple de la stratégie des colonies de fourmis, le contournement d'obstacles ..	38
Figure 2.5. Différentes branches des algorithmes évolutionnaires.	41
Figure 2.6. Cycle d'un algorithme évolutionnaire.	42
Figure 2.7. Modèle de la roulette. Représentation d'une population de 6 individus.	43
Figure 2.8. Opération de recombinaison.	43
Figure 2.9. Opération de mutation.	44
Figure 2.10. Exemple d'une représentation arborescente	47
Figure 2.11. Le système co-évolonnaire coopératif du Potter et du DeJong	51

Figure 2.12. Les différentes topologies de voisinage.....	57
Figure 2.13. Le principe de PSO	58
Figure 2.14. La relation Prédateur-Proie dans l'essaim de particule.	59
Figure 3.1. Image matricielle.	63
Figure 3.2. Image vectorielle.....	64
Figure 3.3. La résolution d'une image	64
Figure 3.4. Différentes quantifications d'une image.....	65
Figure 3.5. Le codage RGB.....	66
Figure 3.6. Histogramme d'une image.....	67
Figure 3.7. Voisinage d'un pixel.....	68
Figure 3.8. Exemple d'image bruitée.	70
Figure 3.9. Illustration de divers types d'opérations d'image	71
Figure 3.10. Principe général de la restauration d'images	72
Figure 3.11. Binarisation d'une image.....	73
Figure 3.12. L'opération de convolution.....	76
Figure 3.13. Principe du filtre non-linéaire	78
Figure 4.1. Appliquant à chaque Pixel dans l'image l'opération $T(f(x,y))$ dans le voisinage ..	84
Figure 4.2. Le détecteur de bords 'Sobel'	86
Figure 4.3. Représentation d'une particule	88
Figure 4.4. Les étapes de l'algorithme PSO.....	89
Figure 4.5. Application de PSO pour le réhaussement d'images.....	90
Figure 4.6. Les étapes de l'algorithme PPO.....	91
Figure 4.7. Résultat de l'application PPO	92
Figure 4.8. La technique d'égalisation d'histogramme.....	92
Figure 4.9. Histogramme d'une image trop foncée et celui d'une image trop claire.....	93
Figure 4.10. Gauche: (a): image originale, (b): image égalisée, (c): image PPO. (d): image PSO. Droite: leurs histogrammes correspondants.....	94
Figure.4.11. (a) Image originale, (b) Image PPO, (c) Image PSO, (d) Image égalisée	95
Figure 4.12. (a) Image bruitée, (b) Image égalisée, (c) Image PPO, (d) Image PSO.....	96



Liste Des Tableaux

Tableau 1. Les relations entre les individus.	18
Tableau 2. Récapitulatif de dates historiques du rapprochement entre les deux sciences.	34
Tableau 3. Types d'opérations d'images.	71
Tableau 4. Les paramètres de l'algorithme PSO.	88
Tableau 5. Les paramètres de l'algorithme PPO.	88
Tableau 6. Les Caractéristiques des images réhaussées par PPO et PSO.	94



Introduction Générale

Un autre domaine principal de recherche de la vie artificielle est la modélisation du comportement d'écosystème et la dynamique évolutionnaire des populations. Certains modèles de "vie artificielle" proposent de simuler, avec des moyens informatiques, des modèles métaphoriques, de ces phénomènes de génétique des populations, afin d'étudier les propriétés génériques de la dynamique sous-jacente indépendamment des contingences spécifiques à l'histoire effective de l'évolution de la vie. Quelques modèles proposent des représentations très stylisées de co-évolution d'espèces où l'évolution d'une espèce affecte l'évolution des espèces avec lesquelles elle interagit, par exemple par des relations de type chaîne alimentaire, prédateur-proie, symbiose, parasitisme, etc. au sein d'un écosystème et qui exhibent également des processus évolutionnaires caractérisés par des "équilibres ponctués" où les espèces évoluent de manière épisodique, avec des pointes intermittentes d'activité évolutionnaire intense, séparées par de longues périodes de "stase" caractérisées par un niveau plus faible d'activité évolutionnaire (graduelle).

Dans la nature, la co-évolution est le processus du changement génétique réciproque d'une espèce, ou de groupe, en réponse à des autres. Ce processus peut également être utilisé dans des algorithmes évolutionnaires, et récemment il y a eu un intérêt croissant pour l'application de la co-évolution dans des algorithmes évolutionnaires multi-objectives. Le changement réciproque observé dans la co-évolution peut être considéré soit comme une course aux armements concurrentielle (arms race), telle que la co-évolution des cas d'essai pour un problème (prédateurs) avec les solutions (proies) [Hillis 1990], ou plus récemment, des approches coopérative où les sous-populations séparées évoluent des composants de la solution [Coello 2003, Potter 2000].

Notre travail consiste à un premier lieu, à étudier les principes de base des algorithmes Co-évolutionnaires (ACEs), ainsi que quelques exemples de leur application à la résolution des problèmes complexes, ensuite un exemple d'ACE est présenté en détail, inspiré des relations entre deux espèces différentes, qui sont : les prédateurs et les proies. Ce modèle Prédateur-Proie est appliqué pour le réhaussement de contraste des images en niveaux de gris.

Ce mémoire est organisé comme suit :

Dans le chapitre 1, on va présenter les principes de base d'un écosystème naturel, ses composants, ses caractéristiques, les relations entre ses populations, ainsi que sa dynamique d'évolution. Ensuite, on va présenter des écosystèmes artificiels, qui simulent les comportements de quelques écosystèmes naturels. On finalise ce chapitre par l'étude d'un

exemple de ces écosystèmes artificiels, qui est le système Prédateur-Proie, connu sous le nom du modèle de Lotka-Volterra.

Le chapitre 2 consiste à étudier les stratégies de résolution des problèmes complexes. Il se focalise sur les principes des algorithmes évolutionnaires (les stratégies d'évolution, les algorithmes génétiques, la programmation évolutionnaire et la programmation génétique) ; ainsi que la nouvelle extension de ces algorithmes, qui est les algorithmes co-évolutionnaires, (inspirés des relations de co-évolution entre les individus d'une population ou ceux des populations différentes au sein d'un écosystème). Une comparaison entre les deux types d'algorithmes sera présentée par la suite. Deux algorithmes PSO et PPO (l'objectif de notre travail) seront détaillés à la fin de ce chapitre.

Le chapitre 3 est consacré au traitement d'image. D'abord, les principes de base de l'imagerie numérique sont présentés. Ensuite, les étapes du processus de traitement d'images sont décrites, représentant l'ensemble des opérations fondamentales pouvant être appliquées sur l'image pour des buts différents. Enfin, le chevauchement entre le traitement d'images et les algorithmes évolutionnaires et ceux co-évolutionnaires est discuté, en présentant quelques recherches réalisées dans ce cas.

Le chapitre 4 décrit l'application des deux algorithmes PSO et PPO au réhaussement de contraste dans les images en niveau de gris. Un ensemble d'images, en niveau de gris, est utilisé afin de prouver l'efficacité de cette application. Les résultats, obtenus par les deux algorithmes, sont comparés avec ceux obtenus en appliquant l'une des premières méthodes utilisées pour le réhaussement d'images, qui est la méthode d'égalisation d'histogramme.





Chapitre 1: Les Écosystèmes

1. Introduction

Les relations entre les individus d'une population ou des populations différentes au sein d'un écosystème sont une source d'inspiration pour la résolution des problèmes complexes, où les méthodes exactes ne peuvent pas aboutir à des solutions. Pour cette raison, on a choisi d'expliquer d'abord, dans ce chapitre, le concept d'écosystème naturel, ses composants, sa dynamique évolutionnaires ainsi que les relations entre ses composants. Ensuite, on présente comment ces systèmes naturels complexes sont simulés et sont devenus une source d'inspiration pour plusieurs travaux en mathématique et informatique. Enfin, on va présenter un des écosystèmes largement utilisés dans les mathématiques ainsi qu'en informatique, entre autre le système prédateur-proie, qui est la base de notre travail.

2. Les écosystèmes naturels

2.1 L'écologie

Le domaine de l'écologie a vécu ces dernières années quelques évolutions. Le mot aurait été employé pour la première fois en 1858 par un écrivain naturaliste américain (dont l'histoire n'a pas retenu le nom), mais généralement, c'est au biologiste allemand Ernest Haeckel (1866) qu'on attribue la première introduction du mot "*Ecologie*" dans le langage scientifique [Web 1]. Haeckel définit ce terme comme suit :

« Par *Ecologie* nous entendons la science des relations des organismes avec le monde extérieur, dans lequel nous pouvons reconnaître d'une façon plus large les facteurs de la "lutte pour l'existence". Ceux ci sont en partie de nature inorganique. Sous le nom de conditions d'existence, nous comprenons aussi l'ensemble des relations des organismes les uns avec les autres, relations soit favorables soit défavorables » [Davi 2002]

Étymologiquement, le mot "écologie" dérive des racines grecques "OIKOS" qui signifie "habitat" et "LOGOS" qui signifie "la science", donc il se traduit littéralement par "Science de l'habitat"[Web1].

Contrairement à l'éthologie qui ne s'intéresse qu'au comportement, l'écologie pose donc la relation au milieu comme constitutive de la discipline. Comme les organismes, populations et peuplements modifient le milieu, on parlera de l'écosystème, système



d'interactions entre l'ensemble des populations et de celles-ci avec leur milieu. L'écologie se place dans le cadre de la théorie générale des systèmes que ses chercheurs [Le Moigne 1989] présentent comme une théorie de l'interaction entre des parties d'où émergent des propriétés : « *le tout est plus que la somme des parties* » [Bousquet 2001].

L'écologie est subdivisée en trois branches: l'autoécologie, la synécologie et la dynamique des populations.

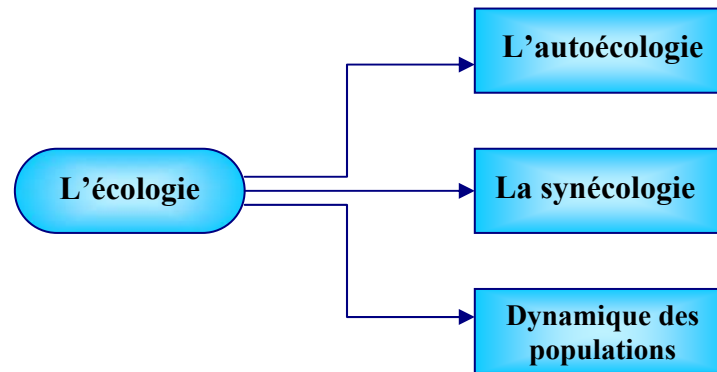


Figure 1.1. Les branches de l'écologie

1. **L'autoécologie** : qui étudie l'influence des facteurs extérieurs sur un individu. Ces facteurs sont de deux sortes : abiotiques, indépendants des êtres vivants (structure et porosité du sol, vitesse du courant d'une rivière, la température, l'éclairement, le vent, etc.), et biotiques, liés aux êtres vivants (étudier les relations entre les organismes vivants dans un écosystème) [Web 1].
2. **Synécologie** : dont l'objet est l'étude des écosystèmes et des systèmes écologiques d'ordre supérieur, y compris la biosphère dans son ensemble [Web 2].
3. **Dynamique des populations** : On obtient un système dynamique quand on a des boucles d'actions et de rétroactions. Notons qu'un système peut évoluer sans être dynamique. En effet, il peut changer l'état du fait de l'évolution du milieu extérieur, ces changements d'états ne sont pas alors dus aux interactions du système. Un cas très explicite de système dynamique est le système prédateur-proie de Lotka Volterra, où la population des proies et celle des prédateurs évoluent au cours du temps et ce sans aucune hypothèse sur l'évolution du milieu extérieur [Davi 2002].

On peut voir aussi l'écologie comme l'étude de la distribution et de l'abondance des organismes. Les questions qu'on se pose se retrouvent à des niveaux d'organisation de plus en plus vastes [Web 3]:

- Les ***individus*** : l'autoécologie ou écologie physiologique se penche sur les aspects comportementaux, physiologiques et morphologiques des réactions d'un organisme aux conditions physiologiques de son milieu (les limites de tolérance des organismes aux stress écologiques déterminent en effet où ils peuvent vivre).
- La ***population*** : groupe d'individus d'une même espèce vivant dans une aire géographique donnée à un moment précis.



- La **communauté** : tous les organismes qui habitent dans une aire donnée et qui sont un assemblage de populations de différentes espèces. On y étudie la prédation, la compétition et d'autres interactions influant sur la structure et l'organisation de l'ensemble.
- L'**écosystème** : ensemble formé par les facteurs abiotiques et la communauté d'une aire donnée.
- Les **biomes** : écosystèmes terrestres ou aquatiques caractérisés de grandes zones biogéographiques qui sont soumises à un climat particulier.
- La **Biosphère**: ensemble des écosystèmes de la planète. La partie de la Terre où l'on trouve la vie.

Claude Villeneuve [Web 4] présente trois types de courbe de survie illustrant l'évolution d'un groupe d'organismes vivants:

- **Type 1** : faible mortalité infantile augmentant progressivement parés la moitié de l'espérance de vie
- **Type 2** : taux de mortalité constant.
- **Type 3** : fort taux de mortalité infantile et survie des individus restant jusqu'à la limite de l'espérance de vie.

2.2 L'écosystème et ces composants

Pour comprendre le concept d'un écosystème, qui est un système écologique, on doit d'abord comprendre ces composants. Un écosystème est un ensemble d'une complexité variable, composé d'individus et de règles qui ont des rôles particuliers au sein de cet ensemble, dont l'équilibre et l'évolution dépendent de ses composants [Web 5].

L'écosystème est une notion dynamique qui représente l'ensemble formé par les conditions physico-chimiques du milieu et l'ensemble des organismes qui y vivent et y entretiennent des relations les uns avec les autres et avec le milieu. L'écosystème est un ensemble formé de la communauté d'organismes qui interagissent dans un milieu donné (**la biocénose**), et des conditions physico-chimique de ce milieu (**le biotope**) [Web 4].

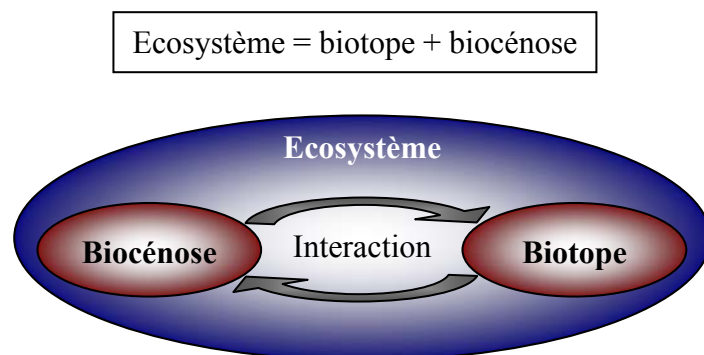


Figure 1.2. Les composants d'un écosystème.



En d'autres termes, L'écosystème est une unité fonctionnelle en écologie, c'est-à-dire une entité qui évolue en permanence et de manière autonome sous l'influence des flux d'énergie et de matière qui la traverse. Cette unité peut être décomposée en plusieurs niveaux hiérarchiques d'organisation. Nous pouvons considérer que le niveau le plus bas en écologie est l'individu et le niveau le plus haut est l'écosystème [Duboz 2004].

Une des caractéristiques essentielles des écosystèmes naturels est l'adaptabilité. Les organismes vivants s'adaptent aux changements des conditions de leur environnement en modifiant leur physiologie ou leur comportement à l'intérieur des limites qui sont propres à leur capacité individuelle. Cette capacité détermine des formes et fonctions qui définissent le phénotype. Les organismes sont adaptés les uns aux autres et développent des stratégies pour mieux occuper une niche écologique à un niveau trophique déterminé [Web 4].

Chaque biocénose dépend de son biotope, et réciproquement le biotope est influencé par la biocénose [Web 1] :

- » **L'action** est l'influence exercée par le biotope sur la biocénose. Il s'agit de la température, de la nature du sol, etc., responsables de l'adaptation, de la conservation, de la sélection ou de la régulation des espèces.
- » **La réaction** est l'influence exercée par la biocénose sur son biotope. Elle se manifeste par la destruction, l'édification ou la modification de celui-ci (ex. tassement du sol sous l'effet du pâturage, aération du sol par les lombrics, fixation des dunes par les oyats).
- » **La coaction** est l'influence que les organismes exercent les uns sur les autres.

Les organismes vivant peuvent produire énormément des descendants. Ces descendants ont fondamentalement les mêmes besoins, donc ils sont en compétition pour se procurer les matériaux est l'énergie qui leur sont nécessaires pour leur survie. Les plus aptes, au sein de l'écosystème, survivent [Web 4].

2.3 Quelques principes écologiques

Les principes écologiques venant des écosystèmes peuvent être transformés en principe d'organisation. Voyons comment sont décrits ces principes en ce basant sur le travail d'alphabetisation écologique de l'Institut Elmwood [Catalao 2002] :

1. **Interdépendance** – Tous les membres d'un écosystème sont reliés dans une toile de relations où tous les processus vitaux dépendent les uns des autres. Le succès du système comme un tout dépend du succès de ses individus, alors que le succès de chaque membre dépend du succès de tout le système.
2. **Durabilité** – La survie, à long terme de chaque espèce dépend d'une base de ressources limitée.
3. **Circularité** - L'interdépendance, entre les membres d'un écosystème, comprend des échanges de matière et d'énergie dans des cycles continus. Ces cycles écologiques agissent comme des circuits de régénération.
4. **Flux d'énergie** – L'énergie solaire, transformée en énergie chimique par la photosynthèse des plantes, commande les cycles écologiques.



5. **Association** – Tous les membres d'un écosystème participent à une interaction subtile, par l'intermédiaire de la compétition et de la coopération, qui comprend des formes associatives innombrables.
6. **Flexibilité** – Fonctionnant comme des circuits de régénération, les cycles écologiques montrent une tendance à la flexibilité, caractérisée par les fluctuations de ses variables.
7. **Diversité** – La stabilité d'un écosystème dépend beaucoup du niveau de complexité ou de diversité de son réseau de relations.
8. **Co-évolution** – La plupart des espèces d'un écosystème co-évoluent à travers l'interaction, la production et l'adaptation mutuelle.

2.4 Caractéristiques des populations

Les populations sont caractérisées par plusieurs caractéristiques, dont les plus importantes sont [Web 6] :

- **la répartition spatiale:** sur une surface donnée, les individus peuvent être répartis de différentes façons :
 - uniforme
 - contagieuse
 - au hasard

La répartition uniforme est très rare dans la nature mais fréquente dans les milieux artificiels (plantation). On observe une répartition au hasard chez des espèces qui n'ont aucune tendance à l'agrégation (ex: plantes supérieures pour lesquelles la dispersion des semences est assurée par le vent). Le type de répartition le plus fréquent est la répartition contagieuse, c'est à dire par groupe.

- **la densité** : c'est le nombre d'individus d'une espèce donnée par unité de surface. On peut la mesurer par comptage direct des individus ou l'évaluer par comptage indirect (terriers → lapin) sur toute la surface ou par échantillonnage sur une portion de celle-ci.
- **la croissance** : c'est la variation numérique par unité de temps des individus au sein de la population. Elle dépend essentiellement de la natalité et de la mortalité, mais aussi de l'émigration et de l'immigration.

2.5 Diversité des écosystèmes

Il existe trois niveaux de biodiversité dans les écosystèmes qui sont:

- La diversité génétique
- La diversité des espèces
- La diversité des communautés, au sein des écosystèmes



Des trois types de biodiversité, la diversité des habitats ou des communautés est sans doute la plus difficile à définir, car il n'existe pas de méthode de classification des écosystèmes. On peut toutefois estimer leur diversité en tenant compte, par exemple, de leur distribution géographique, ou du nombre d'espèces qu'ils renferment [Web 4].

Il existe ainsi quelques critères généraux de classification des écosystèmes: le climat, la biogéographie, la végétation existante. Ces critères permettent d'appréhender la diversité globale, mais fournissent peu d'éléments de comparaison dans et entre les écosystèmes. C'est pourquoi la diversité des écosystèmes est souvent estimée à partir de la diversité des espèces : si toutes les espèces sont présentes en proportions équivalentes à l'intérieur d'un écosystème, on considère que cet écosystème est plus diversifié que si certaines espèces étaient prédominantes et d'autres minoritaires [Web 4].

2.6 La complexité des écosystèmes

L'écologie moderne a pour objet l'étude des êtres vivants dans leur milieu (y compris l'Homme) et a pour méthodes non seulement la méthode expérimentale mais aussi les méthodes de nombreuses autres sciences avec lesquelles elle travaille : mathématiques, chimie, paléontologie, géographie, économie, informatique... Elle a une dimension pratique qui en fait aussi une science appliquée avec une dimension politique et sociale. L'écologie peut être définie comme la science des écosystèmes [Duboz 2004].

Un écologue travaille à de nombreux niveaux d'abstraction pour représenter et comprendre la dynamique d'un écosystème. De plus, chaque niveau est constitué par un très grand nombre d'entités et de relations entre ces entités qui font des écosystèmes un archétype de système hiérarchique complexe [Duboz 2004].

La complexité des écosystèmes a été étudiée par plusieurs chercheurs dans le domaine de l'écologie. Les chercheurs de l'Institut Elmwood [Catalao 2002] conçoivent un environnement comme un système d'auto-organisation et d'auto-régulation où les animaux, les plantes, les micro-organismes et les substances inorganiques sont liés les uns aux autres, formant un réseau de relations. Dans ce réseau, interagissent des mouvements opposés et complémentaires, comme ceux de la dépendance et de l'autonomie entre ses composants. Il s'agit d'un système intégré où les totalités ne peuvent être réduites à la somme des parties. Les parties, de leur côté, sont aussi dotées de complexité fonctionnant comme des sous-systèmes qui s'auto-régulent selon le même modèle d'intégration et de complémentarité présent dans le système.

De ces relations complexes, où le tout et les parties sont indissociables, découle un ensemble de propriétés et de principes d'organisation qui sont les principes écologiques (présentés dans la section 2.3). Ceux-ci peuvent fournir une ébauche conceptuelle pour établir des liens entre les communautés écologiques et les communautés humaines. Toute les deux sont des systèmes vivants qui exhibent les mêmes principes de base de l'organisation. Il s'agit de réseaux qui sont fermés d'un point de vue de l'organisation, mais ouverts au flux d'énergie et de ressources ; leurs structures sont déterminées par leurs histoires de changements structurels; ils sont intelligents étant donné les dimensions cognitives inhérentes aux processus de vie [Catalao 2002].

Dans un essai sur la complexité des écosystèmes, Holling [Holling 1987] qui est un des écologues les plus cités dans le domaine de la gestion des ressources, définit trois concepts qui ont dominé la perception de la causalité dans les systèmes écologiques, avec pour



conséquence de définir les éléments d'une gestion des écosystèmes. Le premier est centré sur la notion d'équilibre (la *nature est constante*), le deuxième considère plusieurs états de stabilité (*nature engineered* ou *nature resilient*). Cette deuxième perception donne de l'importance aux dynamiques causées par la variabilité, par des événements qui apparaissent à l'échelle dite micro. Plutôt qu'un seul équilibre, le système est caractérisé par un ensemble d'attracteurs. Le troisième point de vue est celui du changement organisationnel (*nature evolving*). Ce point de vue s'intéresse à la connectivité des éléments du système, à leur organisation sur plusieurs échelles [Bousquet 2001].

Une référence théorique des écologues, sur ce dernier point de vue, est la théorie de la hiérarchie, introduite par Allen et Starr [Allen 1982] comme une théorie de l'observation de systèmes complexes. Ces systèmes complexes sont présentés comme les intermédiaires entre les systèmes à grands nombres pour lesquels une approche statistique est adaptée et les systèmes à petits nombres pour lesquels les approches mathématiques de type équations différentielles, qui manipulent la dynamique de quelques populations, sont pertinentes. Pour les auteurs les systèmes intermédiaires restent opaques à moins qu'on les modélise (observe) en tant qu'organisations hiérarchiques [Allen 1982]. Une organisation hiérarchique ne doit pas être vue comme un système de poupées gigognes mais plutôt comme des hiérarchies dynamiques enchevêtrées [Bousquet 2001].

Dans un récent ouvrage, Allen et Hoekstra [Allen 1992] expliquent que la compréhension d'un phénomène complexe en écologie implique de comprendre les interactions entre les différents niveaux : les organismes, les populations, les communautés, les écosystèmes, les paysages et les biomes. « *A part from organism and biosphere levels, there is plenty of room for entities from almost any type of ecological system to be contained within an entity belonging to any other class of the system* ». Les auteurs donnent l'exemple du phénomène de la rumination qui fait intervenir un écosystème sous le contrôle d'un organisme (le ruminant).

2.7 Co-évolution

L'une des parties les plus importantes dans l'environnement d'une organisation est les autres organisations. La co-évolution se produit quand, en s'adaptant à leur environnement, deux organisations ou plus évoluent ensemble. Pour "faire le meilleur" de leur habitat, des organisations se servent d'autres organisations en les mangeant, en vivant dans ou sur elles, et/ou en établissant une "association" avec elles. Les organisations co-évoluent avec beaucoup d'espèces en même temps, parce qu'un environnement inclut différents types d'organisations [Web 7].

Les espèces qui ont des exigences écologiques semblables ont tendance à évoluer l'une par rapport à l'autre dans le temps. Cette évolution favorise la réduction de la compétition et la spécialisation de chacun des groupes. Les populations de proies s'adaptent à la prédation, soit par leur démographie, soit par la sélection naturelle de moyens de défenses plus efficaces. Les prédateurs s'adaptent aux défenses de leurs proies [Web 4].

Les organisations doivent s'adapter pour former ces relations, juste comme elles doivent s'adapter à n'importe quelle autre partie de leurs environnements. Ces adaptations causent plus de changements sur les descendants que les organisations, qui sont mieux convenus de tirer profit de la relation, peuvent être des espèces entièrement différentes que leurs parents. Trois types de co-évolution sont : les relations prédateur-proie, les relations symbiotiques, et les relations parasites [Web 7].



2.8 Les interactions entre les populations

Diverses interactions ont lieu entre les individus d'une même population, ou encore entre ceux de populations différentes. Ces interactions peuvent prendre diverses formes dont les plus importantes sont la compétition, la prédation, le parasitisme et la symbiose. Les relations interspécifiques (effets sur deux populations A et B; 0 = pas d'effet; + = effet favorable; - = effet défavorable) sont classifiées comme suit [Borcard 2005]:

	Population A	Population B
Neutralisme	0	0
Commensalisme	+	0
Protocoopération	+	+
Symbiose	+	+
Compétition	-	-
Amensalisme	0(+)	-
Prédation	+	-
Parasitisme	+	-

Tableau 1. Les relations entre les individus.

2.8.1 Compétition

C'est l'usage commun d'une ressource limitée, dont l'abondance ne peut suffire aux deux populations concernées [Borcard 2005]. La compétition peut avoir deux formes : intra-spécifique et interspécifique. La compétition intra-spécifique est la concurrence s'exerçant entre plusieurs organismes lorsque la somme de leurs demandes en nourriture, en certains éléments minéraux, en eau, en sources énergétiques, en espace libre, etc., est supérieure à ce qui est réellement disponible. La compétition n'est pas exclusivement du domaine des relations intra-spécifiques, elle peut également exister entre des individus d'espèces différentes (on l'appelle dans ce cas la compétition inter-spécifique) [Web 1].

2.8.2 Prédation

Le prédateur est un organisme libre qui recherche une nourriture vivante; il tue sa proie pour s'en nourrir. Cette relation de type antagoniste entraîne à court terme la disparition de l'un des deux individus; seul le prédateur en tire bénéfice. [Web 1]. Quelques exemples des prédateurs et des proies sont : lion et zèbre, ours et poissons, et renard et lapin ...etc. Les mots " prédateur " et " proie " sont presque toujours employés pour signifier seulement les animaux qui mangent des animaux, mais le même concept s'applique également aux plantes:



Ours et baie, lapin et laitue, sauterelle et feuille (exemples de ces systèmes sont présentés dans la figure 1.3) [web 8].



Figure 1.3. Quelques exemples des écosystèmes prédateur-proie.

On peut distinguer trois types des prédateurs [Borcard 2005]:

- » **Les prédateurs vrais:** tuent leur proie: loups, lions; mais certains étendent la définition aux granivores, planctonophages;
- » **Les brouteurs:** prélèvent une partie de la proie, avec des dommages variables pour cette dernière. Ex.: herbivores, insectes piqueurs, sangsues...
- » **Parasitoïdes:** essentiellement des larves d'insectes (Diptères et Hyménoptères) qui sont pondus dans d'autres insectes et finissent par les tuer.

2.8.3 Parasitisme

Généralement, le terme parasite est employé lorsque l'organisme "exploiteur" est petit et qu'il vit (sur) ou (dans) l'hôte qui sert alors de source d'énergie et d'habitat [Web 1]. Exemple de cette relation est le mycorhize (du grec «mukês» pour champignon et «rhiza» pour racine) qui est l'association symbiotique d'un champignon avec les racines d'une plante (figure 1.4). Les champignons, vivant en symbiose avec les arbres de la forêt, jouent un rôle tout autant important pour l'écosystème forestier. Cette symbiose est bénéfique aussi bien à l'arbre qu'au champignon [Egli 2002].



Figure 1.4. Le champignon et l’arbre vivent en association symbiotique.

2.8.4 Symbiose ou Mutualisme

La symbiose, dans sa définition générale, est simplement la vie ensemble de différentes organisations. Souvent, dans l'utilisation étendue, le terme est employé pour se rapporter au cas spécial de la mutation où les symbionts (organisations dans le rapport symbiotique) sont mutuellement profitables [Watson 1999].

Un exemple d'un rapport symbiotique est celui de l'Oxpecker (un genre d'oiseau) et les rhinocéros ou le zèbre. La terre des Oxpeckers est sur les rhinos ou les zèbres, et ils mangent les coutils et d'autres parasites qui vivent sur leur peau. Les Oxpeckers obtiennent la nourriture et les bêtes obtiennent le contrôle des parasites. En outre, quand il y a danger, les Oxpeckers volent vers le haut et crient un avertissement, qui aide le symbiont [Web 9].



Figure 1.5. Relation de symbiose entre l’Oxpecker et le zèbre.

2.8.5 Autres

Elles existent d’autres relations entre les individus au sein d’un écosystème, qu’on peut décrire brièvement comme suit [Borcard 2005] :

- » **Neutralisme:** absence de relation, indifférence. Ex.: chevreuil et Musaraigne.



- » **Commensalisme:** C'est la relation entre deux individus d'espèces différentes dont l'un profite de l'autre (source de nourriture, support) sans toutefois lui nuire ou lui apporter un avantage [Web 1]. Ex.: poisson rémora attaché par une ventouse céphalique au requin; il est transporté par ce dernier et bénéficie de ses restes de nourriture.
- » **Proto-coopération:** symbiose facultative; les deux populations bénéficient de leur présence mutuelle, mais peuvent s'en passer. Ex.: relation entre certains crabes et cœlentérés (anémones) qu'ils portent sur leur dos: le crabe bénéficie de la protection de l'anémone qui devient mobile et peut partager la nourriture du crabe.
- » **Amensalisme** (= allélopathie chez les végétaux): élimination d'une espèce par une autre qui sécrète une substance toxique.

3. Les écosystèmes artificiels

Les écosystèmes artificiels sont les résultats de la simulation des écosystèmes naturels. On nomme simulation la démarche scientifique qui consiste à réaliser une reproduction artificielle, appelée **modèle**, d'un phénomène réel que l'on désire étudier, à observer le comportement de cette reproduction lorsqu'on en fait varier certains paramètres, et à en induire ce qui se passerait dans la réalité sous l'influence de variations analogues. La démarche de simulation passe donc par trois étapes distinctes (Figure 1.6): l'étape de modélisation, qui consiste à construire le modèle du phénomène à étudier, l'étape d'expérimentation, qui consiste à soumettre ce modèle à un certain type de variations, et l'étape de validation, qui consiste à confronter les données expérimentales obtenues avec le modèle à la réalité [Drogoul 1996].

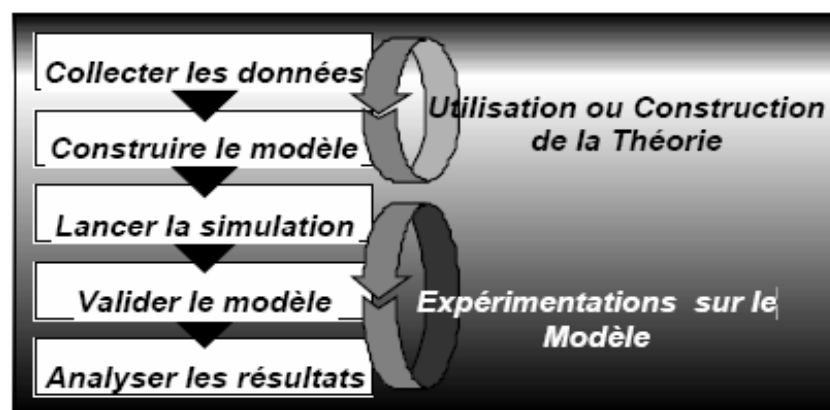


Figure 1.6. Les étapes du processus de simulation.

Des écosystèmes artificiels ont été définis dans la littérature [Olson 1995] comme système dans lequel un seul environnement est peuplé par des entités indépendantes (par exemple des agents) représentant un certain niveau d'organisation. Ils sont une classe du modèle de la vie artificiel, qui est une abstraction des systèmes naturels vivants, préservant



certaines caractéristiques de ces systèmes. Les agents dans un écosystème agissent localement l'un sur l'autre et avec leurs environnements [Kleber 1997].

Modeler de tel système et de son exécution informatique a fait face à quelques difficultés. Des modèles mathématiques sont limités aux cas analytiques pour lesquels il y a des solutions menables. En conséquence, seulement des modèles de système de la simplicité extrême en tant que, par exemple, des équations simples avec très peu de variables, peuvent être résolus. Par exemple, même un système simple représentant une relation de prédateur-proie eut avoir comme conséquence un système des équations non-linéaires l'où il n'y a aucune solution analytique explicite [Kleber 1997].

3.1 Un peu d'histoire

L'homme a songé également à créer des écosystèmes artificiels dont le support était la machine et les composants divers programmes qui ont chacun un rôle. Cet historique est tiré du travail d'Antoine [Web 5] sur l'apport de la biologie en informatique.

Parmi les premiers écosystèmes créés en informatique, il ne s'agissait souvent que de programmes interagissant entre eux en faisant évoluer un environnement, comme le Jeu de la Vie de Conway qui se devait de modéliser une population d'êtres vivants à l'aide de règles simples. Ce Jeu de la Vie était un réseau d'automates, où chaque automate représentait un être vivant qui comportait deux états possibles : mort ou vivants. Chaque automate changeait d'état suivant l'état de ses automates voisins, sachant que chacun influençait les autres, ainsi on arrivait à modéliser un écosystème simple ce qui fut exceptionnel pour l'époque. En effet créer un écosystème artificiel demande un effort soutenu pour un résultat peut démonstratif en général, et on utilise des outils pointus et complexes comme les algorithmes génétiques, les agents multiples, les réseaux neuronaux. Les écosystèmes artificiels sont en fait des programmes résultants de tous les outils et méthodes informatiques les plus pointus, et ne sont pas eux même généralement des outils de résolutions pour des problèmes spécifiques.

Mais l'intérêt de ces écosystèmes est intéressant par le fait qu'en plus de simuler et copier un écosystème naturel, comme ils le font à l'habitude, ils vérifient des lois bien naturelles ce qui montre leur rapprochement important avec la biologie.

Un intérêt se porte également de plus en plus vers l'intelligence collective qui apparaît dans certains écosystèmes, car en effet des composants de l'écosystème ont la possibilité dans certain cas de communiquer avec d'autres composants et cela parfois de manière importante au sein de l'écosystème, ce qui établit alors une intelligence collective similaire à certains insectes vivant en communauté tel que les fourmis. Ce type d'intelligence collective, apparut dans des écosystèmes et qui fut conçu dans des universités, pris de l'intérêt pour certaines entreprises comme I.B.M. qui virent dans cela un moyen d'utiliser des agents informatiques communicants entre eux et travaillant ensemble sur l'élaboration d'une tâche, ainsi grâce à leur intelligence collective, on pouvait observer une façon intéressante de la résolution d'un problème.

Mais il faut également noter que certains programmes complexes, qui furent réalisés dans un but bien différent de celui de créer un écosystème artificiel, se sont avérés rapidement comme de vrais écosystèmes en établissant un équilibre et un certain ordre au sein de leur fonctionnement. Certains informaticiens commencent même à parler du futur développement d'Internet comme un écosystème virtuel et mondial composé de ses agents, de ses virus, le tout essayant de former un équilibre.



3.2 Simulation SMA et écologie

La simulation de systèmes complexes à partir de systèmes collectifs artificiels repose sur un principe d'analogie structurelle consistant à transcrire dans des agents artificiels les objets du système source. En construisant ainsi un ensemble d'agents en interaction reproduisant individuellement la dynamique locale des entités systémiques, on obtient une image artificielle du système source, qui peut soit valider des hypothèses locales par observation des processus émergents, soit servir de laboratoire virtuel pour tester des hypothèses comportementales. La condition de réalisation et d'utilisation d'une telle simulation est que le support matériel à l'implémentation des agents respecte le parallélisme intrinsèque (ou la séquentialité fine) des processus réels simulés [Web 10].

Les systèmes multi-agents apparaissent comme une métaphore efficace pour décrire des composants en interaction dans un système, dès lors que l'on cherche des modèles explicatifs pour comprendre des phénomènes globaux. C'est pourquoi nous voyons une utilisation croissante de ce type de modèles en écologie où la complexité est une caractéristique essentielle qui doit être appréhendée pour comprendre la dynamique des écosystèmes. Nous pouvons dire que les SMAs sont l'antithèse des modèles phénoménologiques classiques en faisant de la complexité leur moteur de construction [Duboz 2004].

Du point de vue des avancées en écologie, Grimm [Grimm 1999] a classifié l'utilisation des modèles basés individus suivant deux motivations. La première motivation est dite pragmatique : elle s'adresse à des problèmes qui ne pourraient pas être étudiés avec des modèles à variable d'état, comme par exemple expliquer une structure de poissons [DeAngelis 1979] ou la dynamique d'une hiérarchie. Ce type d'application a donné lieu à beaucoup de modèles pour comprendre l'hétérogénéité d'une structure de population animale ou végétale. La deuxième motivation est dite paradigmatique: elle considère que les modèles à variables d'état sont pertinents mais montrent des insuffisances pour étudier les questions classiques de l'écologie à savoir la régulation, la stabilité ou la résilience [Bousquet 2002].

Parallèlement à la dynamique scientifique présentée ci-dessus et sans qu'il y ait beaucoup d'interactions - seuls quelques auteurs comme Hogeweg ou Bonabeau publient de nombreux développements, qui ont été effectués par des chercheurs informaticiens qui invoquent une référence à l'écologie. Des statistiques ont été effectuées montrant que dans une très grande majorité des cas, la méthode utilisée est celle de la simulation. Quelques recherches abordent la question de la résolution de problèmes, appelée résolution collective de problèmes [Théraulaz 1994] dans le contexte des SMAs. À partir de travaux en éthologie sur les capacités de résolution des problèmes par des insectes sociaux [Deneubourg 1989] sont apparus de nombreux travaux dont les problématiques sont transposables à des problématiques logicielles ou en robotique [Bousquet 2002]. On trouvera une excellente revue dans [Drogoul 1996] qui développe une approche originale nommée: éco-résolution.

Un exemple des écosystèmes artificiels basé sur les SMAs est le jeu de PENGI conçu par Drogoul [Drogoul 1996]. Ce jeu se déroule dans un micro-monde peuplé d'un pingouin, d'abeilles, de diamants et de blocs de glace (voir Figure 1.7). Le pingouin, contrôlé par le joueur, se déplace dans un labyrinthe dont les parois sont faites des blocs de glace qui peuvent être poussés afin de libérer un chemin ou d'écraser les individus qui se trouvent de l'autre côté. Le but du jeu est de faire récolter par le pingouin un certain nombre de diamants, tout en lui évitant d'être piqué par les abeilles, ou écrasé par un bloc de glace poussé par l'une d'entre elles. Celles-ci ont un déplacement semi-aléatoire, ce qui signifie qu'elles suivent le pingouin quand elles en sont assez proches, mais qu'elles peuvent tout à coup décider de l'abandonner, la probabilité pour qu'elles le fassent étant fixée une fois pour toutes au commencement de la



partie. Un bloc de glace ne peut être poussé que dans les directions où il n'est pas immédiatement adjacent à un autre bloc. Après avoir été poussé, il glisse dans la direction choisie jusqu'à ce qu'il heurte un autre bloc de glace (ou les bords de l'écran).

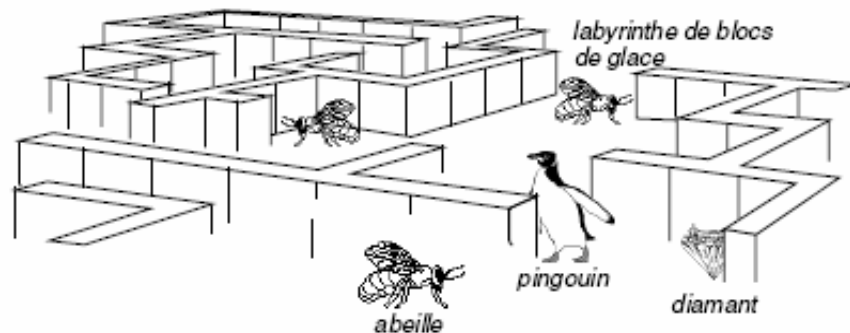


Figure 1.7. Une représentation en perspective de PENGI, du point de vue du pingouin.

3.3 Quelques plates-formes de simulation des écosystèmes

Les applications présentées sont le plus souvent élaborées avec un langage Orienté-Objet. Certaines d'entre elles utilisent des plates-formes. On peut classer ces dernières en trois types:

- **Les plates-formes génériques** : aujourd'hui, quelques-unes d'entre elles sont régulièrement citées dans des applications environnementales. Swarm [Minar 1996] est l'outil privilégié de la communauté américaine et des chercheurs en Vie Artificielle. De nombreuses applications ont été développées à partir de Swarm qui existe aujourd'hui en plusieurs langages (Java, Objective-C). Des outils génériques ont même été créés à partir de Swarm. La plate-forme Gammes [Marcenac 1997] a été utilisée pour quelques applications sur des écosystèmes. Alors que la plateforme MaKis implémente la notion de rôle [Gutknecht 1997] et Geamas propose une implémentation du groupe. On peut également citer Oris [Ballet 1998].

- **Les plates-formes orientées écosystème** : Ce type de plates-formes parmi lesquelles on peut ranger l'outil Ecosim ou encore les plates-formes Sugarscape ou Cormas [Bousquet 1998], proposent des utilitaires pour simuler des écosystèmes ou des problèmes de gestion des ressources. Grilles spatiales, utilitaires de simulation pour méthodes de type Monte-Carlo, liens vers des logiciels (SIG, bases de données) font de ces outils des ateliers pour l'implémentation de divers systèmes écologiques. Des algorithmes ou des structures sont prévus pour implémenter le lien des agents avec leur environnement et on trouve aussi des éléments pour l'organisation de sociétés d'agents (marchés, enchères, mécanismes de prédation, ...).

3.4 Exemple d'écosystème artificiel : le modèle "Prédateur/Proie"

Plusieurs écosystèmes artificiels sont développés, par des chercheurs en écologies, biologies, mathématique et en informatique, pour des raisons différentes. Un des exemples les plus connus de modélisation mathématique est celui de Lotka et Volterra [Volterra 1926].



Dans les années 1920, le mathématicien italien Vito Volterra a proposé un modèle d'équation pour décrire la dynamique de population de deux espèces agissantes l'un sur l'autre, un prédateur et sa proie. Il a espéré d'expliquer l'augmentation observée des poissons prédateurs (et la diminution correspondante des poissons de proie) en mer adriatique pendant la guerre mondiale I. de tels modèles mathématiques ont longtemps prouvé utile en décrivant comment les populations changent durant le temps. Les données concernant les divers taux de croissance, de mort, et d'interaction des espèces mènent naturellement aux modèles impliquant des équations [Chauvet 2002].

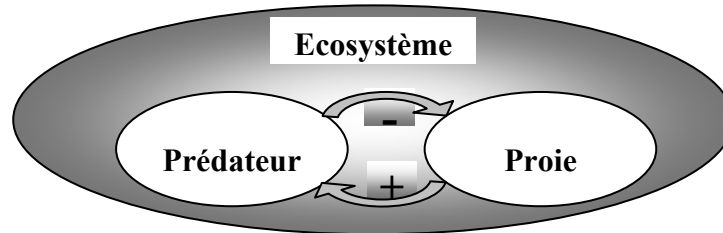


Figure 1.8. L'interaction entre la population des Prédateurs et celle des Proies

Indépendamment, aux Etats-Unis, les mêmes équations étudiées par Volterra ont été dérivées par Alfred Lotka, pour décrire une réaction chimique hypothétique, dans laquelle les concentrations chimiques oscillent [Chauvet 2002]. Ce modèle a été étudié en détail à partir du célèbre livre de Volterra [Volterra 1931].

Le modèle de Lotka-Volterra reçut par la suite des modifications et il est, encore aujourd'hui, à la base de nombreux travaux théoriques et appliqués. Ce modèle s'intéresse aux relations de prédateurs et cherche à prédire comment les effectifs des populations de proie et de prédateurs vont évoluer si elles se rencontrent [Goreaud 2004].

3.4.1 Les équations du modèle

Le modèle de Lotka-Volterra est un système ordinaire qui exprime par deux équations différentielles les taux de croissance respectifs d'une population de prédateurs et d'une population de proies partageant un même environnement. Ces deux équations différentielles sont expliquées comme suit :

Soit :

- * N = densité des proies,
- * P = densité des prédateurs,
- * r_1 = taux intrinsèque d'augmentation de population de proie,
- * k_1 = la capacité des proies à échapper aux prédateurs
- * r_2 = taux de reproduction de prédateurs par une proie mangée,
- * k_2 = la capacité des prédateurs à attraper des proies.

La croissance de la population de proies (N), en cas d'absence de prédateurs (P), est calculée par:



$$\frac{dN}{dt} = r_1 N$$

En présence de prédateurs (P), l'équation devient:

$$\frac{dN}{dt} = (r_1 - k_1 P) N$$

La croissance de la population de prédateurs (N) en cas d'absence de proies (P) est exprimée par:

$$\frac{dP}{dt} = -r_2 P$$

En présence de proies (P), l'équation sera:

$$\frac{dP}{dt} = (-r_2 + k_2 N) P$$

Les deux points d'équilibre de ce système sont : N_{eq} et P_{eq} , la dynamique de ce système est représentée dans la figure suivante [Francour 2005].

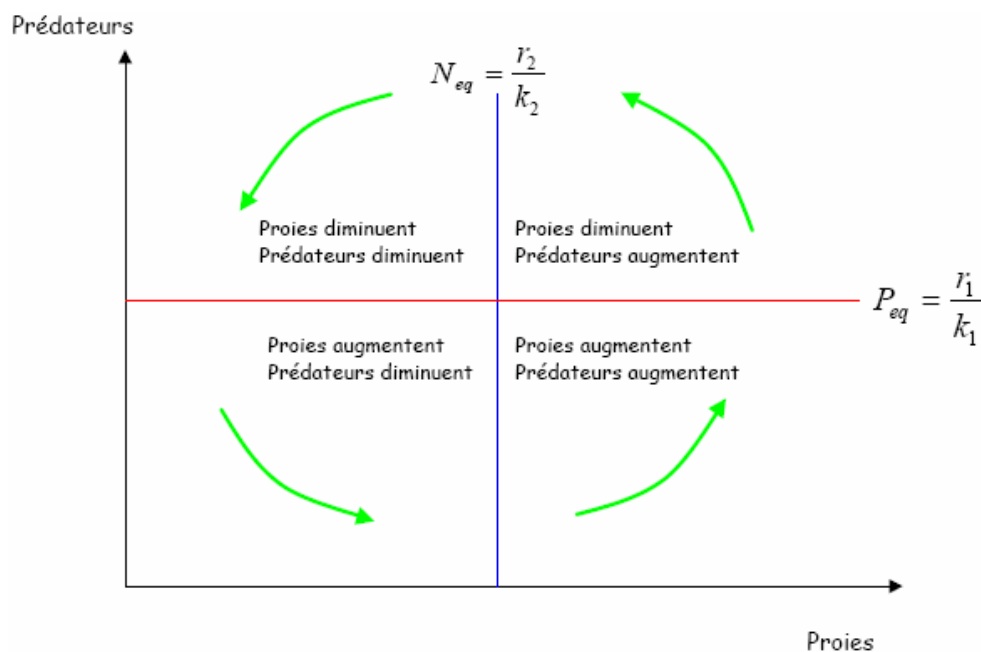


Figure 1.9. La dynamique du système proie-prédateur

Figure 1.10 décrit la famille des trajectoires du système prédateur-proie dans le plan x-y circulant autour du point (1.1) avec les paramètres $r_1 = k_1 = r_2 = k_2 = 1$ chacun peut examiner son savoir biologique en déterminant pourquoi les trajectoires circulent dans le sens contraire des aiguilles d'une montre.

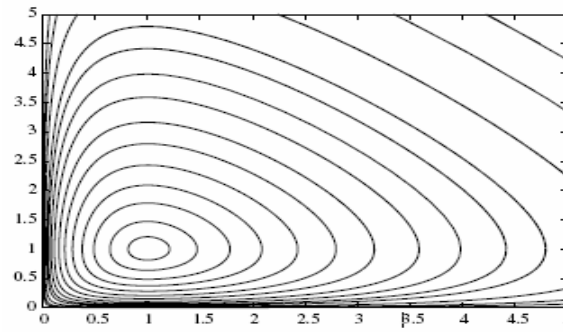


Figure 1.10. Une famille des orbites fermées dans le plan x-y.

La Figure 1.11. Trace du comportement d'une solution particulière avec le temps (avec les paramètres $r_1 = k_1 = r_2 = k_2 = 1$) [Chauvet 2002].

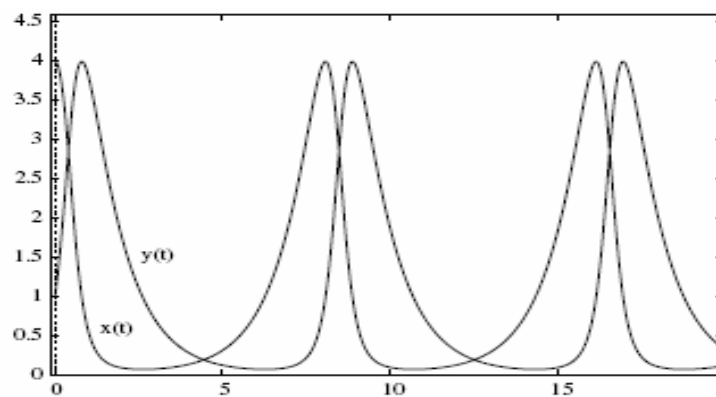


Figure 1.11. Une solution avec les conditions initiales $(x, y) = (4, 1)$

Le modèle de prédateur-proie prévoit un comportement périodique dans les populations des deux espèces avec une période commune. Ce comportement est vu dans les rapports historiques de la compagnie Bay de Hudson (Figure 1.12), qui a enregistré le nombre annuel de peaux des lièvres (proie) et du lynx (prédateur) rassemblé de 1845 –1935 [Elton 1942].

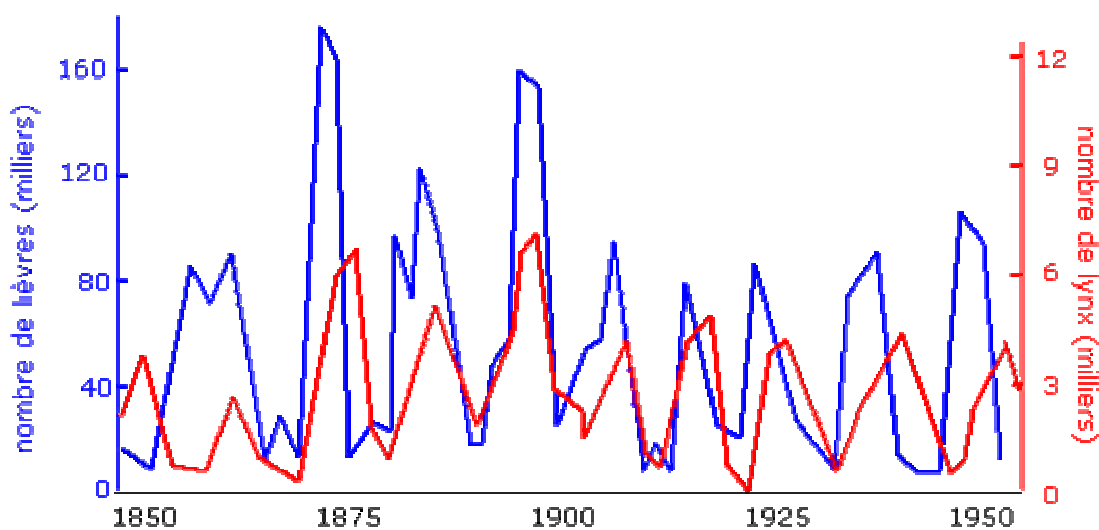


Figure 1.12. Résultat du rapport historique de la compagnie Bay de Hudson.



Les équations différentielles de Lotka-Volterra de type Prédateur-Proie ont joué un rôle important dans la biologie théorique de population. Cependant, en examinant des cas plus compliqués où il y a plus d'un type de nourriture disponibles pour le prédateur, ou où il y a plus de type de prédateurs mangeant un type particulier de proie, les équations classique de Lotka-Volterra ne raffette pas les conséquences du comportement adaptatif des individus. Les modèles standard de la théorie optimale de fourragement tels que les modèles *Prey-Patch* supposent que les animaux sont les optimiseurs parfaits maximisant une quantité (par exemple, le taux de prise d'énergie) qui est directement liée au « fitness » [David 1999].

Le modèle de Volterra-Lotka a été largement étudié, car il est d'une part relativement simple, et d'autre part il est considéré comme la base des systèmes d'interaction prédateur-proie. L'évolution décrite par ce système n'a cependant que peu de rapport avec la réalité biologique. Seules les oscillations en opposition de phase des populations proie et prédateur s'approchent d'un comportement naturel [web1].

3.4.2 Réponse fonctionnelle et Réponse numérique

Holling [Holling 1959] a étudié la prédation de petits mammifères sur les tenthrèdes de pin, et lui a constaté que les taux de prédation ont augmenté avec l'augmentation de la densité de population de proie. Ceci a résulté de 2 effets: (1) prédateur a augmenté son taux de consommation une fois exposé à une densité plus élevée de proie, et (2) la densité des prédateurs a augmenté avec l'augmentation de la densité de proie. Holling a considéré ces effets en tant que 2 genres de réponses de population du prédateur à la densité de proie: (1) la réponse fonctionnelle et (2) la réponse numérique [Web 11].

- **La Réponse Fonctionnelle**

Holling [Holling 1959] a suggéré un modèle de la réponse fonctionnelle qui reste le plus populaire entre les écologistes. Il a considéré 3 types principaux de réponse fonctionnelle:

- » **La réponse fonctionnelle du type I** Les prédateurs capturent leurs proies à un taux proportionnel à celui de leurs rencontres avec elles (taux de capture proportionnel à la disponibilité de proies), et ce jusqu'à satiété. Jusqu'à ce point, la mortalité des proies ne dépend donc pas de la densité (de prédateurs) [Borcard 2005].
- » **La réponse fonctionnelle du type II** est la plus typique. Le taux de recherche de proie est constant. La mortalité de proie diminue avec la densité de proie. Les Prédateurs de ce type causent une mortalité maximale à la basse densité de proie [Web 11].
- » **La réponse fonctionnelle du type III** se produit dans les prédateurs qui augmentent leur activité de recherche avec l'augmentation de la densité de proie. La mortalité augmente d'abord avec la densité croissante de proie, et diminue ensuite [Web 11].

- **Réponse Numérique**

La réponse numérique signifie que les prédateurs deviennent plus abondants à mesure que la densité de proie augmente. Cependant, le terme " réponse numérique " est plutôt embrouillant parce qu'il peut résulter de 2 mécanismes différents [Web 11]:



- Augmentation du taux de reproduction des prédateurs quand les proies sont abondantes (réponse numérique intrinsèquement)
- Attraction des prédateurs aux agrégations de proie ("réponse agrégationnelle")

Le taux de reproduction de prédateurs dépend naturellement de leur taux de prédation. Plus la proie est consommée, plus l'énergie du prédateur peut être assignée pour la reproduction. Le taux de mortalité réduit également avec la consommation accrue de proie.

Le modèle le plus simple de la réponse numérique du prédateur est fondé sur l'hypothèse que le taux de reproduction de prédateurs est proportionnel au nombre de proie consommé. C'est comme la conversion de la proie en nouveaux prédateurs.

3.4.3 Exemples d'application du modèle Prédateur/Proie

Hawick et al. [Hawick 2004] reporte sur des modèles émergents en spirale produits d'un modèle basé sur les règles d'automate de prédateur-proie (voir la figure 1.13). Ils discutent quelques techniques systématiques d'analyse appliquées aux modèles et supposent que leur origine se situe dans un comportement défensif qui résulte des interactions de prédateur-proie. Ils croient que les modèles émergents de ce modèle simple correspondent aux comportements semblables dans des situations du monde réel où deux groupes agissent l'un sur l'autre.

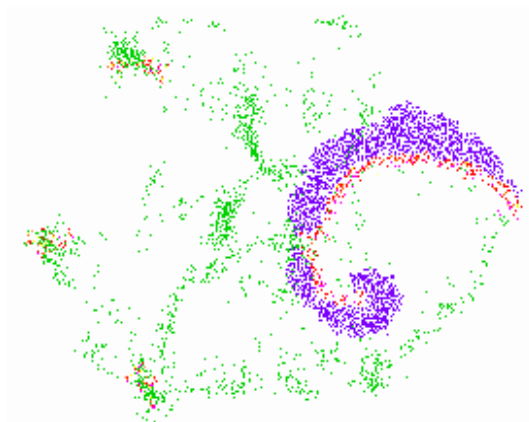


Figure 1.13. Visualisation de l'état de modèle de prédateur-proie.

Dans la figure 1.13, les Pixel foncés (bleus) représentent les proies ("lapins"). Les Pixels plus légers (rouges) représentent des prédateurs ("renards"). Les lapins se fondent sur leur nombre exact tout en se sauvant des renards attaquants. Les Pixel très légers (verts) montrent l'endroit des cadavres de renard quand ils sont morts de la famine.

Les règles évolutionnaires de ce système sont relativement simples. Les règles de proie sont:

1. Eloigner du prédateur s'il est adjacent;
2. Reproduire si une proie est adjacente et moins de 5 proies sont voisins;
3. Se déplacer vers une proie si la proie est plus proche que 20 unités spatiales;



4. Se déplacer vers une position adjacente aléatoirement choisie

Les règles de prédateur ou de " renard " sont comme suit:

1. Manger la proie si la proie est adjacente;
2. Se déplacer vers une proie si elle est plus proche que 80 unités spatiales et ce prédateur a faim;
3. Reproduire si un renard est adjacent et moins de 3 renards sont voisins;
4. Se déplacer vers un prédateur s'il est plus proche que 80 unités spatiales et ce prédateur n'a pas faim;
5. Se déplacer vers une position adjacente aléatoirement choisie.

À chaque étape, chaque animal exécute une règle dans l'ordre de priorité. Si la règle 1 est exécutée, toutes autres règles pour l'animal sont ignorées. Si la règle 1 ne peut pas être exécutée, la règle 2 est employée, et ainsi de suite [Hawick 2004].

Li et Sutherland [Li 2002] ont développé un algorithme génétique cellulaire simulant les interactions entre une population des prédateurs et celle des proies. Le modèle dépend sur la dynamique produite des interactions spatiale entre les deux populations pour maintenir la pression de sélection et aussi la diversité de population. Dans ce modèle, les proies représentent les solutions potentielles, se déplacent dans une grille de 2-dimensions, et reproduisent avec d'autres proies. La pression de sélection est exercée par le prédateur, qui se déplace également pour garder la proie dans le contrôle en enlevant la proie la plus faible dans leur proximité. Ce genre de pression de sélection conduit efficacement la population de proie pendant les générations successives d'augmenter la fitness. Leur étude préliminaire a montré que la dynamique d'interaction prédateur-proie joue un rôle important en maintenant une pression appropriée de sélection dans la population de proie, par conséquent produire des solutions convenablement adaptées de proie. Les résultats expérimentaux sont comparables ou meilleurs dans l'exécution que ceux de l'algorithme génétique standard, en les utilisant pour optimiser quatre fonctions utilisées comme benchmark. La représentation de leur modèle initiale est illustrée dans la figure 1.14.

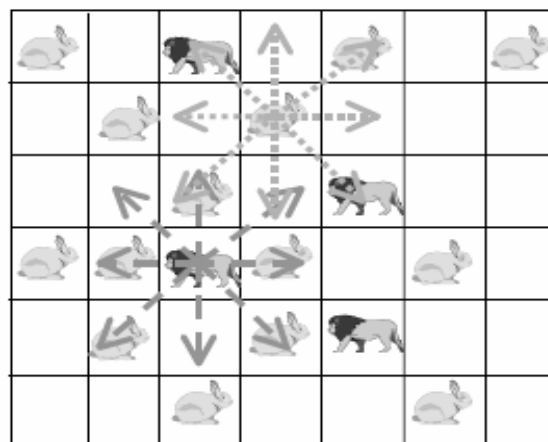


Figure 1.14. Distribution aléatoire des prédateurs et des proies à travers une grille de 2D.



Armetta et al. [Armetta 2004], ont réalisé un travail portant sur la gestion dynamique de flux dans une chaîne de production, à travers les mécanismes de stigmergie et d'auto-organisation. L'approche utilisée se base sur l'utilisation du paradigme multi-agents situés, exploitant dans sa dynamique un processus de combat selon la métaphore "Proie-Prédateur".

Dans une étude de Schmitt et al. [Schmitt 2004], le problème d'adapter la stratégie bien connue d'évolution dans le champ de l'optimisation multi-objectives est traité. La stratégie modifiée d'évolution a adapté des idées des modèles de prédateur-proie de l'écologie. En particulier, cette approche de Prédateur-Proie permet au praticien de basculer entre l'optimisation simple-objective et l'optimisation multi-objectives d'une façon très facile. Les expériences effectuées avec cinq fonctions d'essai ont prouvé que le modèle de prédateur-proie peut produire un bon ensemble de solutions diverses avec Pareto-front pour les fonctions d'essai convexe aussi bien que pour celles non convexe.

David et Křivan [David 1999] ont analysé le comportement de trois différents systèmes de Lotka-Volterra de Prédateur-Proie avec le comportement optimal de fourragement. Ils ont étudié un modèle prédateur-proie avec la nourriture alternative, un modèle de deux pièces rapportées avec les prédateurs mobiles et la proie résidente, et un modèle de deux pièces rapportées avec des prédateurs et des proies mobiles. Ils ont montré que, dans des exemples étudiés, le comportement optimal de fourragement change la qualité intrinsèque neutre de stabilité aux systèmes de Lotka-Volterra à l'existence d'un attracteur global lié.

Silva et al. [Silva 2003], ont proposé une nouvelle approche d'optimisation de fonction, basée sur l'optimisation d'essaim de particules (PSO : Particle Swarm Optimization). Dans leur approche, appelée PPO (Predator Prey Optimization), ils ont combiné les principes de base de PSO et une stratégie inspirée de la relation Prédateur-Proie au sein d'un écosystème. Une comparaison expérimentale est faite entre les résultats d'exécution de plusieurs variantes de PSO et la nouvelle approche (PPO). Ce travail est détaillé dans le chapitre suivant.

4. Conclusion

Dans ce chapitre, on a présenté la notion d'un système complexe naturel qui est l'écosystème. Pour comprendre son fonctionnement et son évolution, il était nécessaire de comprendre ces composants et quelques principes écologiques qui le gouvernent, comme : l'interdépendance, la durabilité et la diversité...etc. Ensuite, on a montré comment l'homme a essayé de simuler ces écosystèmes et leurs principes de fonctionnement, pour construire des écosystèmes artificiels pour des raisons différents (comme la validation de quelques hypothèses sur leur dynamique, la résolution des problèmes complexes, ...etc). La notion de problème complexe est examinée dans le chapitre suivant ainsi que quelques stratégies de résolutions.





Chapitre 2 : Les Algorithmes Évolutionnaires Et Co- Évolutionnaires

1. Introduction

Dans ce chapitre on va introduire la notion des algorithmes co-évolutionnaires (ACEs) - qui sont l'extension des algorithmes évolutionnaires et leur utilisation comme des mécanismes de résolution de problèmes complexes. Donc, on va d'abord présenter la notion d'un problème complexe, et les différentes stratégies utilisées pour les résoudre. Ensuite, on va expliquer brièvement le principe de fonctionnement, ainsi que les différents types des algorithmes évolutionnaires (AEs), et après on détaille les algorithmes co-évolutionnaires (ACEs) avec une comparaison entre les ACEs et AEs. Enfin, on va présenter en détail les deux algorithmes qui sont la base de notre travail : le modèle PSO (Particle Swarm Optimisation) et sa nouvelle extension PPO (Predator Prey Optimization).

2. La résolution des problèmes complexes

2.1 Définition d'un problème complexe

Un système complexe est un système qui peut être divisé en plusieurs parties plus simples pour simplifier sa résolution, avec une interaction mutuelle entre ces composants, et avec leur environnement en échangeant des stimuli avec cet environnement. Ces composants doivent pouvoir adapter leurs comportements pour s'adapter aux différents changements de l'environnement.

Mitchell [Mitchell 2002] définit la notion de problème complexe comme un groupe ou une organisation qui se compose de beaucoup de pièces agissantes l'une sur l'autre. Il a présenté aussi plusieurs archétypes des systèmes complexes comme: le climat global, les économies, les colonies de fourmi, et les systèmes immunitaires. Dans de tels systèmes, les individus sont appelés des " composants " ou des " agents " - et les interactions entre eux mènent souvent aux comportements à grande échelle qui ne sont pas facilement prévus d'une



connaissance seulement du comportement des différents agents. De tels effets collectifs s'appellent "les comportements émergents". Les exemples des comportements émergents incluent les changements à court et à long terme de climat, fluctuations des prix sur les marchés, fourragement et construction par des fourmis, et la capacité des systèmes immunitaires de distinguer un "individu" d'un "autre", de protéger l'ancien et de supprimer le dernier. Un autre exemple important d'un système complexe est les écosystèmes (présentés dans le chapitre précédent).

2.2 Les propriétés d'un problème complexe

Pour bien comprendre les propriétés des systèmes complexes, on va présenter quelques définitions qui ont été tirées de la littérature.

- ✓ **Non linéarité:** Un système complexe est vu comme un système non linéaire, dont le résultat des interactions entre composants excède les contributions individuelles de chaque composant [Marcenac 1997]. De plus, si ces composants ont des interactions non-linéaires, il se peut que l'on ait une émergence d'un comportement global.
- ✓ **Nouveauté:** Les systèmes compliqués sont des systèmes que l'on peut réduire en éléments plus simples que l'on peut analyser séparément pour comprendre le système global. Dans le cas des systèmes complexes, la somme des éléments fait émerger de nouvelles propriétés qui ne sont pas dans les éléments eux même [Le Moigne 1989].
- ✓ **Emergence:** Les interactions entre les individus du système complexe mènent souvent aux comportements à grande échelle (comportements émergents) qui ne sont pas facilement prévus seulement de la connaissance du comportement individuel des agents [Mitchell 2002]. Ces interactions locales se font donc à un niveau micro du système (Figure 2.1). Mais les phénomènes que l'on observe au niveau macro (phénomène émergent) ne présentent aucun lien direct avec les phénomènes locaux (notion de surprise du phénomène émergent) [Géorgé 2003].

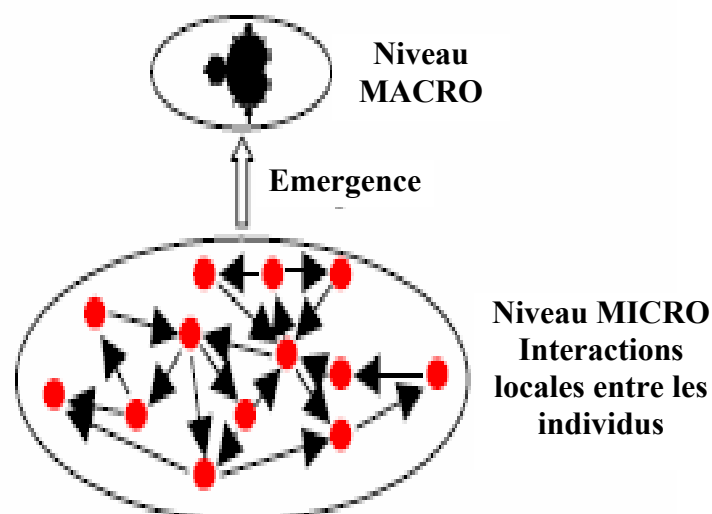


Figure 2.1. Emergence d'une propriété globale à partir d'interactions locales



De nombreux travaux, en particulier dans le domaine de la vie artificielle et des systèmes multi-agents, ont montré expérimentalement que des organisations pouvaient émerger des interactions entre individus [Deneubourg 1989, Ferber 1994]. Néanmoins, l'émergence est souvent présentée comme une sorte de phénomène "magique" qui apparaît "par surprise" et comme conséquence des interactions locales sans que l'on essaye de comprendre pour quelle raison ces structures émergent et quel est leur intérêt [Ferber 1994].

- ✓ **Auto-organisation** est vue comme un ensemble de mécanismes dynamiques dans un système où des structures apparaissent aux niveaux de système qui ne sont pas imposés extérieurement. L'auto-organisation exige que les interactions locales des individus ne fassent pas référence à ces structures globales. Elle est aussi invoquée plus récemment en référence aux colonies d'insectes sociaux où les grands choix de phénomènes collectifs sont auto-organisés [Deneubourg 1989]. Le principe d'auto-organisation désigne la propriété que possèdent certains systèmes d'augmenter leur capacité d'adaptation au cours de leur fonctionnement alors qu'ils sont soumis au bruit (bruit entendu au sens de désordre). C'est ce que l'on appelle également le modèle de « l'ordre à partir du bruit » [Hutzler 2000].

2.3 L'apport de la biologie en informatique

Si l'on ne peut définir un début historique, il y a en effet le livre de Norbert Wiener, *Cybernetics or Control and Communication in the animal and the machine* qui parut en 1947 et qui se montra une référence dans les débuts de l'intelligence artificielle (appelé à l'époque la cybernétique). Voici un tableau récapitulatif de dates historiques du rapprochement entre les deux sciences [Web 12]:

1943	Mc Culloch et Pitts donnent une première interprétation du neurone formel sous un modèle logique.
1947	Parution du livre de référence mondial de Norbert Wiener : <i>Cybernetics or Control and Communication in the animal and the machine</i>
1949	Publication par Hebb d'une théorie formel sur l'apprentissage biologique par modification des synapses (connexions neuronales).
1957	Création du premier système copiant le principe du neurone appelé le perceptron, inventé par Rosenblatt .
1962	John Holland propose la formalisation actuelle des algorithmes génétiques.
1970	Création du Jeu de la Vie par Conway , premier véritable écosystème artificiel.
Depuis 1985	Les réseaux neuronaux deviennent de plus en plus couramment utiliser en informatique.
1999	Une équipe de chercheurs allemands réussis à connecter un neurone sur un circuit et à le stimuler afin d'obtenir des réponses.
2000	Des chercheurs américains fabriquent pour la première fois un processeur biologique, il est composé de quatre neurones de sangsues et arrive à effectuer des additions.

Tableau 2. Récapitulatif de dates historiques du rapprochement entre les deux sciences.



2.4 Les stratégies de résolution des problèmes complexes

Dans le cas de la résolution des problèmes, l'idée est de construire un système collectif artificiel qui va pouvoir résoudre collectivement le problème posé [Web 10]. Il existe une catégorie de problèmes pour lesquels il est difficile, voire impossible, de trouver une solution en un temps limité. Il est alors utile de trouver une technique permettant la localisation rapide de solutions sous-optimales, sachant que l'espace de recherche a une taille et une complexité suffisamment importantes pour éliminer toute garantie d'optimalité. Pour cela, un système capable de s'auto-modifier au cours du temps, tout en améliorant sa performance dans l'accomplissement des tâches auxquelles il est confronté, semble ouvrir la voie à une recherche intéressante [Spalanzani 1999].

L'observation des phénomènes biologiques est une très riche source d'inspiration pour les informaticiens [Lutton 2004] afin de résoudre des problèmes complexes. L'idée principale est d'essayer de simuler la réalité pour construire des modèles informatiques aidant à résoudre des problèmes complexes. Le processus de simulation de la réalité est illustré dans la figure ci-dessous [Drogoul 1996]. A noter que la théorie retenue peut être partie prenante de l'étape de validation du modèle.

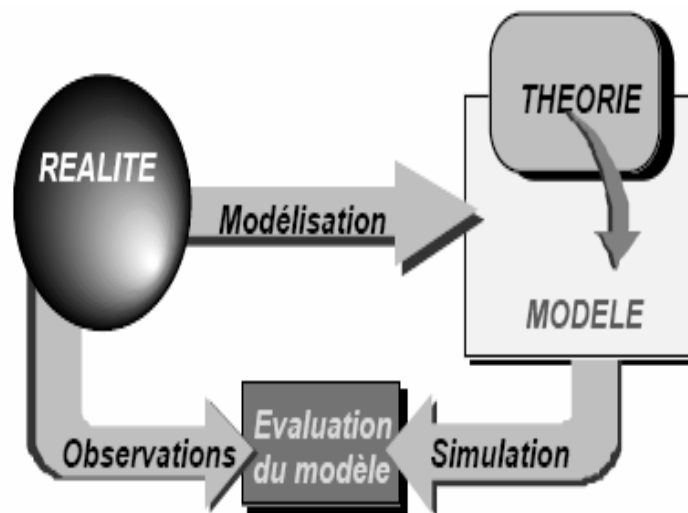


Figure 2.2. Intervention de la théorie dans la construction d'une simulation.

Ces modèles sont maintenant très utilisés en optimisation numérique, lorsque les fonctions à optimiser sont complexes, de forte dimensionnalité, irrégulières, mal connues, ou en optimisation combinatoire, pour des problèmes de théorie des graphes (voyageur de commerce, coloration de graphes), de séquençement de tâches, répartition de ressources, d'emploi du temps, du sac à dos, dont la résolution efficace est fondée le plus souvent sur des AE hybridés avec des techniques "classiques". C'est dans ce cadre, de nombreuses études sur la représentation des problèmes et le traitement des contraintes ont été menées, de façon à produire des algorithmes adaptés aux problèmes réels et de grande taille [Lutton 2004].

Løvbjerg [Løvbjerg 2002] classifie les méthodes de recherche de solution en plusieurs classes (comme il est présenté dans la figure 2.3).

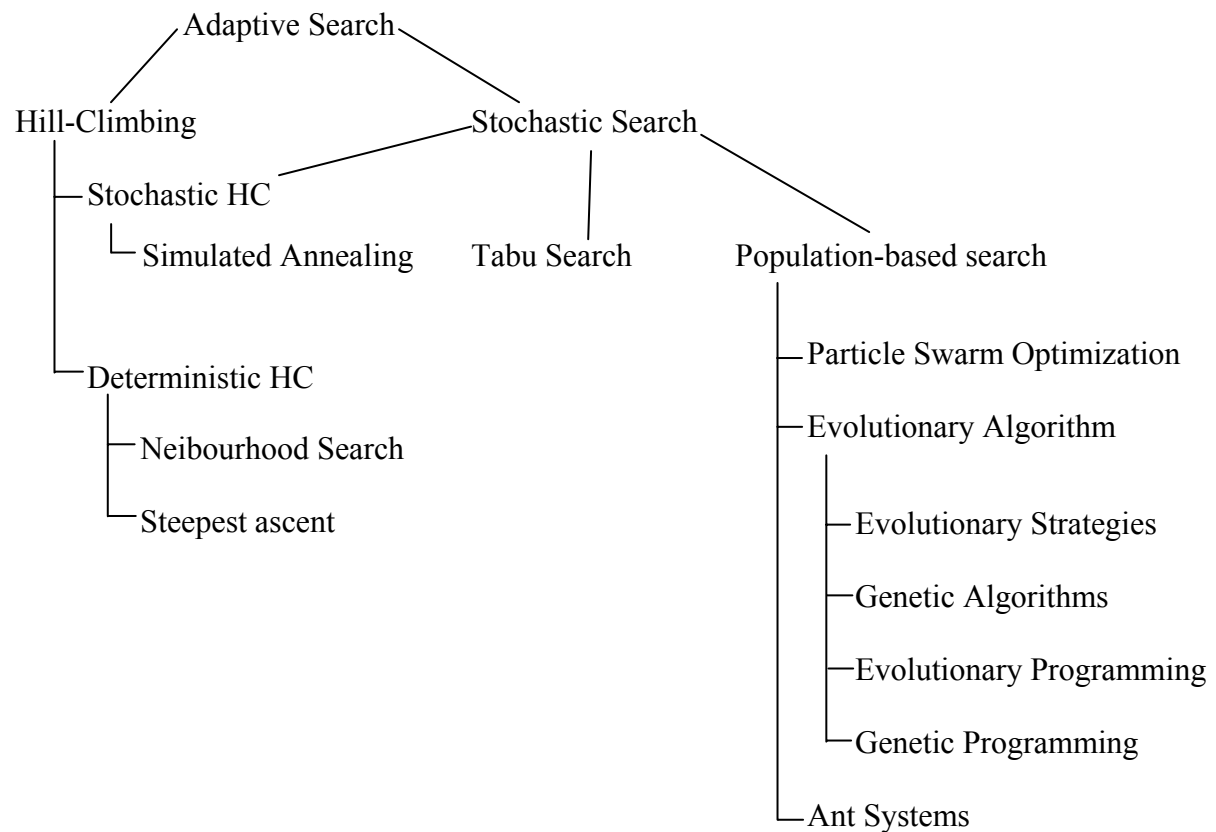


Figure 2.3. L’arbre de différentes techniques adaptatives de recherche.

Dans cette section, on va présenter brièvement les méthodes appropriées les plus répandues de recherche de solution.

2.4.1 Les stratégies Hill-climbing

Les stratégies Hill-climbing sont des méthodes de recherche locale qui modifient une solution et acceptent seulement les changements si la solution modifiée est meilleure que la solution courante. L'efficacité des méthodes Hill-climbing dépend des propriétés de la fonction objective: quand une fonction objective contient beaucoup d'optimums locaux en ce qui concerne le voisinage étudié par la stratégie de recherche, Hill-climbing est facilement emprisonné dans un des optimums locaux. Donc, Hill-climbing est souvent employé en même temps qu'une stratégie de relancement: si emprisonné dans un optimum local, remettez en marche la recherche d'une position aléatoirement choisie dans l'espace de solution [De Meyer 2004].

2.4.2 Le Recuit Simulé

Le recuit simulé (SA : Simulated Annealing) est basé sur une métaphore de la physique: le refroidissement lent (*recuit*) des molécules dans un cristal régulier [Kirkpatrick 1983]. Comme algorithme d'optimisation, il suit une stratégie semblable à Hill-climbing, avec la distinction principale qu'il accepte parfois les solutions qui sont *plus mauvaises* que les courantes. Si une solution nouvellement évaluée est meilleure que la solution courante, on



l'accepte toujours. Si la nouvelle solution est plus mauvaise que la courante, on l'accepte avec une telle probabilité. À la différence des stratégies Hill-climbing, SA peut échapper à des optimums locaux sans besoin de relancement [De Meyer 2004].

Le recuit simulé (SA) a trouvé de nombreuses applications dans toutes les branches de la science et de technologie. Essentiellement elle ajoute au procédé simple de Monte Carlo le prélèvement d'importance selon la distribution de Boltzmann (connue de la thermodynamique) au choix de nouveaux vecteurs de paramètre, évoluant le vecteur des paramètres adaptatifs [Duch 1998].

2.4.3 La recherche Tabou

La recherche Tabou est basée sur une idée très simple. Elle ajoute le concept de *la mémoire* aux méthodes locales de recherche, comme le Recuit simulé. Cette mémoire contient une liste d'endroits déjà visités mais faibles dans l'espace de solution –*la liste de tabou*– qui empêche le processus de recherche de retourner inutilement à ces endroits [Taillard 2001].

La recherche est commencée à un point aléatoire et le meilleur mouvement élémentaire est choisi; des cycles sont évités en gardant la trajectoire de la recherche et en décourageant le système de visiter les mêmes régions encore [Duch 1998].

2.4.4 Les algorithmes évolutionnaires

Le paradigme *du calcul évolutionnaire* (CE) [Kennedy 2001] est inspiré du principe de l'adaptation par la sélection naturelle. La plupart des méthodes de CE sont basées sur la population; en d'autres termes, un groupe d'individus explorent l'espace de solution en parallèle. Les mécanismes principaux de la sélection naturelle sont mis en application: variation, sélection et reproduction; une mesure de fitness détermine la probabilité de la reproduction. La métaphore des paysages de fitness est utile pour envisager le processus de recherche: la population s'adapte en se déplaçant graduellement vers des régions avec des valeurs élevées de fitness. Quatre secteurs développés quasi-indépendamment: les *algorithmes génétiques*, la *programmation évolutionnaire*, les *stratégies d'évolution* et la *programmation génétique* [De Meyer 2004].

Récemment, ces quatre secteurs ont commencé à converger; beaucoup d'algorithmes hybrides ont été développés, et des aspects qui étaient à l'origine représentatifs pour une approche ont été incorporés à d'autres approches [De Meyer 2004].

2.4.5 Les systèmes immunitaires artificiels

Aspects modèles artificiels des systèmes immunitaires (SIA) des systèmes immunitaires vertébrés pour résoudre des tâches d'identification et d'optimisation de modèle [De Castro 2003]. Le principe de cette approche est d'employer le concept de maturation d'affinité de la théorie clonale de choix. Quelques différences avec des approches du CE sont l'utilisation de l' 'affinité' de mesurer la qualité d'une solution, ainsi que la métaphore de l'espace de forme; la confiance unique dans la mutation pour produire la variation; un taux de reproduction qui est proportionnel à l'affinité; et un taux de mutation inversement proportionnel à l'affinité [De Meyer 2004].



2.4.6 Optimisation par colonies de fourmis

Le principe de l'optimisation par colonies de fourmis est apparu au début des années 1990. Il est dû aux chercheurs M. Dorigo, V. Maniezzo et A. Colomi qui expliquent leur théorie dans un article fondateur [Maniezzo 1996], dans lequel ils proposent une nouvelle approche pour l'optimisation stochastique combinatoire et mettent en avant la rapidité de leur nouvelle méthode à trouver des solutions acceptables tout en évitant des convergences prématurées. Ils qualifient leur méthode de versatile (elle peut s'appliquer à des versions similaires d'un même problème), robuste et bien sûr basée sur une population d'individus.

L'optimisation par colonies de fourmis s'inspire du comportement des fourmis lorsque celles-ci sont à la recherche de nourriture. Les fourmis en se déplaçant déposent des phéromones, substances olfactives et volatiles. Chaque fourmi se dirige en tenant compte des phéromones qui sont déposées par les autres membres de la colonie. Les fourmis choisissent leur chemin de manière probabiliste. Comme les phéromones s'évaporent progressivement, le choix probabiliste que prend une fourmi pour choisir son chemin évolue continuellement. Ce mécanisme de choix des chemins peut être illustré par l'exemple suivant - figure 2.4- [Bichot 2005].

À T=1, 32 nouvelles fourmis se présentent en D et 32 autres en B. Mais pendant ce temps, les 16 fourmis parties de D en passant par C sont arrivées en B, de même les 16 autres parties de B en passant par C sont en D. Il y a donc $16 + 16/2 = 24$ phéromones qui indiquent le chemin en passant par C. Ces $16/2 = 8$ phéromones sont dues à l'évaporation. De l'autre côté les 16 fourmis parties de D et les 16 autres parties de B se retrouvent en H. Il n'y a donc que $16/2 = 8$ phéromones qui indiquent en D comme en C le chemin en passant par H. Les nouvelles fourmis vont donc majoritairement choisir le chemin par C. C'est ce que l'on voit à T=2. Dans l'exemple précédent, les phéromones s'évaporent de moitié à chaque nouveau pas d'horloge. Mais le schéma de décroissance peut être différent, c'est l'un des paramètres de modulation des colonies de fourmis avec le nombre de fourmis [Bichot 2005].

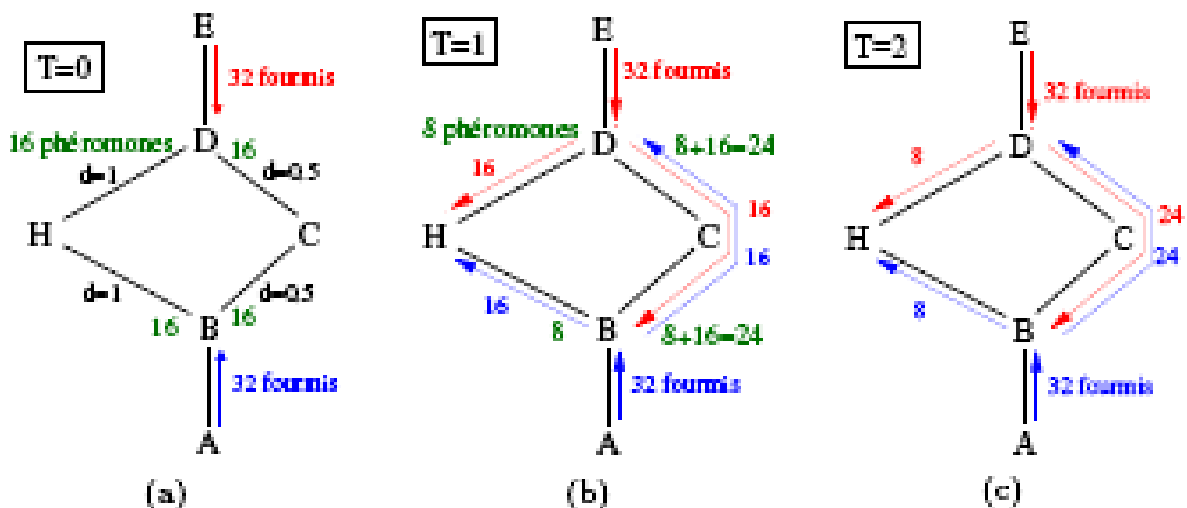


Figure 2.4. Exemple de la stratégie des colonies de fourmis, le contournement d'obstacles

(a) Le schéma initial avec les distances, 32 fourmis arrivent en D et en B. Pas de phéromones, les fourmis choisissent une direction de manière aléatoire.



- (b) A $t=1$, évaporation des phéromones, arrivée en B des fourmis passées par C.
- (c) Les fourmis sont allées majoritairement dans la plus courte branche.

2.4.7 Optimisation par essaim de particules

L'optimisation par essaim de particules (PSO : Particle Swarm Optimization) est une abstraction de la résolution des problèmes individuelle et coopérative dans la société humaine [Kennedy 2001]. Le processus de recherche est envisagé comme groupe de particules volant dans l'espace de recherche. La position et la vitesse des particules sont mises à jour basées sur une expérience individuelle antérieure et l'expérience d'autres particules voisines [De Meyer 2004]. Le principe de fonctionnement de l'algorithme PSO est expliqué en détail dans la section 6.

3. La vie artificielle

"En synthétisant les mécanismes sous-tendant le processus évolutionnaire dans les ordinateurs et d'autres médias 'non biologiques', nous pouvons découvrir des solutions aux problèmes de technologie qui ont longtemps résisté à nos approches traditionnelles", [Langton 1997].

Lors de l'introduction à la vie artificielle, il peut paraître presque comme un concept contradictoire celui d'atteindre « les objectifs complexes ». L'homme devrait aborder ses problèmes tout en attendant le comportement émergent d'un ensemble simple de règles localisées. ce qui est vraiment le cas quand on fait face à la vie réelle. Dans toute la littérature de la vie artificielle, Von Neumann est souvent cité comme le pionnier ou l'un des premiers explorateurs avec ses premiers travaux sur les automates cellulaires (voir [Emmeche 1994]). Ces organisations de la logique pure' sont essentiellement des automates vivantes auto-reproduites dans un espace de grille tout comme un tableau d'échecs. Un développement important dans l'évolution des automates cellulaires est venu avec le jeu de la 'vie' de Horton Conway dans lequel chaque cellule a eu seulement deux états possibles. Fondamentalement, le but de la 'vie' était d'observer le comportement émergent des organisations capables de l'interaction mutuelle en même temps que la modification et la recombinaison génétique des génomes. Plutôt que l'essai pour dicter la direction de l'évolution vers un but prédéterminé, Conway était plus intéressé par les modèles comportementaux chez la population et ainsi aucune fonction de fitness n'a été attribuée aux cellules [Fogel 2000]. Il était observé que la complexité peut rapidement résulter des simulations des modèles initiales très simples de cellules [John 1999].

Le projet Tierra de Ray [Ray 1991] est composé d'une simulation des programmes de code assembleur 'vivant' dans l'unité centrale de traitement et le système d'exploitation, concurrençant pour le TEMPS-CPU. Ray [Ray 1991] déclare un "ensemble d'instructions machines semblables à celles utilisées dans la simulation de Tierra se sont avérés capables du calcul universel. Ceci suggère que les codes machines d'évolution devraient pouvoir produire n'importe quel niveau de complexité." [John 1999].

Bach [Bach 1996] déclare que, " [dans toute la vie artificielle] dans plusieurs cas les agents sont équipés des règles ou des stratégies internes déterminant leur comportement, et un algorithme évolutionnaire est employé pour évoluer ces stratégies".



4. Les algorithmes évolutionnaires

Le terme d'algorithme évolutionnaire (AE) est relatif à une classe de méthodes d'optimisation stochastiques qui simulent le processus de l'évolution naturelle [Francisco 2002]. Ces algorithmes font partie de ce que l'on appelle les méta-heuristiques au même titre que les méthodes de recuit simulé, de recherche Tabou, de colonies de fourmis, d'essaims particulaires, . . . etc, grâce auxquelles on tente de résoudre des problèmes d'optimisation difficiles. Plus intéressé par la résolution de problèmes pratiques que par une classification générale des problèmes d'optimisation et de méthodes susceptibles de les résoudre [Durand 2004].

4.1 Principe de base

L'évolution biologique s'opère en deux étapes : la sélection et la reproduction [Spalanzani 1999] :

- ▶ La sélection naturelle est le mécanisme central qui opère au niveau des populations, en sélectionnant les individus les mieux adaptés à leur environnement.
- ▶ La reproduction implique une mémoire : l'hérédité, sous forme de gènes. Ce matériel héréditaire subit, au niveau moléculaire, des modifications constantes par mutations et recombinaisons, aboutissant ainsi à une grande diversité.

Les AEs sont une classe d'algorithmes d'optimisation par recherche probabiliste basés sur le modèle de l'évolution naturelle. Ils modélisent une *population d'individus* par des points dans un espace. Un individu est codé dans un *génotype* composé de *gènes* correspondant aux valeurs des paramètres du problème à traiter. Le génotype de l'individu correspond à une solution potentielle au problème posé, le but des AEs est de trouver la solution optimale [Spalanzani 1999].

En général, un AE est caractérisé par trois faits [Francisco 2002] :

- Un ensemble de solutions candidates est maintenu ;
- Celui-ci subit un processus de sélection et
- Il est manipulé par des opérateurs génétiques, le plus souvent le croisement et la mutation.

Par analogie avec l'évolution naturelle, les solutions candidates sont appelées *individus*, et l'ensemble des solutions candidates est appelé la *population*. Chaque individu représente une solution possible. Dans le processus de sélection, qui peut être soit stochastique, soit totalement déterministe, les individus de faible qualité sont supprimés de la population, tandis que les individus de haute qualité sont reproduits. Le but est de diriger la recherche sur certaines portions de l'espace de recherche et d'augmenter la qualité moyenne à l'intérieur de la population. [Francisco 2002]. L'environnement a pour but d'évaluer la qualité des individus en leur attribuant une performance (ou *fitness*). Cette valeur favorisera la sélection des meilleurs individus, en vue, après reproduction (opérée par la mutation et/ou recombinaison), d'améliorer les performances globales de la population [Spalanzani 1999].



Plusieurs types d'évolution ont été développés, donnant naissance à trois grandes tendances : les Algorithmes Génétiques (ou *Genetic Algorithms* (GA)), les Stratégies d'Evolution (ou *Evolution Strategies* (ES)) et la Programmation Evolutionnaire (ou *Evolutionary Programing* (EP)). Une branche annexe, la Programmation Génétique (ou *Genetic Programing* (GP)) peut aussi rentrer dans ce type de systèmes.

De ces quatre méthodes classiques, différentes techniques sont dérivées mélangeant les méthodes d'évolution des unes et des autres. Impossible à classer dans l'une des quatre familles citées ci-dessus, elles sont néanmoins considérées comme des AE (Figure 2.5) [Spalanzani 1999].

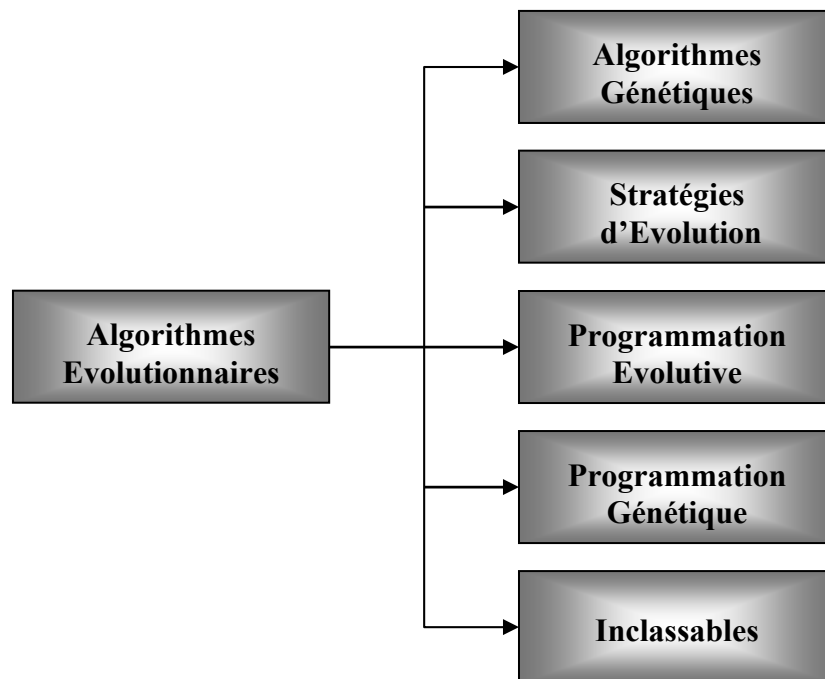


Figure 2.5. Différentes branches des algorithmes évolutionnaires.

Le cycle d'un algorithme évolutionnaire est représenté dans le diagramme suivant (figure 2.6) (il a été réalisé par Marc Schoenauer [Schoenauer 2003]). Dans la Figure 2.6, La variation (stochastique, en rouge) et darwinisme (déterministe ou stochastique, en bleu) sont les notions de base de l'algorithme évolutionnaire. Les étapes de ce cycle sont décrites brièvement dans cette section.

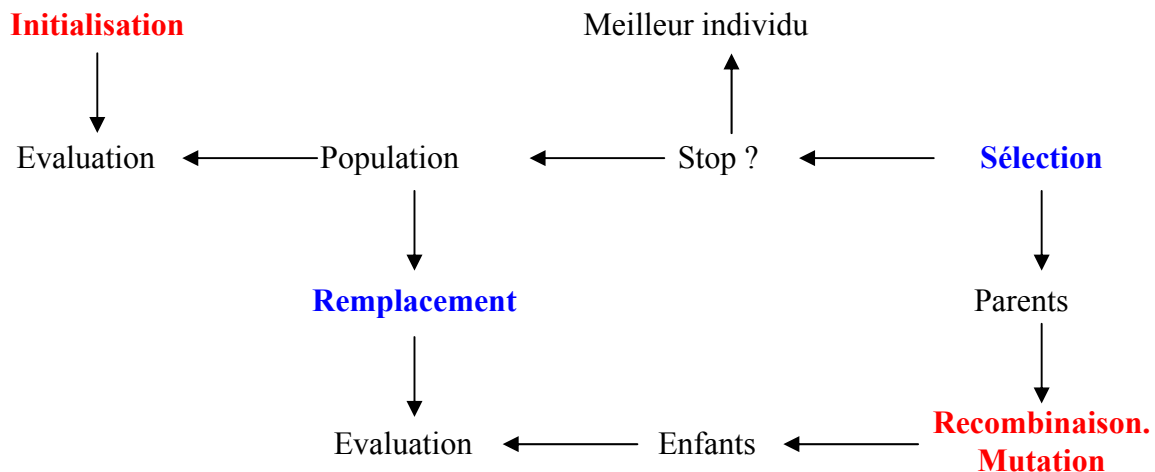


Figure 2.6. Cycle d'un algorithme évolutionnaire.

2.2.1 L'initialisation

Elle est usuellement aléatoire (diverses stratégies sont d'ailleurs envisageables pour échantillonner correctement un espace de recherche complexe ou de grande dimensionnalité). C'est là que l'on peut injecter la ou les solutions initiales au problème que l'on a pu obtenir par ailleurs (par exemple à l'aide d'autres techniques de résolution) [Lutton 2004].

2.2.2 Evolution

L'évolution de la population se fait à l'aide des opérateurs de sélection par roulette [Goldberg 2004], de mutation binaire et de recombinaison. Les AGs favorisent la recombinaison plutôt que la mutation, en donnant de forts taux de recombinaison et de faibles taux de mutation [Spalanzani 1999]. Goldberg [Goldberg 2004] propose dans son livre 80% de croisement et 3% de mutation.

2.2.3 Sélection

La sélection proposée par Goldberg [Goldberg 2004] consiste à sélectionner les individus proportionnellement à leur performance. Un individu, ayant une forte valeur d'adaptation, a alors plus de chances d'être sélectionné qu'un individu mal adapté à l'environnement. Donc, la sélection est basée sur la fitness des chromosomes: plus la fitness est haute, plus est haute la probabilité qu'un chromosome est choisi pour la prochaine génération. Dans *la sélection de la roulette*, la probabilité du choix est proportionnelle à la valeur de fitness. *Les stratégies d'élite* s'assurent que les solutions avec la fitness la plus élevée sont toujours choisies. Dans *la sélection de tournois*, les individus doivent concurrencer directement un ou quelques autres individus pour déterminer la sélection. La variation est présentée par les mécanismes *de la mutation et du croisement* [De Meyer 2004]

La Figure 2.7 représente sur un disque l'ensemble des individus et leur répartition selon leur performance. Si l'on considère la somme des performances de la population (le disque entier), la performance de l'individu 1 représente 9% de la somme, l'individu 2, 40%, etc. Ce



qui équivaut à dire que l'individu 1 a 9% de chance d'être sélectionné, l'individu 2, 40%, etc. [Spalanzani 1999].

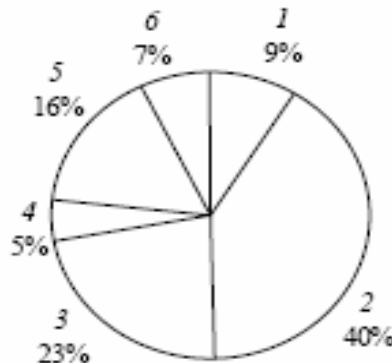


Figure 2.7. Modèle de la roulette. Représentation d'une population de 6 individus.

2.2.4 L'évaluation

Cette étape consiste à calculer (ou estimer) la qualité des individus nouvellement créés. C'est là, et uniquement là qu'intervient la fonction à optimiser. Aucune hypothèse n'est faite sur la fonction elle-même, excepté le fait qu'elle doit servir de base au processus de sélection (elle doit pouvoir permettre de définir une probabilité ou au minimum un ordonnancement des solutions) [Lutton 2004].

2.2.5 Recombinaison

Les parents sélectionnés sont utilisés pour générer des descendants. Les deux opérations principales sont le croisement, qui combine les gènes de 2 parents, et la mutation qui consiste en une légère perturbation du génome [Lutton 2004].

La recombinaison consiste à sélectionner aléatoirement une position de césure (*locus*) et de permuter les parties droites des deux parents. La Figure 2.8 nous donne un exemple de recombinaison [Spalanzani 1999].

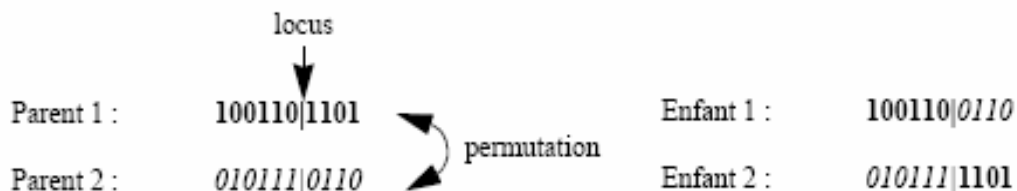


Figure 2.8. Opération de recombinaison.

D'autres types de recombinaison ont été proposés comme la recombinaison:

- ▶ en plusieurs points, qui consiste à sélectionner plusieurs locus et à permuter les différentes parties de la même manière que s'il y avait un seul point de césure,



- ▶ discrète, qui consiste à permuter en moyenne une fois sur deux les éléments du génotype,
- ▶ intermédiaire, qui propose de remplacer le gène de l'individu enfant par une combinaison linéaire des gènes des deux parents,
- ▶ géométrique, qui prend la racine carrée de la multiplication des deux gènes parents.

2.2.6 Mutation

La mutation consiste à prendre des gènes aléatoires dans le génotype et à les altérer selon leur type. Un gène binaire i est muté en non (i). La Figure 2.9 nous donne un exemple de mutation [Spalanzani 1999].



Figure 2.9. Opération de mutation.

2.2.7 L'arrêt

Stopper le processus au bon moment est essentiel du point de vue pratique. Si l'on a peu ou pas d'informations sur la valeur cible de l'optimum recherché (ce qui autorise un arrêt dès que cette valeur est atteinte par le meilleur individu de la population courante), il est délicat de savoir quand arrêter l'évolution. En l'absence de toute information, une stratégie couramment employée consiste à stopper l'algorithme dès qu'un nombre maximal d'itérations est atteint, ou qu'un stade de "stagnation" est identifié. Il est aussi possible de tester la dispersion de la population [Lutton 2004].

4.2 Les algorithmes génétiques

Les algorithmes génétiques AG (ou GA : Genetic Algorithm) ont été développés dans les années 70 par John Holland, ses collègues, et ses étudiants, à l'université du Michigan [Holland 1975], puis approfondis par Goldberg [Goldberg 2004], les AGs sont certainement la branche des AEs la plus connue et la plus utilisée. En fait, la publication du livre de John Holland en 1975 (*Adaptation in Natural and Artificial Systems*) marque la véritable date de naissance des AG, ou du moins la date où ils sont devenus "publics".

La particularité de ces algorithmes est le fait qu'ils font évoluer des populations d'individus codés par une chaîne binaire. Ils utilisent les opérateurs de mutation binaire et de recombinaison de différents types [Spalanzani 1999]. Leurs recherches avaient deux objectifs principaux : (1) mettre en évidence et expliquer rigoureusement les processus d'adaptation des systèmes naturels, et (2) concevoir des systèmes artificiels qui possèdent les propriétés importantes des systèmes naturels [Goldberg 1994]. Pour plus d'informations voir le livre de Goldberge [Goldberg 1994].



Un algorithme génétique recherche le ou les extremums d'une fonction définie sur un espace de données. Pour l'utiliser, on doit disposer des cinq éléments suivants [Durand 2004]:

1. Un principe de codage des éléments de la population. Cette étape associe à chacun des points de l'espace de recherche une structure de données.
2. Un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures.
3. Une fonction à optimiser. Celle-ci prend ses valeurs dans R^+ et est appelée *fitness* ou fonction d'évaluation de l'individu. Elle est utilisée pour sélectionner et reproduire les meilleurs individus de la population.
4. Des opérateurs permettent de diversifier la population au cours des générations et d'explorer l'espace d'état. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace de recherche.
5. Des paramètres de dimensionnement : la taille de la population, le nombre total de générations ou critère d'arrêt et la probabilité d'application des opérateurs de croisement et de mutation.

4.3 La programmation évolutionnaire

La programmation évolutionnaire (PE) a essentiellement commencé par les travaux de Fogel et al. [Fogel 1966]. "Pour simuler l'évolution, il est nécessaire de choisir une représentation mathématique pour l'organisation. Supposant que l'organisation prenne la forme d'une machine à état fini, la succession des machines d'état fini considérées dépend en partie sur le processus statistique de la mutation et en d'autre partie de l'environnement "[Fogel 1966]. L'environnement a été décrit comme une séquence des symboles pris d'un alphabet fini, avec le problème évolutionnaire d'évoluer un algorithme qui opérerait les symboles pour maximiser un résultat. Les machines d'état fini ont fourni la représentation nécessaire du comportement exigé [John 1999].

Bien que les stratégies d'évolution et la programmation évolutionnaire soient étonnamment semblables, le développement a procédé indépendamment les uns des autres jusqu'en 1992. La PE a opéré en exposant une population des machines finies à l'environnement. Une fonction de profit a été employée comme fonction de fitness, basée sur l'erreur de rendement. Chaque parent a fait créer une progéniture, par mutation aléatoire, avant que la population entière soit devenue sujet à *l'évolution* non régressive, c-à-d une machine doit se ranger dans la moitié supérieure de la population afin de survivre. La programmation évolutionnaire n'a pas des processus génétiques tels que le croisement et l'inversion (pour d'autres détails référez-vous à [Bach 1996]).

Les années 80 ont vu l'utilisation des machines de Moore utilisées dans la PE [Fogel 2000] et plus récemment divers problèmes combinatoires d'optimisation.

La programmation évolutionnaire (PE) ne simule pas les mécanismes de la transmission génétique, mais plutôt l'adaptabilité du comportement. À l'origine, les individus étaient des machines d'état fini, principalement utilisées pour résoudre des problèmes de prévision. La fitness des individus est calculée en les exposant à l'environnement et en évaluant leur



exécution de prévision. La probabilité de la mutation dans la PE est en général beaucoup plus haute que dans les AGs [De Meyer 2004].

4.4 Les stratégies d'évolution

La représentation continue, ou représentation réelle, est historiquement liée aux approches de type stratégies d'évolution [Lutton 2004]. Les stratégies d'évolution (SE) sont apparues dans les années 70 avec les travaux d'Ingo Rechenberg et ces travaux ont été poursuivis par Hans-Paul Schwefel [Schwefel 1984].

La première particularité de ces méthodes est de coder les paramètres du problème à résoudre en nombres réels. Ceci implique une méthode de mutation adaptée aux réels. La seconde est d'effectuer une sélection déterministe des individus en ne choisissant que les n ($n \geq 1$) individus classés selon leur performance. Enfin, la troisième, est d'encoder les paramètres d'évolution directement dans le génotype afin de les faire évoluer au même titre que les valeurs des paramètres de la solution du problème [Spalanzani 1999].

La variation est produite par mutation, aussi bien que par *la recombinaison* (semblable au croisement dans les AGs). Les paramètres sont mutés en ajoutant des nombres aléatoires naturellement distribués avec la valeur moyenne nulle aux valeurs courantes, de sorte que les individus de la prochaine génération explorent le secteur dans l'espace de solution autour des solutions codées par les parents. La quantité de mutation (commandée par les désaccords de la distribution naturelle) elle-même est soumise à la variation et au choix, menant à *l'auto-adaptation* de l'algorithme. La recombinaison peut faire participer deux parents (*recombinaison locale*) ou la population entière (*recombinaison globale*), et elle est mise en application dans un des deux manières: elle est *discrète* signifiant qu'une valeur de paramètre est choisie parmi seulement un des parents; ou *intermédiaire* signifiant que la valeur de paramètre de l'enfant est *une combinaison linéaire* des valeurs de paramètre des parents [De Meyer 2004]. Pour une discussion sur les stratégies évolutionnaires, référez-vous à [Bach 1996].

4.5 La programmation génétique

La programmation génétique (PG) correspond à une représentation de structures de longueurs variables sous forme d'arbres. Elle a été inventée à l'origine pour manipuler des programmes codés en LISP, dans le but de créer des programmes qui puissent résoudre des problèmes pour lesquels ils n'ont pas été explicitement programmés. La richesse de la représentation arborescente de taille variable est l'une des clés du succès de ce courant dans le milieu évolutionnaire, en effet beaucoup de problèmes d'optimisation, de commande ou de contrôle peuvent se formuler comme un problème d'induction de programme [Lutton 2004].

La PG emploie la puissance des AGs pour rechercher, dans l'espace des programmes possibles, pour trouver des programmes appropriés pour l'application désirée. Koza la décrit comme " technique la plus générale d'étude de machine ", une réclamation qui est justifiée par le grand nombre d'applications auxquelles elle a été appliquée avec succès [Koza 1999]. La représentation généralement utilisée dans la PG est l'arbre d'analyse d'un programme. Les arbres d'analyse des programmes ont l'avantage qu'ils représentent des programmes machines syntaxiquement corrects, et sont facile à manipuler. Les branches dans l'arbre d'analyse représentent les fonctions qui prennent des arguments, les feuilles représentent des fonctions à



zéro-argument, des variables ou des constantes. La fitness dans la PG est mesurée en testant le programme qu'elle représente. La figure suivante illustre la représentation arborescente de la fonction $((\cos(x) + 2*y)*(1+x))$ [Qureshi 2001].

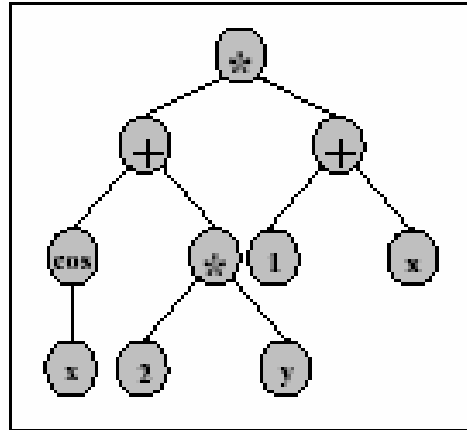


Figure 2.10. Exemple d'une représentation arborescente

La PG évolue les programmes exécutables représentés en tant que structures arborescentes au lieu des représentations binaires de chaînes des AGs. En plus, dans la PG les structures évoluées ne sont généralement pas de longueur fixe. La fitness d'un individu est calculée en exécutant les programmes évolués et en mesurant leur exécution en résolvant le problème [De Meyer 2004].

5. Les algorithmes co-évolutionnaires

Les algorithmes Co-évolutionnaires (ACEs) sont une nouvelle extension des algorithmes évolutionnaires. Ils sont basés sur le concept de Co-évolution, qui se rapporte à l'évolution simultanée de deux espèces ou plus avec une fitness couplée. En général, les algorithmes co-évolutionnaires peuvent être classés en deux catégories : coopératifs et concurrentiels.

5.1 Pourquoi les ACEs ?

les algorithmes évolutionnaires à population simple effectuent souvent la mal stagnation de manifestation, convergence aux optimums locaux, et le précieux informatique – quand confronté avec des problèmes présentant un ou plusieurs des caractéristiques suivantes [Peña-Reyes 2001]: (1) après la recherche, la solution est complexe, (2) le problème ou sa solution est clairement décomposable, (3) le génome code différents types de valeurs, (4) des interdépendances fortes entre les composants de la solution, (5) l'ordre des composants affecte rigoureusement la fitness. La co-évolution coopérative aborde efficacement ces questions, élargissant par conséquent l'étendue des applications de calcul évolutionnaire [Zaritsky 2004].



La dynamique des algorithmes Co-évolutionnaires est plus complexe que celle des algorithmes évolutionnaires. Les travaux récents sur le calcul Co-évolutionnaire adressent des tentatives de fournir un cadre mathématique pour l'application des algorithmes Co-évolutionnaires (ACEs) [Bucci 2002]. Juille et Pollack [Juille 1990] ont discuté l'étude Co-évolutionnaire et les issues relatives.

Le Calcul Co-évolutionnaire est une approche relativement récente à résoudre des problèmes d'optimisation avec la majorité de recherche étant éditée dans la dernière décennie et une grande partie s'est appliquée à la classification par les automates cellulaires ou les réseaux de neurones [Watson 2001].

5.2 Principe de fonctionnement

En nature, les organisations interagissent non simplement avec leur environnement, mais également, l'un avec l'autre. Parfois elles sont en concurrence directe, l'un avec l'autre, comme se produit avec les prédateurs et la proie, ou l'hôte et le parasite. La fitness de ces organisations devient relative. Un prédateur est considéré convenable s'il peut attraper la proie nécessaire pour sa survie. Réciproquement, une proie convenable peut mieux échapper à des prédateurs [Qureshi 2001].

Dans un ACE, la fitness d'un individu est basée sur son interaction avec d'autres individus dans la population : ainsi l'évaluation de la fitness est sensible au contexte et subjective [Panait 2003]. Dans la *co-évolution concurrentielle*, les individus bénéficient à l'échec de leurs pairs ; mais dans la *co-évolution coopérative*, les individus réussissent ou échouent ensemble dans la collaboration. Une approche standard à appliquer des algorithmes co-évolutionnaires coopératifs (ou ACECs) à un problème d'optimisation commence par la décomposition du problème en sous-composants, puis en assignant chaque sous-composant à une population séparée des individus [Potter 1995, Potter 1997, Potter 2000].

La co-évolution se rapporte à l'évolution simultanée de deux espèces ou plus avec une fitness couplée. C'est contrairement au paradigme évolutionnaire usuel où une population simple évolue sous la pression de choix d'une fonction donnée de fitness qui joue le rôle de l'environnement. Dans ces conditions, on peut penser que l'évolution comme processus co-évolutionnaire où les changements d'une certaine espèce influencent d'autres espèces, c.-à-d., l'environnement change dynamiquement [Zaritsky 2004].

5.3 Les algorithmes co-évolutionnaires compétitifs

Les relations Prédateur-Proie sont les exemples les plus connus de la co-évolution concurrentielle. Dans de telles relations, le succès pour un côté signifie l'échec à l'autre côté. Ainsi, l'amélioration du prédateur fait pression sur la proie de se défendre mieux afin de maintenir ses chances de survie, et quand la proie s'améliore, le prédateur doit également gagner de meilleures possibilités afin de maintenir ses chances de survie.

Dans les algorithmes co-évolutionnaires concurrentiels, la fitness d'un individu d'une des espèces est basée sur la concurrence directe avec des individus d'autres espèces, qui évoluent à leur tour séparément dans leurs propres populations. L'augmentation de la fitness d'une des espèces implique une diminution dans la fitness de l'autre espèce.

Hillis [Hillis 1990] était le premier qui a proposé l'utilisation informatique de la co-évolution de prédateur-proie. Le problème a consisté en évoluant un réseau assortissant pour



16 nombres entiers impliquant un nombre minimum d'échanges. Hillis a évolué deux populations, la première est composée des réseaux assortissants, la seconde des problèmes assortissants (cas d'essai). La fitness d'un réseau assortissant est définie en tant qu'à quel point il assortit les problèmes assortissants; un problème assortissant est marqué selon la façon dont il est résistant pour des individus de la première population pour l'assortir. De cette façon, les meilleurs réseaux apprennent à assortir les ensembles de nombres de plus en plus difficiles, de ce fait évitant le besoin d'examiner chacun des 16! problèmes assortissants possibles.

Paredis [Paredis 1995] a présenté une évaluation partielle et continue de fitness, appelée l'évaluation de fitness de vie (LTFE : life-time fitness evaluation). Cette technique est inspirée de la nature, tandis que la "fitness" d'un individu se compose d'une série continue d'essais pendant sa vie. En outre, ces essais ne sont pas strictement prédéfinis, mais ils sont déterminés en se basant sur l'environnement, qui est influencé par l'individu en cours d'évaluation aussi bien que par d'autres individus. Paredis a employé cette technique, combinée avec la co-évolution concurrentielle, pour améliorer la puissance de la recherche artificielle sur une classe des problèmes qui impliquent la recherche d'une solution qui rencontre certain a priori donné des critères (cette classe des problèmes est appelée par Paredis comme "test-solutions problems"). Dans [Paredis 1995], Paredis emploie la co-évolution et le LTFE concurrentiels pour un problème de classification. Il a utilisé deux espèces co-évoluées; une population de solutions candidates et une population des exemples pré-classifiés de formation. Dans [Paredis 1997] Paredis a employé les techniques semblables au co-évoluer un réseau de neurones pour un problème bien connu de commande de bioréacteur.

Rosin et Belew [Rosin 1997 a] utilisent deux jeux (Nim et Tic-Tac-Toe à 3D) comme problèmes d'essai pour explorer trois nouvelles techniques dans la co-évolution concurrentielle: la fitness concurrentielle partagée change la manière dont la fitness de solution est mesurée, le prélèvement partagé fournit une méthode pour choisir un ensemble fort et divers de "tests" pour l'évaluation de solution, et l'arrivée de la nouvelle encourage des courses aux armements (arms races) en gardant les individus convenables des générations antérieures.

Dawkins [Dawkins 1986] décrit comment une telle *Co-évolution concurrentielle* mène à une course aux armements, où chaque organisation évolue pour contrecarrer n'importe quelle amélioration par évolution de son concurrent. Cette approche a été appliquée à la PG [Haynes 1995], et implique de créer et d'évoluer deux populations ou plus au lieu d'une. La fitness des individus dans une population est mesurée relativement aux individus dans l'autre et vice versa [Qureshi 2001].

5.4 Les algorithmes co-évolutionnaires coopératifs

Les algorithmes co-évolutionnaires coopératifs (également appelés symbiotiques) impliquent un certain nombre d'espèces évoluées indépendamment, qui forment ensemble des structures complexes, bien adaptées pour résoudre un problème. L'idée est d'employer plusieurs populations (espèces) indépendamment maintenues, chacune spécialisée à une partie du problème complet, avec la fitness d'un individu mesurée selon sa capacité de collaborer avec des individus d'autres espèces pour construire une solution globale. Un certain nombre d'algorithmes co-évolutionnaires coopératifs ont été présentés ces dernières années [Eriksson 1997, Paredis 1995].



Paredis [Paredis 1995] a appliqué la co-évolution coopérative aux problèmes impliquant de trouver simultanément les valeurs d'une solution et de leur ordre proportionné. Moritz [Moriarty 1997] a employé une approche co-évolutionnaires coopérative pour évoluer les réseaux de neurones. Chaque individu dans une espèce correspond à un neurone caché simple d'un réseau de neurones et de ses raccordements avec les couches d'entrée et de sortie, et les individus de la seconde population codent des ensembles de neurones cachés.

Potter [Potter 1997] et Potter et DeJong [Potter 2000] ont développé un modèle dans lequel un certain nombre de populations explorent différentes décompositions du problème. Dans le système de Potter et DeJong, chaque espèce représente un sous-composant d'une solution (globale) potentielle. Des solutions complètes sont obtenues en rassemblant les membres représentatifs de chacune des espèces. La fitness de chacun individuel dépend de la qualité (une partie de) des solutions complètes qu'elle a participées dedans, de ce fait mesurant à quel point elle coopère à résoudre le problème. L'évolution de chaque espèce est commandée par un algorithme évolutionnaire séparé et indépendant. La figure 2.12 montre l'architecture générale du Potter et du cadre co-évolutionnaires coopératif de DeJong, et la façon dont chaque algorithme évolutionnaire calcule la fitness de ses individus en les combinant avec les représentants choisis de l'autre espèce. Des représentants sont habituellement choisis par l'intermédiaire d'une stratégie avide en tant qu'individus les plus convenables à partir de la dernière génération. Les résultats présentés par Potter et DeJong [Potter 2000] prouvent que leur approche aborde en juste proportion des questions comme la décomposition et les interdépendances entre les sous-composants de problème. L'approche co-évolutionnaires coopérative exécute aussi bien, et améliore souvent, que des algorithmes évolutionnaires à simple population.

La figure 2.12 montre le processus évolutionnaire de la perspective de l'espèce 1. L'individu étant évalué est combiné avec un ou plusieurs représentants de l'autre espèce afin de construire plusieurs solutions qui sont testées sur le problème. La fitness de l'individu dépend de la qualité de ces solutions.

Eriksson et Olsson [Eriksson 1997] ont appliqué le modèle de Potter et DeJong à un problème d'optimisation de vérification de l'inventaire. Leur résultat principal montrait cela recueillant un certain nombre d'espèces fort-connexe, et les évoluant pendant qu'une espèce simple dans le modèle co-évolutionnaire coopératif pourrait prouver favorable. Plus récemment, Peña-Reyes et Sipper [Peña-Reyes 2001] ont employé la co-évolution coopérative pour évoluer les systèmes brouillés pour le diagnostic de cancer de sein et pour des classificateurs Fuzzy d'Iris.

Concernant la PG, dans *la co-évolution coopérative*, les programmes sont examinés ensemble, et des paires de programmes qui fonctionnent bien ensemble sont récompensées avec une fitness plus élevée [Qureshi 2001].

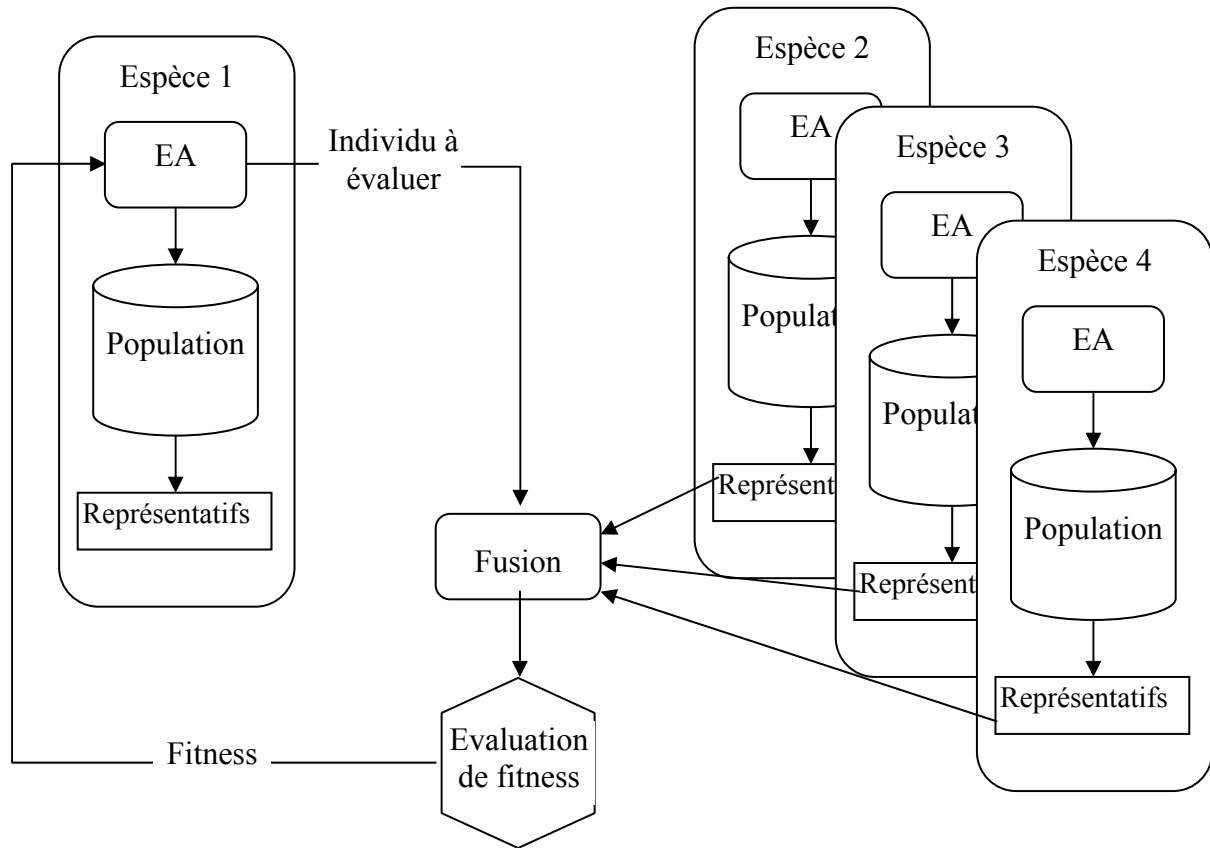


Figure 2.11. Le système co-évolonnaire coopératif du Potter et du DeJong.

5.5 L'application des ACEs pour la résolution des problèmes complexes

L'optimisation Multi-objectives est, clairement, la classe la plus importante des problèmes en science et technologie. Michelle Galea [Michelle 2003], a présenté un algorithme génétique co-évolonnaire inspirés de la théorie d'étude de développement. Il a comparé empiriquement ses résultats à sept autres techniques évolutionnaires de recherche pour l'optimisation multi-objectives. Les résultats obtenus prouvent que l'algorithme génétique co-évolonnaire est très performant comparé aux autres algorithmes évolutionnaires et recherche aléatoire.

Un autre exemple des ACEs est l'algorithme MOCCGA (Multi-Objective Co-evolutionary Cooperative Genetic Algorithm) [Keerativuttitumrong 2002], qui intègre la co-évolution coopérative [Potter 2000] avec MOGA (Multi-Objective Genetic Algorithm) [Fonseca 1993]. Il utilise un rang de dominance pour des individus où un calcul du nombre d'individus dominant un individu est le critère de fitness. Dans MOCCGA les objectifs sont évalués deux fois pour chaque individu; avec les meilleurs individus rangés de chaque sous-population, aussi bien que le choix aléatoirement des individus, qui est l'approche décrite par Potter et DeJong pour réduire au minimum la convergence prématurée sur quelques problèmes d'essai [Potter 1995].

Coello et Sierra [Coello 2003] ont également démontré un algorithme coopératif co-évolonnaire pour l'optimisation multi-objectifs qui subdivise l'espace variable de décision.



Cette approche détermine quelles parties des intervalles des variables de la décision sont employées et jette les parties des intervalles qui ne sont pas utilisées par le processus de recherche. Elle subdivise également les intervalles, ainsi les sous-populations séparées peuvent opérer les parties de ces intervalles qui ont contribué à la recherche. On élimine les sous-populations qui n'apportent pas des contributions de la recherche.

Un autre algorithme génétique coopératif qui a été développé par Mao [Mao 2001], utilise les paramètres de symbiose qui peuvent affecter la fitness des individus avant qu'ils soient rangés. Ces paramètres permettent la modification de la fitness d'un individu basé sur l'interaction entre les fonctions objectives et entre les individus.

Dans [Antony 2004], un nouveau mécanisme de formation de collaboration est démontré dans un algorithme co-évolutionnaire coopératif, qui utilise les aspects uniques de l'optimisation multi-objectives. Par la co-évolution coopérative des composants qui participent aux solutions candidates non-dominées, il est possible de trouver des collaborateurs qui coopèrent à des bonnes solutions. Il est aussi démontré que le NSCCGA (Cooperative Coevolutionary Multi-Objective Algorithm Using Non-dominated Sorting) concurrence bien avec le NSGA-II sur une variété de problèmes d'essai montrant des caractéristiques diverses.

L'algorithme coopératif co-évolutionnaire est utilisée aussi pour résoudre le problème complexe de SCS (Shortest Common Superstring) [Zaritsky 2004], une comparaison est faite entre 4 approches pour trouver des solutions au problème de SCS : SGA (Standard Genetic Algorithm), CCA (Cooperative-Coevolutionary Algorithm), GA (Greedy Algorithm) et PCGA (Parallel Coevolutionary-Greedy Approach). Il est prouvé que l'approche co-évolutionnaire produit le meilleur résultat.

Un autre exemple emploie une approche Co-évolutionnaire à modéliser le système *Fuzzy* pour la classification. Les auteurs comparent les résultats obtenus par *Fuzzy CoCo* à ceux obtenus à partir de leurs premiers travaux qui sont évolutionnaires mais non-co-évolutionnaires. Ils rapportent une augmentation de l'exactitude de classification, et, quand vu le nombre d'évaluations de fitness effectuées, une diminution de la dépense de calcul. *Fuzzy CoCo* marque également mieux sur l'exactitude de classification en comparaison avec un système qui extrait des règles de classification des réseaux de neurones [Setiono 2000]. Les auteurs croient qu'ils ont obtenu les meilleurs résultats jusqu'ici sur cet ensemble de données pour les systèmes de règle Genetic-Fuzzy et Neuro-Booléenne [Michelle 2003].

Un PSO Co-évolutionnaire amélioré a été présenté par Krohling et Hoffmann [Krohling 2004] pour résoudre les problèmes minimum-maximum et il a été examiné sur quelques problèmes d'optimisation contraints. Les résultats de simulation indiquent que l'algorithme amélioré de PSO employant la distribution gaussienne surpasse le PSO standard et montre des résultats concurrentiels avec ceux édités dans la littérature.

En ce qui concerne le traitement d'images, un algorithme co-évolutionnaire cellulaire (CCA) est présenté dans [Veenman 2003], pour optimiser d'une manière distribuée le problème de segmentation d'images. Dans les expériences, ils ont montré l'efficacité de la méthode et ils ont la comparé à deux autres méthodes bien connues de segmentation : le modèle MRF (Markov Random Field) basé sur la méthode de segmentation et le modèle SMG (Split-and-Merge-and-Group).



6. L'algorithme PSO et l'extension PPO

L'idée de Silva et al [Silva 2003] consiste à comparer expérimentalement l'exécution de plusieurs variantes de PSO et une nouvelle approche appelée PPO, basée aussi sur la notion d'essaim. PPO combine les idées de PSO avec une stratégie inspirée du système prédateur-proie. L'objectif principal de cette nouvelle approche est d'introduire une nouvelle particule, appelée Prédateur, dans l'essaim, dont le but est de maintenir la diversité dans l'essaim (écosystème), et pour éviter la convergence prématurée aux optimums locaux. Les principes des deux algorithmes sont décrits en détail dans cette section.

6.1 L'algorithme PSO

L'optimisation par essaim particulaires (PSO : Particle Swarm Optimization) est une stratégie relativement récente pour l'optimisation des fonctions. Conceptuellement des similitudes doivent être trouvées dans les domaines des algorithmes génétiques aussi bien que la vie artificielle [Schoeman 2005]. L'optimisation d'essaim de particules (PSO) a été à l'origine conçue par James Kennedy et Russel Eberhart [Kennedy 1995] et a été comparée aux algorithmes génétiques [Eberhart 1998] pour trouver efficacement les solutions optimales dans les grands espaces de recherche.

PSO est un nouvel algorithme évolutionnaire, où la population s'appelle maintenant un essaim et chaque individu s'appelle une particule. Il a été inspiré de la dynamique des interactions sociales entre les individus, étant au commencement influencé par le travail dans la simulation du vol coordonné en bandes d'oiseaux. [Silva 2003]. PSO a été avec succès appliqué aux domaines de l'optimisation de fonction, du jeu apprenant, groupement de données (data Clustering), de l'analyse d'images et de réseau de neurones [Franken 2004].

PSO est aussi liée à l'intelligence en essaim. Selon Millonas [Millonas 1994], qui a développé ses modèles pour des applications dans la vie artificielle, cinq principes de base d'intelligence d'essaim : (1) principe de proximité: la population devrait pouvoir effectuer des calculs simples de l'espace et de temps, (2) le principe de qualité: la population devrait pouvoir répondre aux facteurs de qualité dans l'environnement, (3) le principe de la réponse diverse: la population ne devrait pas réaliser ses activités le long des canaux excessivement étroits, (4) le principe de la stabilité: la population ne devrait pas changer son mode de comportement chaque fois que l'environnement change, (5) est le principe de l'adaptabilité: la population doit pouvoir changer le mode de comportement pour répondre aux changements sérieux ou durable de l'environnement. Notez que les principes (4) et (5) sont les côtés opposés de la même pièce de monnaie [Kennedy 1995].

La nature stochastique de l'algorithme, facilitera cependant, également l'exploration des régions non découvertes dans l'espace de recherche, ayant pour résultat une technique rapide et efficace pour trouver un optimum global simple. [Schoeman 2005].

6.1.1 Principe de fonctionnement

Comme on a dit précédemment, la population dans PSO s'appelle un essaim, dont chaque individu s'appelle particule. Le mouvement des particules est influencé non seulement par l'expérience précédente de chaque particule (la meilleure position qu'elle a déjà trouvée), mais également par une contrainte sociale de se déplacer vers la meilleure position trouvée par



ses voisins. Pour mettre en application ces comportements, chaque particule est définie par sa position et sa vitesse dans l'espace de recherche. Dans chaque itération, des changements - résultants des deux influences de la trajectoire des particules - sont faits à sa vitesse. La position des particules est alors mise à jour en conséquence à la vitesse calculée [Silva 2003]. Les variantes principales de PSO et le modèle culturel derrière elle sont en détail discutés dans [Kennedy 2001].

Le mouvement des particules est fait de telle manière qu'il leur permet de trouver incrémentalement de meilleures solutions. Ce qui conduit principalement un algorithme de PSO, est l'interaction sociale entre ses particules. Les particules dans un essaim partagent leur connaissance l'une avec l'autre, spécifiquement quant à la qualité des solutions qu'elles ont trouvées aux points spécifiques dans l'espace de recherche. La meilleure solution découverte par une particule est désignée sous le nom de "*la meilleure solution personnelle*". Les particules se déplacent alors vers d'autres meilleures solutions personnelles en utilisant certaines vitesses, afin d'essayer de découvrir des solutions améliorées [Papacostantis 2005].

La position de chaque particule, dans l'espace multidimensionnel de recherche, sera représentée dans l'équation par un vecteur X (où chaque dimension représente un paramètre dans la fonction objective). La vitesse de chaque particule est représentée par un vecteur V, et elle permet de mettre à jours la position de la particule dans chaque itération. La particule se déplace dans l'espace de recherche en ajoutant la valeur de la vitesse à sa position. V détermine la trajectoire des particules et dépend de deux paramètres pour chaque particule i : voler vers sa meilleure position précédente et voler vers la meilleure position précédente de ses voisins.

Les équations générales, pour mettre à jour la position et la vitesse pour une certaine particule i, sont les suivantes:

$$\begin{cases} v(t+1) = v(t) + \varphi_1 * (p(t) - x(t)) + \varphi_2 * (g(t) - x(t)) \\ x(t+1) = x(t) + v(t+1) \end{cases}$$

Où

$$\left\{ \begin{array}{l} v(t) : \text{est la vitesse courante} \\ x(t) : \text{est la position courante} \\ p(t) : \text{est sa meilleure position précédente} \\ g(t) : \text{est la meilleure position trouvée par ses voisins} \\ \varphi_1 = c_1 * r_1 \text{ et } \varphi_2 = c_2 * r_2 : c_1 \text{ et } c_2 \text{ sont les constantes: cognitive et sociale} \\ \text{d'accélération respectivement, et } r_1 \text{ et } r_2 \text{ sont des nombres aléatoires distribués entre 0} \\ \text{et 1} \end{array} \right.$$

La nouvelle position de la particule, au temps t+1, est calculée en additionnant sa position précédente et sa nouvelle vitesse v(t+1). La nouvelle vitesse est obtenue en additionnant la vitesse précédente, le terme $\varphi_1 * (p(t)-x(t))$ qui représente la partie cognitive de l'apprentissage et le terme $\varphi_2 * (g(t)-x(t))$ qui représente la partie sociale de l'apprentissage.



Shi et Eberhart concluent [Shi 1999] que l'exécution du PSO n'est pas sensible à la taille de population. Ceci signifie que PSO fonctionnera bien avec un petit nombre de particules comparées au nombre d'individus requis pour un algorithme génétique. Comme chaque particule a une évaluation de fonction de fitness par itération, le nombre d'évaluations de fonction de fitness peut être réduit ou plus d'itérations peuvent être effectuées. Shi et Eberhart observent également que les balances de PSO - avec l'insensibilité de taille - ont été vérifiées par leur étude [Løvbjerg 2002].

6.1.2 Comparaison entre PSO et les algorithmes génétiques

L'algorithme PSO partage beaucoup de points communs avec l'algorithme génétique (AG). Les deux algorithmes commencent par une population d'individus générée aléatoirement, tout les deux ont des valeurs de fonction objective pour évaluer la population. Les deux algorithmes mettent à jours avec la population et cherchent l'optimum avec des techniques aléatoires. Les deux systèmes ne garantissent pas le succès. Elles ont également la mémoire, qui est importante pour l'algorithme. Comme les algorithmes génétiques, PSO est basé sur une population qui converge lentement vers une ou plusieurs solutions. Cependant, avec PSO, les particules sont préservées dans tout le processus entier; ils ne meurent pas. Le contraire à l'algorithme génétique, qui est basé sur la concurrence pour de meilleures chances de survie et de reproduction. PSO emploie un genre de coopération entre les particules, ceci est réalisé par l'échange des coordonnées des meilleures solutions qui ont été produites jusqu'à ce point [Bourquin 2004].

PSO n'a traditionnellement aucun croisement entre les individus, n'a aucune mutation et les particules ne sont jamais substituées par d'autres individus pendant l'exécution. Au lieu de cela PSO raffine sa recherche en attirant les particules aux positions représentant les bonnes solutions. [Løvbjerg 2002].

6.1.3 Les extensions de PSO

Selon Angeline [Angeline 1998], PSO, tout en rapidement convergeant vers un optimum dans les premières itérations, a des problèmes quand il vient pour atteindre une solution optimale proche. Pour résoudre ceci, Shi et Eberhart [Shi 1998] présentent l'utilisation d'un poids décroissant linéaire d'inertie. Dans [Shi 1999], les résultats d'utilisation de PSO avec le poids d'inertie sur plusieurs fonctions de repère sont présentés et prouvent que cette nouvelle extension donne de meilleurs résultats que PSO original [Silva 2003]

La formule originale développée par Kennedy et Eberhart a été améliorée par Shi et Eberhart [Shi 1999] avec l'introduction d'un paramètre d'inertie w qui augmente l'exécution globale de PSO [Carlisle 2000]. L'équation, de mise à jour de la vitesse, est modifiée après l'introduction du poids d'inertie comme suit :

$$v(t+1) = w * v(t) + \varphi_1 * (p(t) - x(t)) + \varphi_2 * (g(t) - x(t))$$

Dans la formule, on voit que le poids d'inertie w est multiplier à la vitesse de chaque particule au pas de temps t ; avec $w \in [0,1]$ pour imposer la convergence. On peut utiliser une vitesse maximale, v_{\max} qui permet de fixer une limite supérieure pour les vitesses des



particules dans toutes les dimensions, qui limite les particules pour ne pas se déplacer trop rapidement [Papacostantis 2005].

Une deuxième approche pour le contrôle de la convergence dans PSO présente l'utilisation d'un coefficient de constriction χ [Clerc 2002]. L'équation, de mise à jour de la vitesse, est modifiée après l'introduction du coefficient de constriction comme suit :

$$v(t+1) = \chi(w * v(t) + \varphi_1 * (p(t) - x(t)) + \varphi_2 * (g(t) - x(t)))$$

L'algorithme de PSO a été analytiquement étudié et amélioré de plusieurs manières. Un certain nombre de variations conventionnelles de PSO sont présentées pour différents buts:

- ▶ PSO discret est une version de PSO conventionnel pour l'optimisation dans l'espace discret [Kennedy 1997].
- ▶ PSO coopératif sépare le vecteur d'entrée du problème à travers un certain nombre d'essaims; chaque essaim optimise sa propre partie du vecteur [Krohling 2002].
- ▶ Une autre variation concernant les essaims multiples est la niche PSO qui emploie un certain nombre secondaire d'essaims pour localiser et découvrir des solutions optimales multiples simultanément [Brits 2002].
- ▶ Plusieurs variations de PSO sont développées en utilisant un hybride entre les stratégies évolutionnaires et l'optimisation par essaim de particules en combinant l'idée de la sélection et de la reproduction évolutionnaires dans le PSO conventionnel afin d'améliorer le résultat d'optimisation [Reynolds 2003].
- ▶ Un PSO hiérarchique est une variation de PSO qui utilise le concept dynamique de l'hierarchie d'arbre basé sur la performance de chaque particule dans l'essaim pour définir une structure de voisinage [Janson 2003].
- ▶ Un essaim de Kalman combine l'utilisation du filtre de Kalman pour mettre à jour les positions de particules dans le PSO conventionnel. Il augmente l'exploration et la capacité de trouver de bonnes solutions rapidement [Monson 2004].
- ▶ PPSO (Perceptive PSO) est une nouvelle extension qui fonctionne dans l'espace de recherche de n+1 dimensions. Au lieu d'échanger directement l'information entre les particules dans leurs voisinages, chaque individu a une gamme finie de perception de sorte qu'il puisse observer l'espace de recherche, qui est l'environnement de l'essaim, et percevoir les positions approximatives d'autres individus dans sa marge de perception pendant que les insectes sociaux observent le monde et d'autres individus par des sens. Ainsi, les particules dans l'algorithme PPSO sont attirées aux meilleures positions dans l'espace de recherche qu'elles perçoivent et aux voisins qu'elles perçoivent. La dimension supplémentaire représente la performance fondamentale des particules à leurs positions dans l'espace de n-dimensions [Kaewkamnerdpon 2005].

6.1.4 La topologie de voisinage

Il est évident que le modèle de communication entre les particules affectera finalement la façon par laquelle les particules se déplacent dans l'espace de recherche. L'information partagée entre les particules permet à l'exploration de l'espace de recherche d'être de différentes manières. La topologie est une limite qui se rapporte à un modèle par lequel les



particules communiquent l'un avec l'autre. Les topologies les plus utilisées sont [Papacostantis 2005]: Meilleur Global, Meilleur Local et Von Neumann (voire la figure 2.12) :

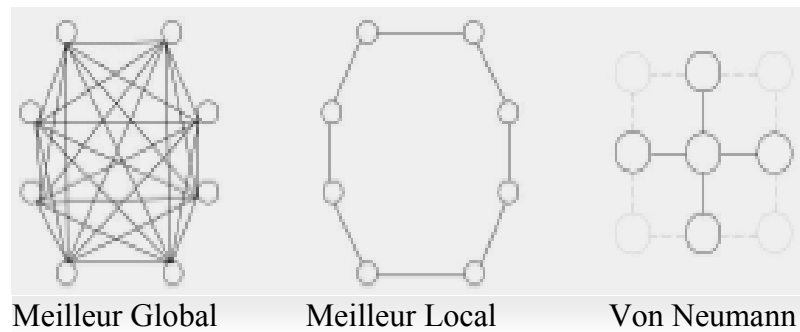


Figure 2.12. Les différentes topologies de voisinage

- **Meilleur Global (GBest):** Toutes les particules communiquent les unes avec les autres, formant un réseau social, entièrement reliées ensemble. Avec cette topologie tous les mouvements de particules sont affectés par leur propre meilleure solution personnelle et la meilleure solution globale. Cette dernière forme la meilleure solution de l'essaim entier [Papacostantis 2005]. Quoiqu'il soit le plus simple des structures de voisinage, GBest n'est pas le plus souvent la meilleure structure d'exécution. L'essaim entier se déplace vers l'endroit de la meilleure particule dans l'espace de n -dimensions, qui peut avoir comme conséquence la convergence prématurée sur les solutions sous-optimales [Franken 2004]
- **Meilleur Local (LBest):** La structure LBest réduit la mesure d'inter-connectivité entre les particules [Franken 2004]. Une taille de voisinage est définie pour cette topologie, qui détermine le nombre de particules avec lesquelles chaque particule peut communiquer et partager l'information. Si une taille de voisinage est 3, par exemple, des voisinages de 3 particules sont constitués en choisissant les deux voisins adjacents de chaque particule dans l'espace variable. Avec cette topologie tous les mouvements de particules sont affectés par leur propre meilleure solution personnelle et la meilleure solution locale. La meilleure solution locale forme la meilleure solution dans le voisinage au quel appartient la particule [Papacostantis 2005]
- **Von Neumann:** (présenté par Kennedy et Mendes [Kennedy 2002]), la structure de voisinage de Von Neumann étend la grille à une dimension de la structure de LBest en une grille à deux dimensions dans l'espace variable, permettant le partage d'une information plus complexe de comportement, mais évitant les pièges de GBest [Franken 2004].

Tout en se déplaçant, les particules enregistrent la meilleure position qu'elles ont découverte précédemment. Elles communiquent avec leurs voisins et apprennent, d'eux, la meilleure solution locale. PSO est basé sur les concepts de l'interaction sociale ou plus exactement, la tendance d'un individu de faire à sa guise, par opposition à sa tendance de suivre la manière de son groupe. À chaque étape, la direction de déplacement des particules est conduite par trois facteurs: d'abord, sa propre vitesse, deuxièmement, sa tendance de retourner à la meilleure solution qu'elle a découverte précédemment, et troisièmement, la



tendance d'aller vers la meilleure solution découverte par ses voisins. Ces principes sont illustrés dans la figure 2.13 [Bourquin 2004].

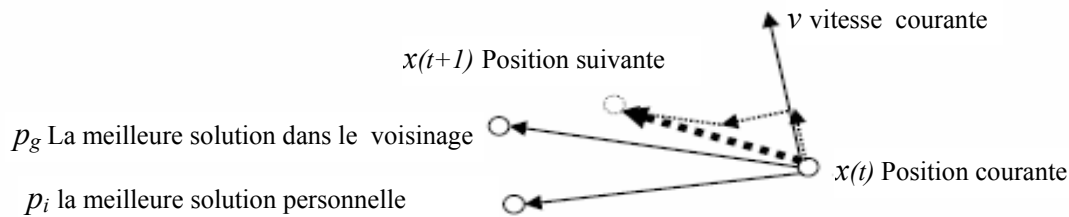


Figure 2.13. Le principe de PSO

Dans le voisinage social, les particules sont associées à d'autres particules du commencement et leur rapport est maintenu dans tout le processus. Un voisinage géométrique est défini selon "la proximité" des particules courantes dans l'espace de paramètre. Dans ce cas-ci, les distances entre les particules doivent être recalculées à chaque itération. L'avantage habituellement mentionné du voisinage social est sa charge de calcul inférieur comparé au voisinage géométrique [Bourquin 2004].

6.2 L'algorithme PPO

Dans cette section, nous présentons le principe de la nouvelle extension de l'algorithme PSO inventée par Silva et al. en 2003 en se basant sur leur article [Silva 2003] comme une référence principale.

Silva et al. [Silva 2003] présentent une nouvelle approche (PPO : Predator Pray Optimization) à l'exploration et à l'exploitation d'équilibrage dans PSO, en présentant une deuxième population des particules, qu'ils appellent "*des prédateurs*". Les prédateurs ont un comportement dynamique différent des particules d'essaim (appelées "*proies*"). Ces prédateurs sont attirés aux meilleurs individus dans l'essaim, alors que les autres particules (proies) sont repoussées par leur présence. Contrôlant la puissance et la fréquence des interactions entre les prédateurs et les proies, ils ont pourraient influencer l'équilibre entre l'exploration et l'exploitation et maintenir une certaine diversité dans la population, même lorsqu'elle approche la convergence, de ce fait réduire le risque de convergence aux optimums locaux. Ils ont également présenté et discuté les résultats expérimentaux obtenus en comparant ce modèle aux variantes de poids d'inertie et du coefficient de constriction de PSO sur un ensemble de fonctions de repère.

6.2.1 Motivations de cet algorithme

La motivation de Silva et al. [Silva 2003], pour développer le modèle de prédateur-proie est principalement de présenter un mécanisme pour créer la diversité dans l'essaim à tout moment pendant l'exécution de l'algorithme, pas selon le niveau de la convergence déjà réalisé. Ceci permettrait "l'évasion" des particules même lorsque la convergence de l'essaim autour d'un optimum secondaire local s'était déjà produite. La seconde, et moins pratique, la motivation était de maintenir le principe d'intelligence d'essaim derrière l'algorithme. D'autres mécanismes pourraient peut-être avoir été employés au même effet, mais il a semblé plus



approprié de présenter un mécanisme qui pourrait également être mis en application comme comportement distribué dans l'essaim. Le modèle prédateur-proie est inspiré de la chasse des animaux groupés en bandes par un ou plusieurs prédateurs (figure 2.14). Une fois chassés, les animaux ont plus de difficulté à rester autour de leurs endroits plus préférables et doivent rechercher d'autres endroits. C'est l'effet que Silva et al. veulent modeler dans un algorithme, où les pâturages meilleurs métaphoriques sont les optimums secondaires locaux des fonctions [Silva 2003].

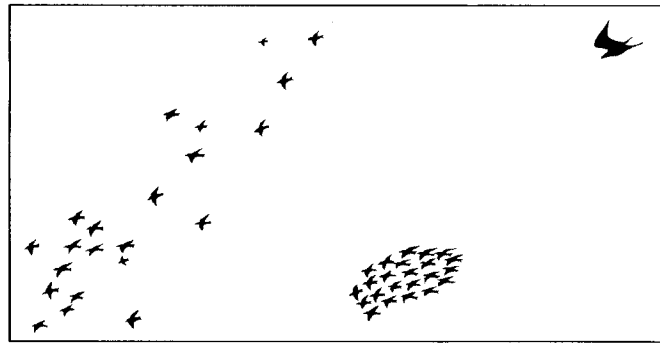


Figure 2.14. La relation Prédateur-Proie dans l'essaim de particule.

6.2.2 Principe de fonctionnement

Silva et al. ont utilisé dans le développement de l'algorithme de prédateur-proie, seulement un seul prédateur. L'objectif du prédateur est de poursuivre le meilleur individu dans l'essaim, c.-à-d. l'individu qui a trouvé le meilleur point dans l'espace de recherche correspondant à la fonction étant à optimiser. Les équations de mise à jour, de la vitesse et de la position du prédateur, sont [Silva 2003]:

$$\begin{cases} V_p(t) = \varphi_4 (X_g(t-1) - X_p(t-1)) \\ X_p(t) = X_p(t-1) + V_p(t) \end{cases}$$

- ▶ φ_4 est un nombre aléatoire distribué entre 0 et une limite supérieure
- ▶ X_p est la position du prédateur
- ▶ X_g est la position actuelle de la meilleure particule dans l'essaim.

La limite supérieure sur φ_4 permet de contrôler comment le prédateur "attrape" rapidement le meilleur individu.

L'influence du prédateur sur un individu dans l'essaim est contrôlée par une probabilité φ_{3ij} de "peur", qui est la probabilité qu'une particule changeant sa vitesse dans une des dimensions disponibles due à la présence du prédateur. Pour une certaine particule i s'il n'y a aucun changement de la vitesse dans une dimension j les équations de mise à jour de la vitesse et de la position sont toujours, comme suit:

$$\begin{cases} v_{ij}(t) = wv_{ij}(t-1) + \varphi_{1ij}(p_{ij} - x_{ij}(t-1)) + \varphi_{2ij}(p_{gi} - x_{ij}(t-1)) \\ x_{ij}(t) = x_{ij}(t-1) + v_{ij}(t) \end{cases}$$



Où : φ_{1ij} et φ_{2ij} sont des nombres aléatoires distribués entre 0 et une limite supérieure et différents pour chaque dimension dans chaque individu. Le paramètre w est un poids décroissant linéaire d'inertie.

Mais, si le prédateur influe la vitesse d'une particule dans une dimension j , les équations de mise à jour deviennent :

$$\begin{cases} v_{ij}(t) = wv_{ij}(t-1) + \varphi_{1ij}(p_{ij} - x_{ij}(t-1)) + \varphi_{2ij}(p_{gi} - x_{ij}(t-1)) + \varphi_{3ij}D(d) \\ x_{ij}(t) = x_{ij}(t-1) + v_{ij}(t) \end{cases}$$

Le quatrième terme, dans la nouvelle équation de mise à jour de la vitesse, mesure l'influence répulsive du prédateur en ajoutant à la vitesse une valeur qui est une fonction de la différence entre la position du prédateur et la particule. d est la distance euclidienne entre le prédateur et la proie. $D(x)$ est une fonction de distance décroissante exponentielle:

$$D(x) = ae^{-bx}$$

$D(x)$ fait l'influence du prédateur se développer exponentiellement avec la proximité. L'objectif de son utilisation est de présenter plus de perturbation dans l'essaim quand les particules sont plus proches du prédateur, qui arrive habituellement quand la convergence se produit. Quand la distance est plus grande (par exemple pendant la phase initiale d'exploration de l'essaim, quand w est encore grand), l'influence du prédateur est plus petite et la dynamique habituelle d'essaim prend le contrôle.

Les paramètres a et b définissent la forme de la fonction D : a représente l'amplitude maximale de l'effet du prédateur sur une proie et b permet de contrôler la distance à laquelle l'effet est encore significatif.

L'effet du prédateur a été conçu pour tirer l'avantage de l'utilisation de w comme paramètre d'inertie dans les équations de mise à jour d'essaim. L'idée est d'abaisser les valeurs de w de ce fait forçant une convergence plus rapide, tout en comptant sur le prédateur pour maintenir la diversité de population.

7. Conclusion

Ce chapitre a été consacré à expliquer le principe de base de quelques algorithmes utilisés pour la résolution des problèmes complexes, où les méthodes exactes ne peuvent pas être appliquées. On a détaillé aussi le principe de fonctionnement des algorithmes évolutionnaires (les algorithmes génétiques, les stratégies d'évolution, la programmation évolutionnaires et la programmation génétiques). Ensuite, on a arboré la nouvelle extension des AEs qui sont les algorithmes co-évolutionnaires inspirés des relations de co-évolution entre les différentes populations au sein d'un écosystème. Enfin, on a présenté les deux algorithmes PSO et PPO, qui sont la base de notre travail.





Chapitre 3: Introduction Au Traitement D'images

1. Introduction

Les premières applications des images numériques remontent aux années 1920 et ne concernent que les premiers pas en transfert d'images et en correction-amélioration d'images. Transmission d'images de journaux par câble sous marin entre Londres et New York en 1921 (une image transmise en trois heures) était par codage au départ et décodage à l'arrivée par une imprimante spécialisée. Chaque niveau de gris étant codé par un caractère particulier (1921 : 5 niveaux de gris, 1929 : 15 niveaux de gris). En 1929, c'était le premier brevet déposé sur la reconnaissance optique de caractères [Desachy 2001].

Les potentialités du traitement numérique des images digitales pour le transfert et l'amélioration des images sont apparues avec le développement des grands ordinateurs et surtout avec les nécessités des programmes de recherche spatiale.

L'imagerie numérique trouve ses origines dans l'histoire de la conquête spatiale au début des années soixante lorsque la NASA eu besoin dans le cadre de la mission Ranger 7, de réaliser des cartes d'alunissage, à partir d'images vidéo de la lune, afin de préparer les futures missions Appolo [Web 13]. Puis l'explosion des applications dans tous les domaines... mais toujours pas de méthodologie universelle pour automatiser par ordinateur la vision humaine...d'où de nombreuses recherches dans ce domaine en pleine expansion [Desachy 2001].

L'imagerie numérique appliquée à la médecine trouve pour sa part ses origines à la fin des années 60 avec l'apparition de la radiographie numérique, suivie à la fin des années 70 par la tomodensitométrie à rayon X. les années 80 furent marquées par l'introduction de la résonance magnétique nucléaire et la fin des années 90 par l'imagerie PET ou SPECT [Web 13].

Dans ce chapitre on va présenter, quelques notions de base des images numériques. Ensuite, on va montrer les étapes les plus importantes du processus de traitement des images numériques. Enfin, le chevauchement de traitement d'images avec les algorithmes évolutionnaires, et avec les algorithmes co-évolutionnaires, est présenté en donnant quelques exemples des travaux réalisés précédemment.



2. Notions de base

2.1 Définition d'une image

Une image est la représentation d'une scène acquise à l'aide de systèmes de production d'images (appareils photographique, caméra, radiographes, scanner, sonar,.....). Sa forme peut être analogique (ex: photographie, vidéo..) ou numérique (images numérisées suivant divers formats, images compressées ou non..) ou obtenues par des capteurs fournissant des images numérisées) et dans ce cas un traitement par ordinateur est possible [Desachy 2001].

Sonka et al [Sonka 1993] donne un aperçu dans ce que nous comprendrons pour être une image. Il dit que l'image "peut être modélée par une fonction continue de deux ou trois variables; dans le cas simple les arguments sont les coordonnées (x, y) dans le plan, alors que si les images changent à temps une troisième variable t peut être ajouté. Les valeurs de fonction d'image correspondent à la luminosité aux points de l'image; ceci nous permet d'éviter la description du processus très compliqué de la formation d'image." [John 1999]

2.2 Les images numériques

L'opération de numérisation (ou digitalisation); est une opération qui permet de transformer une image analogique (signal analogique ou continu) en une image numérique ou discrète.

Convertissant une image continue $a(x, y)$ à sa représentation numérique $b(m, n)$ exige le processus d'échantillonnage [Young 1998]. On désigne sous le terme d'image numérique toute image (dessin, icône, photographie ...) acquise, créée, traitée, stockée sous forme binaire (suite de 0 et de 1) [Web 14]:

- Acquisées par des dispositifs comme les scanners, les appareils photo ou caméscopes numériques, les cartes d'acquisition vidéo (qui numérisent directement une source comme la télévision).
- Créées directement par des programmes informatiques, via la souris, les tablettes graphiques ou par la modélisation 3D (ce que l'on appelle par abus de langage les «images de synthèse»).
- Traitées (grâce à des outils informatiques). Il est facile de les modifier en taille, en couleur, d'ajouter ou supprimer tel ou tel élément, d'appliquer des filtres variés, etc.
- Stockées sur un support informatique (disquettes, disques durs, CD-ROM ...)

L'image numérique est constituée d'un ensemble de points discrets. Chacun de ces points se voit affecté une intensité lumineuse permettant de définir sa couleur. On l'oppose aux images optiques constituées d'une suite *continue* de tons. Toute image est initialement issue d'un système optique. Pour transformer cette image en une image numérique il faudra [Web 13]:



- a) Couper le continu en autant de mesures discrètes : on parle d'*échantillonnage*
- b) Affecter à chacun de ces points de mesure une intensité lumineuse qui traduira la couleur : on parle de *quantification*.

L'aspect final de l'image et sa capacité à traduire une certaine réalité physique dépendra donc des capacités d'échantillonnage et de quantification du système de numérisation. Si ces dernières sont médiocres, l'image numérique ne donnera qu'une représentation médiocre de la réalité.

Au terme de cette numérisation l'image pourra être représenté sous la forme d'un tableau à 2D, chaque cellule de ce tableau contenant une valeur numérique qui représente l'intensité du point de coordonnées (x, y) dans l'image. Chaque case définit un point de l'image que l'on appellera PIXEL (contraction du terme anglo-saxon PICTURE ELEMENT ou élément de l'image en français).

2.3 Codage des images numériques

2.3.1 L'image matricielle (ou image bitmap)

Il s'agit d'images pixellisées (appelées aussi *images raster*) [Krempp 2005], Elle est composée comme son nom l'indique d'une matrice (tableau) de points, appelés pixels (figure 3.1), chacun de ces points possédant une ou plusieurs valeurs décrivant sa couleur [Web 14].

Avec ce type d'image il est possible de représenter des scènes complexes (photographies), mais avec une perte de qualité si on agrandit trop l'image [Krempp 2005].

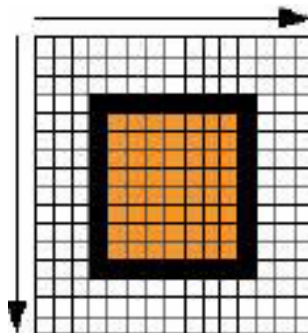


Figure 3.1. Image matricielle.

2.3.2 L'image vectorielle

Les images vectorielles sont des représentations d'entités géométriques (cercle, un rectangle ou un segment) [Krempp 2005]. Le principe est de représenter les données de l'image par des formules géométriques qui vont pouvoir être décrites d'un point de vue mathématique (figure 3.2). Cela signifie qu'au lieu de mémoriser une mosaïque de points élémentaires, on stocke la succession d'opérations conduisant au dessin. Le dessin est mémorisé par l'ordinateur comme « une droite tracée entre les points (x_1, y_1) et (x_2, y_2) », puis « un cercle tracé de centre (x_3, y_3) et de rayon 30 de couleur rouge », etc [Web 14].



Ce type d'image permet de représenter des scènes simples et leur faire subir des transformations (agrandissement, rotation) sans perte de qualité [Krempp 2005].

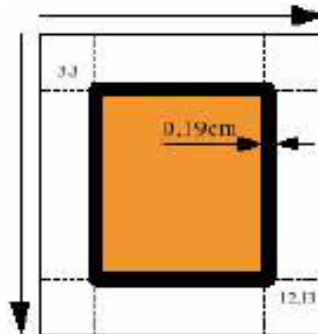


Figure 3.2. Image vectorielle

2.4 Définition et résolution

La définition : c'est la taille de l'image en pixels: une image possédant 640 pixels en largeur et 480 en hauteur aura une définition de 640 pixels par 480, notée 640x480[Krempp 2005].

La résolution : détermine par contre le nombre de points par unité de surface, exprimé en *points par pouce* (**PPP**, en anglais **DPI** pour *Dots Per Inch*); un pouce représentant 2.54 cm [Krempp 2005]. Ce paramètre est défini lors de la numérisation, et dépend principalement des caractéristiques du matériel utilisé. Plus le nombre de pixels par unité de longueur de la structure à numériser est élevé, plus la quantité d'informations, qui décrit cette structure, est importante et plus la résolution est élevée [Web 14].

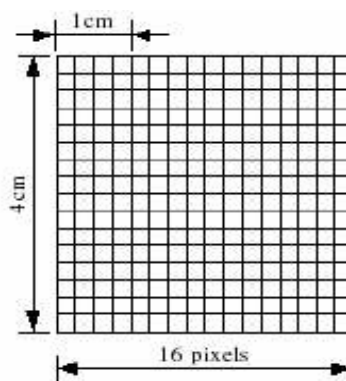


Figure 3.3. La résolution d'une image

La capacité d'une image numérique à traduire la réalité physique d'un objet dépend donc [Web 13] :



- a) Du nombre de pixels par ligne et par colonne pour une taille physique fixée. On parle alors de « *résolution spatiale* »

Résolution : Nombre de pixels/ Dimension physique de l'image (exprimée en Pouce).

- b) Du plus petit écart de couleur codable entre deux variations d'intensité. On parle alors de « *résolution colorimétrique* ». Plus cette dernière est élevée plus l'image numérique reflètera de petites variations d'intensité.

Les 3 images ci-dessous montre les conséquences de la quantification sur une image codée sur:

- a) 24 bits à gauche
b) 8 bits au centre
c) 1 bit à droite



Figure 3.4. Différentes quantifications d'une image

2.5 Avantages et inconvénients de l'image numérique

Les avantages d'une image numérique par rapport à l'image analogique [Bonnet 2004] sont :

- ▶ Plus facile à manier, à stocker
- ▶ Grande dynamique (8 bits → 12 bits)
- ▶ Directement utilisable pour traitement, analyse, quantification
- ▶ Moins coûteuse

Le principal inconvénient est que la résolution un peu plus faible [Bonnet 2004].

2.6 Format d'image en mémoire

2.6.1 Images Noir et blanc (monochromes)

Ces images sont dites en niveaux de gris, car on ne prend pas en compte ici la couleur mais seulement l'intensité lumineuse [Desachy 2001]. Parmi ces images on peut trouver:



- **Images binaires :**

Est une image rectangulaire dont les éléments valent 0 ou 1. Lorsque l'on visualise une telle image, les 0 sont représentés par du noir et les 1 par du blanc [Burel 2001] .

- **Images en niveaux de gris :** appelées aussi images d'intensité

Dans ce cas on dispose d'une échelle de teintes de gris, et la plupart du temps on dispose de 256 niveaux de gris. Par convention la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale) [Web 14]:



2.6.2 Images couleur

L'œil humain analyse la couleur à l'aide de trois types de cellules photo-réceptrices : les cônes. Ces cellules sont sensibles aux basses, moyennes ou hautes fréquences (rouge, vert ou bleu). En conséquence, on peut synthétiser les couleurs perceptibles par l'œil humain en superposant les 3 couleurs de base [Burel 2001].

Pour représenter la couleur d'un point, on utilise trois nombres r,g et b, compris entre 0 et 1, qui correspondent au dosage des trois couleurs de base : rouge, vert et bleu (on parle alors de la représentation RGB : Red, Green et Blue). Une couleur peut donc être représentée par un point dans un espace à 3D, en portant sur les axes les valeurs de r,g et b[Burel 2001].

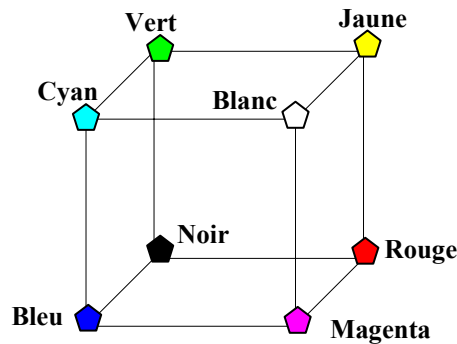


Figure 3.5. Le codage RGB

NB : il existe d'autres représentations des couleurs (espaces de couleur) [Web 14]:

- . CMJN ou CMYK (Cyan Magenta Jaune Noir), utilisé principalement pour l'impression (basé sur synthèse soustractive, contrairement au RGB)

- . TSL ou HSL (Teinte Saturation Luminance), où la couleur est codée suivant l'angle sur le cercle des couleurs.



2.7 Les formats d'image

Comme on a déjà dit, les types de données images sont divisés en deux catégories : vectoriels et matriciels (bitmap)

Le plus simple format d'image matricielle est le BIN (ou RAW). Dans ce cas, le fichier n'a pas d'en-tête et l'utilisateur doit connaître les paramètres de l'image; ce qui n'est pas très utile en pratique. Un meilleur choix est le format PPM qui contient des données de format RAW avec un en-tête rudimentaire. Ce format inclut le format PBM (binaire), PGM (niveaux de gris), PPM (couleur) et PNM (n'importe lequel des formats précédents). L'en-tête contient: (1) un *nombre magique* qui identifie le type de fichier, (2) la largeur et la hauteur de l'image, (3) le nombre de bandes et (4) la valeur maximale d'un pixel (qui détermine le nombre de bits par pixel requis pour chaque bande) [Gagnon 1999].

D'autres formats communs sont : GIF (*graphics interchange format*) et TIFF (*tagged image file formats*). Le format GIF supporte jusqu'à 256 couleurs et permet l'utilisation de la compression sans perte LZW (Lempel-Ziv-Welch). Le format TIFF est plus sophistiqué. Il permet un maximum de 24 bits/pixel et supporte cinq types de compression (avec ou sans perte), incluant RLE (*run-length encoding*), LZW et JPEG (*Joint Photographic Experts Groups*). L'entête TIFF est de dimension variable et contient toutes sortes d'informations sur l'image (par ex., des propriétés de base comme la grandeur de l'image mais aussi des données additionnelles comme la date de création et l'espace de couleur utilisé). TIFF est un des meilleurs formats disponibles, principalement à cause de sa flexibilité. Un autre format d'image largement utilisé sur le Web est le *JPEG file interchange format* (JFIF) [Gagnon 1999].

2.8 L'histogramme d'une image

L'histogramme peut être vu comme une densité de probabilité des niveaux de gris de l'image [Burel 2001]. Pour obtenir un histogramme il faut calculer le nombre de chaque niveau de gris présents dans l'image. Si on travaille sur une image en niveaux de gris avec 256 niveaux alors on obtient 256 nombres entiers [Desachy 2001].

$i \rightarrow H(i) \quad i=0 \dots 255$ ($H(i)$: Nombre d'occurrences du niveau de gris i dans l'image).

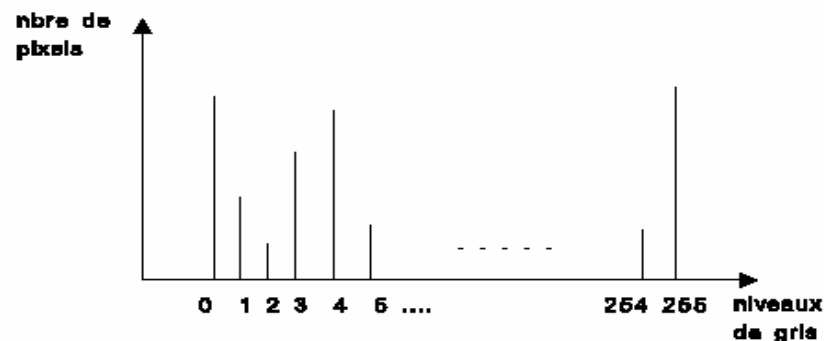


Figure 3.6. Histogramme d'une image



L'étude d'une image numérique débute le plus souvent par le calcul et l'analyse de son histogramme. On obtient ainsi des informations sur [Desachy 2001]:

- la dynamique réelle de l'image (nombre de niveaux de gris réellement utilisés) et donc le contraste disponible....
- Le caractère aléatoire ou non de la distribution des niveaux de gris et la similitude de l'histogramme (ou de morceaux) avec une distribution statistique connue (par exemple une distribution gaussienne).
- la présence de pics significatifs
-

L'algorithme à suivre passe par les deux étapes suivantes [Web 15]:

- Balayer l'image
- Incrémenter le niveau de gris pour chaque pixel

2.9 Connexité dans les images

La notion de connexité entre pixels est une notion particulièrement importante pour la détection des frontières d'objets dans une image et des pixels composant un objet : Deux pixels seront considérés comme connexes (appartenant au même objet donc) s'ils satisfont deux critères [Desachy 2001]:

- La similarité (par exemple même niveau de gris)
- S'ils sont adjacents (voisins)

2.9.1 Voisinage d'un pixel

Un pixel $I(x,y)$ possède quatre voisins horizontaux et verticaux. Si on considère un pixel comme un carré élémentaire, les pixels présentant un coté commun avec $I(x,y)$ sont appelés les 4-voisins de $I(x,y)$. Le pixel $I(x,y)$ possède aussi quatre voisins diagonaux, ce sont les pixels qui ont un sommet commun avec $I(x,y)$. L'ensemble des huit voisins du pixel $I(x,y)$ représentent ce qu'on appelle les 8-voisins (8-voisinage). Ces huit voisins forment la fenêtre 3x3 du pixel $I(x,y)$. On dit aussi que ces pixels sont les 8-adjacents de $I(x,y)$ [Desachy 2001]

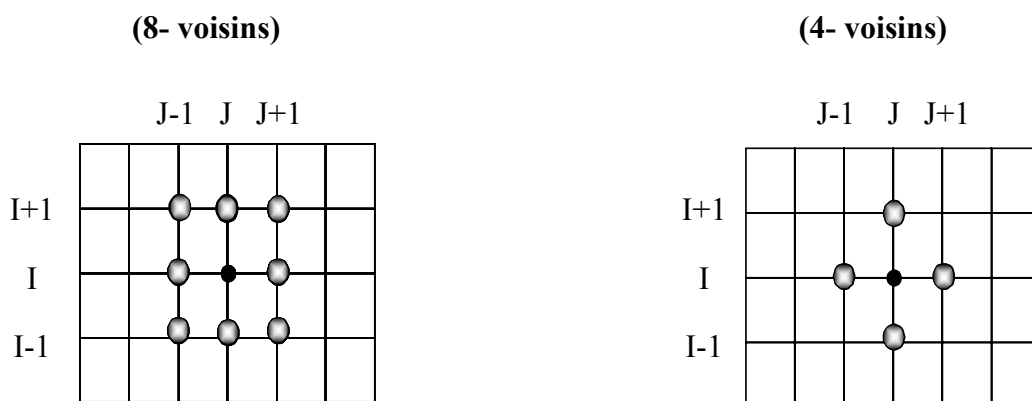


Figure 3.7. Voisinage d'un pixel



2.9.2 Connexité

La connexité est une propriété de liaison entre deux pixels qui fait qu'on les considère comme faisant partie de la même région dans une image. En supposant que deux pixels P et Q vérifient déjà un certain critère de similarité, on peut définir différents types de connexités [Desachy 2001]:

- 4-connexité : Les deux pixels sont deux voisins tels que Q est un des 4-voisins de P
- 8-connexité : Les deux pixels sont deux voisins tels que Q est un des 8-voisins de P
- Connexité mixte : soit P et Q sont 4-voisins, ou bien P et Q sont voisins diagonaux et aucun des 4-voisins communs à P et Q ne sont 4-connexes

2.9.3 Distances entre pixels

Il existe différentes mesures de distances entre deux pixels (x_1, y_1) et (x_2, y_2) [Desachy 2001] :

- Distance Euclidienne : $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
- Distance « City block » : $|x_1 - x_2| + |y_1 - y_2|$
- Distance « Chessboard » : $\max(|x_1 - x_2|, |y_1 - y_2|)$

2.10 Le bruit dans une image

Il en existe quatre sources de dégradation possibles [Web 15]:

- **Bruit lié au contexte de l'acquisition:** L'exemple le plus simple est le bougé, On peut aussi imaginer une modification ponctuelle des conditions d'éclairage, conduisant à une sous (sur) illumination de l'objet observé.
- **Bruit lié au capteur :** si le capteur est de mauvaise qualité ou s'il est mal utilisé, il peut introduire toute sorte de bruits.
- **Bruit lié à l'échantillonnage:** après le passage du domaine continu au domaine discret (phase d'échantillonnage), le bruit peut apparaître dans l'image.
- **Bruit lié à la nature de la scène** (la couverture nuageuse dégrade la qualité des images satellites).



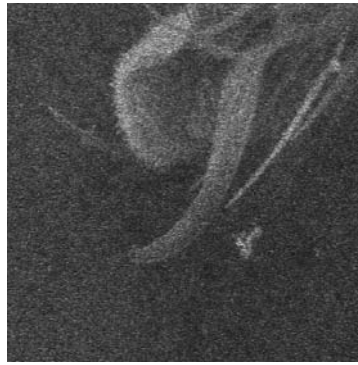


Figure 3.8. Exemple d'image bruitée.

3. Traitement d'images

De nombreuses opérations mathématiques pourront être réalisées sur les images numériques. Elles constituent les bases du traitement d'images [Web 13].

Pearson [Pearson 1991] dit que le traitement d'images est un terme employé pour décrire des opérations effectuées sur des images, dans le but d'accomplir un certain objectif. Ceci peut être une 'définition' très générale et certainement trop vague pour répondre à la question 'c'est quoi le traitement d'images?', mais le manque de précision aide à accentuer la difficulté de définir rigoureusement un ensemble d'applications avec des frontières 'brouillées'. Le traitement d'images, comme beaucoup de disciplines, n'est pas un ensemble exclusif. Awcock et Thomas [Awcock 1995] arguent du fait que le traitement d'images est devenu fortement lié à l'objectif limité de modification des images tels que: (a) corriger les erreurs présentées pendant l'acquisition ou la transmission ('restauration'); ou (b) augmenter pour surmonter la faiblesse du système visuel humain ('réhaussement'). Donc, la discipline du traitement d'images 'pur' peut être brièvement récapitulée comme étant concerné par « *un processus qui prend une image entrée et produit une image modifiée comme résultat* » [John 1999].

Le traitement d'images couvre un grand nombre d'applications [Lina 2004]:

- ❖ L'imagerie grand public: traitement numérique des photos, archivage, ...
- ❖ L'imagerie de synthèse: réalité virtuelle, animation, ...
- ❖ L'imagerie de surveillance (contrôle, monitoring): détection de mouvement, contrôle radar, surveillance maritime par sonar,...
- ❖ L'imagerie médicale: radiologie, tomographie PET, IRM, imagerie optique, ...
- ❖ L'imagerie de type 'vision': contrôle et asservissement en temps réel en robotique, vision artificielle, ...

Les types d'opérations qui peuvent être appliquées aux images numériques, pour transformer une image d'entrée $a[m, n]$ en une image de sortie $b[m, n]$ (ou une représentation



différente), peuvent être classifiés en trois catégories comme montré dans le tableau 3 [Young 1998].

Opération	Caractérisation Générique
<i>Point</i>	La valeur de sortie à une coordonnée spécifique dépend seulement de la valeur d'entrée à cette même coordonnée.
<i>Local</i>	La valeur de sortie à une coordonnée spécifique dépend des valeurs d'entrée dans <i>le voisinage</i> de cette même coordonnée.
<i>Global</i>	La valeur de sortie à une coordonnée spécifique dépend de toutes les valeurs dans l'image d'entrée.

Tableau 3. Types d'opérations d'images.

Les types d'opérations expliqués dans le tableau précédant sont illustrés graphiquement dans la figure 3.9.

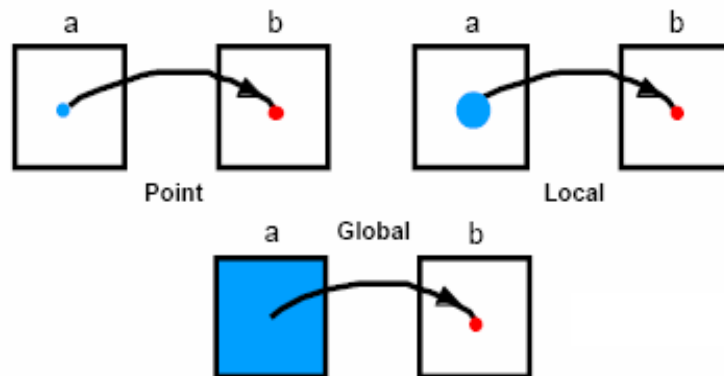


Figure 3.9. Illustration de divers types d'opérations d'image

Dans cette section nous décrivons les opérations qui sont fondamentales au traitement d'images numérique. Ces opérations peuvent être divisées en quatre catégories: opérations basées sur l'histogramme d'image, sur des mathématiques simples, sur la convolution, et sur la morphologie mathématique.

3.1 Restauration

La restauration est une opération objective visant à corriger une image ayant subi une déformation connue [Web 13]. Il s'agit donc de produire une image plus proche de la réalité physique de la scène observée. Le plus souvent, cette étape est la première dans la chaîne de traitements constituant un système de vision [Web 15].



Afin que la restauration soit réalisée avec succès, il est important d'avoir la connaissance précédente quant à la nature de la dégradation, la dégradation d'images peut alors être supprimée en utilisant un processus modelé sur cette *connaissance* a priori de sorte que la dégradation puisse être inversée. Les outils principaux utilisés dans la restauration d'images entrent dans les catégories *des filtres spatiaux, filtres de domaine de fréquence et transformées géométrique* [John 1999]. Pour davantage de lecture voir Awcock et Thomas [Awcock 1995], Pearson [Pearson 1991], Sonka et al [Sonka 1993].

Soit l'image F : le résultat d'une dégradation de l'image idéale I , et \hat{I} l'image restaurée [Burel 2001], le principe de la restauration de l'image dégradée F est illustré dans la figure suivante.

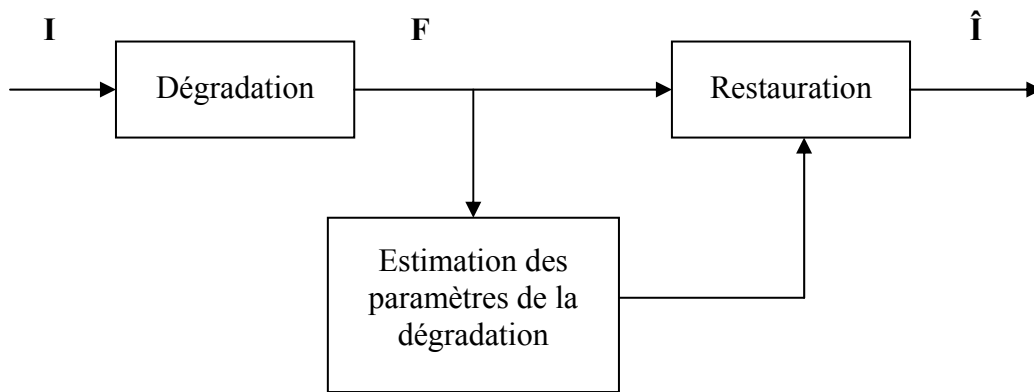


Figure 3.10. Principe général de la restauration d'images

3.2 Binarisation d'une image

Une image binaire est une image codée sur deux niveaux de gris. On binarise généralement une image pour faire apparaître certains détails ou tout simplement pour détacher du fond un objet ou des primitives. Tout le problème de la binarisation réside dans le choix optimal du seuil de binarisation [Web 15].

Algorithme de la binarisation est:

$$\left\{ \begin{array}{l} \text{Si } \text{niveau_gris} (P(x,y)) < \text{seuil_b} \text{ alors} \\ \quad \text{niveau_gris}(P(x,y)) = 0 \\ \text{Sinon } \text{niveau_gris} (P(x,y)) = 255 \end{array} \right.$$

Cas de figure le plus simple : distinction d'un objet du fond (uniforme) dans une image

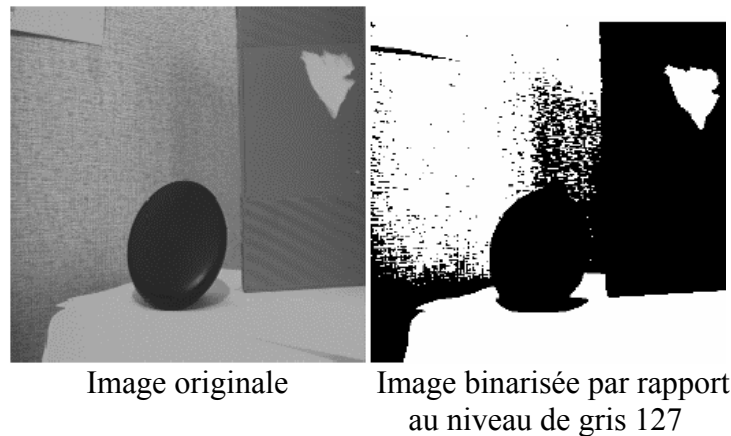


Figure 3.11. Binarisation d'une image.

3.3 Segmentation

Dans l'analyse des objets dans des images, il est essentiel que nous puissions distinguer les objets d'intérêt et " le reste." Ce dernier groupe désigné également sous le nom du fond. Les techniques qui sont employées pour trouver les objets d'intérêt désigné habituellement sous le nom *des techniques de segmentation* – segmentant le premier plan du fond. Les deux techniques les plus communes sont : *le seuillage* et *la détection des bords*. Il est important de comprendre que [Young 1998]:

- ▶▶ Il n'existe pas de technique universelle applicable de segmentation qui fonctionnera pour toutes les images, et,
- ▶▶ Aucune technique de segmentation n'est parfaite.

La segmentation est une des tâches les plus importantes de la phase de prétraitement. L'identification d'objets réels, de pseudo-objets et d'ombres, ou la recherche de tout élément d'intérêt présent dans l'image nécessite une forme de segmentation [Gagnon 1999].

Fondamentalement, la segmentation est un processus qui consiste à découper une image en régions connexes présentant une homogénéité selon un certain critère, comme par exemple la couleur [Baillie 2003]. L'union de ces régions doit redonner l'image initiale [Skarbek 1994]. En pratique, les intensités des pixels composant une région ne sont pas identiques et ces fluctuations constituent le bruit d'image [Tabisa 1998].

La technique de seuillage est basée sur un concept simple. Un paramètre θ appelé *le seuil d'intensité* est choisi et appliqué à l'image $a[m, n]$ comme suit [Young 1998]:

$$\left\{ \begin{array}{l} \text{Si } a[m, n] \geq \theta \quad a[m, n] = \text{object} = 1 \\ \text{Sinon } a[m, n] = \text{background} = 0 \end{array} \right.$$

Cette version de l'algorithme suppose que nous sommes intéressés par les objets légers sur un fond foncé. Pour les objets foncés sur un fond clair nous utiliserons:



$$\begin{cases} \text{Si } a[m,n] < \theta & a[m, n] = \text{object} = 1 \\ \text{Sinon } & a[m, n] = \text{background} = 0 \end{cases}$$

Il y a deux manières de caractériser une région : par contours et par homogénéité (par exemple : même couleur, même niveau de gris, même texture, ...) [Cochard 1998]. Les contours sont les lieux de variations significatives de l'information niveaux de gris. De plus, la transition étant stricte, un contour doit être une chaîne de pixels d'épaisseur 1. Cette restriction sur la nature du contour a été imposée dans un premier temps pour des raisons de formalisation mathématique [Web 15].

La notion de contour étant reliée à celle de variation, il est évident qu'une telle définition nous amène tout naturellement vers une évaluation de la variation en chaque pixel. Une variation existera si le gradient est localement maximum ou si la seconde dérivée (à définir dans un espace bidimensionnel) présente un passage par zéro. Les principaux algorithmes connus (Sobel, Prewitt, Kirsh, Canny, Dérivée, ...) se focalisent sur ce premier aspect du contour. Il existe moins de travaux sur la formalisation de la deuxième partie consistant à passer d'une mesure locale de variation à des chaînes de points d'épaisseur 1. C'est pourtant cette deuxième partie qui fait souvent la différence et la qualité visuelle d'un résultat [Web15].

3.4 Analyse

Le but de l'analyse d'images est d'extraire des informations concernant des dispositifs existants dans l'image (extraction des dispositifs), et de classer chaque dispositif à un groupe d'objets spécifique (classification de modèle).

- **Extraction De Dispositif.**

L'extraction de dispositif cherche à identifier des dispositifs ou des caractéristiques, des objets contenus dans une image. Ces caractéristiques peuvent être employées pour décrire l'objet, ou des attributs de l'objet, tels que la forme ou la couleur. Une liste descriptive de l'objet est produite, connu comme *vecteur de dispositif*, qui est alors employé dans la tâche suivante de la classification de modèle [John 1999].

- **Classification De Modèle.**

La tâche 'finale' du traitement d'images est celle de la classification de modèle. Ce problème consiste à prendre un objet dans une image avant d'essayer de le classer en décidant à quel groupe spécifique d'objets il appartient. Car il y a un certain nombre de choix possibles des groupes, dont le problème à choisir surgit [John 1999].

Awcock et Thomas [Awcock 1995] disent qu' " il y a trois approches principales pour la classification de groupe : (a) la classification basée sur les statistiques se fonde sur la définition d'un ensemble de règles de décision basées sur la théorie statistique standard. (b) la classification syntactique de modèle utilise les structures fondamentales des modèles elles-mêmes. (c) Les approches alternatives emploient des architectures telles que les réseaux de neurones qui peuvent être formés pour associer correctement des modèles d'entrée". Pour davantage lecture voir Awcock et Thomas [Awcock 1995], Pearson [Pearson 1991], Sonka et al. [Sonka 1993].



3.5 Compression

La compression d'images a pour objectif de réduire le volume de données nécessaire au codage d'une image, pour des applications de transmission ou de sauvegarde [Burel 2001]. Dans le traitement d'images une reconstruction fidèle n'est pas toujours exigée à condition que la compression de données ne cause pas *les changements* cruciaux d'une image (voir [Sonka 1993]).

Le taux de compression est spécifié en utilisant un coefficient de compression CR (*compression rate*), défini comme étant le rapport des dimensions du fichier non compressé et du fichier compressé. Les meilleurs procédés de compression sont ceux qui offrent le meilleur rapport entre un CR élevé et les informations essentielles retenues [Gagnon 1999].

Il existe deux types de méthodes de compression : avec ou sans perte d'information. Avec les méthodes sans perte, aucune donnée n'est perdue lors de la phase de compression. Ces méthodes sont nécessaires pour certaines applications telles que les images médicales, pour lesquelles des considérations légales nécessitent que les données compressées ne présentent aucune information incomplète. Pour des images complexes, il est très difficile d'obtenir un CR plus élevé que 3. Pour les images simples, les méthodes de compression de données sans perte peuvent donner une valeur de CR plus élevée [Gagnon 1999].

Les méthodes de compression avec perte impliquent la perte d'une certaine quantité de données; par conséquent, l'image finale ne comporte pas toutes les informations initiales qui étaient présentes dans l'image originale. Avec les techniques de compression avec perte, il est possible d'obtenir un CR de 20 pour des images complexes et pouvant atteindre 200 pour des images simples [Gagnon 1999].

3.6 Recalage

Le recalage est une opération qui permet de mettre en correspondance deux ou plusieurs images prises à différents moments, depuis différents capteurs ou à partir de différents points de vue. Les perturbations dans l'image qui peuvent être compensées par des techniques de recalage sont des transformées spatiales 2D (par ex., rotations, changements d'échelle, etc.). D'autres variations causées par des changements de luminosité, un mauvais alignement dû au changement de perspective et les mouvements d'objets ne sont pas directement soustraits par le recalage mais compliquent le processus [Gagnon 1999]. Pour davantage lecture voir [Burel 2001]

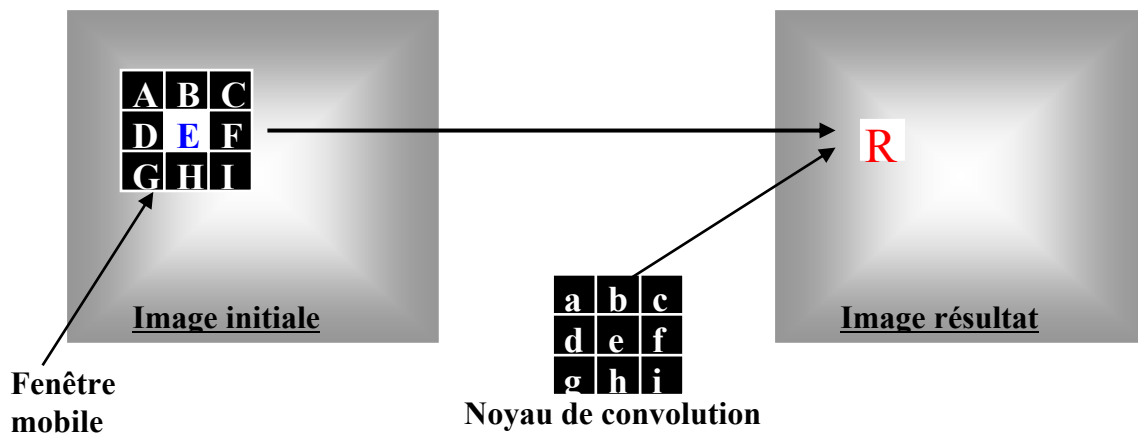
3.7 Filtres de convolution

3.7.1 Définition d'un filtre

Les *opérations de convolution* sont très puissantes, tandis qu'elles sont essentiellement simples, en donnant des moyens de réduire et de lisser une image. La convolution produit une nouvelle image basée sur le résultat d'un filtre de convolution (également connu sous le nom de grain) étant déplacé au-dessus de l'image de source. Le filtre est une grille qui est plus petite que l'image source. Le centre du filtre est le Pixel cible; la nouvelle valeur (R) du Pixel à la position cible est la somme des multiples des valeurs de filtre (K) par leurs valeurs



correspondantes de pixel source (E)[Bridge 1999]. Exemple d'utilisation d'un filtre de convolution est illustré dans la figure suivante [Bonnet 2004]



Résultat du pixel R(i, j):

$$R(i,j)= A*a + B*b + C*c + \dots + H*h + I*i$$

Figure 3.12. L'opération de convolution

3.7.2 Les différents types des filtres

a- Filtrage linéaire

Les filtres les plus simples et les plus faciles à implanter sont fondés sur le filtrage linéaire. Grâce à ces propriétés de nombreux filtres typiquement de type passe-bas (pour le lissage), ont été proposés dans la littérature et appliqués au filtrage d'images [Tabisa 1998].

Le principe du filtrage linéaire est de remplacer la luminance d'un pixel par une combinaison linéaire des luminances des pixels voisins. Cette combinaison linéaire est usuellement représentée par un masque [Cochard 1998]. Le filtrage de l'image I, par le filtre h est une image F dont les luminances sont données par l'équation suivante [Burel 2001] :

$$F(x, y) = \sum_{a,b} h(a, b) \times I(x + a, y + b)$$

En générale, on choisi un filtre de taille impaire afin de pouvoir définir le centre du filtre [Burel 2001]. Par conséquence, ces filtres ont été progressivement remplacés par des techniques de filtrage non linéaires caractérisées par de meilleures performances tant en réduction de bruit qu'en préservation et localisation de contours [Tabisa 1998].

Parmi les filtres linéaires, 3 catégories de filtres sont envisagées [Bonnet 2004]:

- » **Filtres "passe-bas" (Lissage):** ils sont utilisés pour atténuer les détails de l'image qui "tranchent" nettement avec le reste de l'image : les bords, les caractéristiques



particulières notamment. Le résultat de l'application d'un filtre "passe-bas" sera une image "brouillée" ou "trouble".

$$\begin{array}{cc}
 1/9 & \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} & 1/16 & \begin{array}{ccc} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{array}
 \end{array}$$

- » **Filtres "passe-haut"** : contrairement aux précédents, ils sont utilisés pour atténuer les caractéristiques "neutres" et mettre en évidence les détails qui "tranchent".

(Réhaussement)

$$\begin{array}{cc}
 \begin{array}{ccc} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{array} & \begin{array}{ccc} -1 & -1 & -1 \\ -1 & 10 & -1 \\ -1 & -1 & -1 \end{array}
 \end{array}$$

(Détection de bords)

$$\begin{array}{cccc}
 \begin{array}{ccc} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{array} & \begin{array}{ccc} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 1 & 1 \end{array} & \begin{array}{ccc} 0 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \end{array} & \begin{array}{ccc} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{array}
 \end{array}$$

- » **Filtres "passe-bande"** : ils ont peu d'intérêt pour le réhaussement d'images, mais par contre ils sont utilisés dans le cas de la restauration d'images.

b- Filtrage non linéaire

La famille des filtres non linéaire est très vaste. Cette famille regroupe les filtres morphologiques, les filtres homomorphiques, les filtres basés sur l'approche markovienne, les filtres d'ordre ... etc. Une classe peut être très efficace pour une application donnée et s'avérer médiocre pour une autre. Ainsi, la famille des filtres morphologiques est largement utilisée en imagerie médicale. La famille des filtres homomorphiques est plus adaptée pour le réhaussement de contraste, en réduction de bruit multiplicatif. La famille des filtres d'ordre est couramment rencontrée en réduction de bruit [Tabisa 1998].

Le niveau d'un pixel de l'image filtrée est toujours fonction du niveau des pixels environnants, mais le calcul n'est plus basé sur des combinaisons linéaires. Prenons l'exemple le plus classique du filtre médian : les niveaux du pixel et des pixels environnants sont classés par ordre croissant d'importance (figure 3.13); on prend pour niveau du pixel la médiane de cette série [Cochard 1998].





techniques de réhaussement d'images se composent des opérations : *point*, *masque* et *globales*, en utilisant la fréquence et les domaines spatiaux. Les outils principaux tombent sous les catégories: la *modification de niveau de gris*, *image aiguisant* et *lissage d'images*. Pour davantage de lecture voir [Awcock 1995], [Pearson 1991], [Sonka 1993] et [John 1999].

4. *Le traitement d'images et les algorithmes évolutionnaires*

Plusieurs chercheurs ont essayé d'appliquer les algorithmes évolutionnaires dans les différents secteurs de traitement d'images, et ils ont prouvé que ces algorithmes donnent de meilleurs résultats que les méthodes classiques.

Dans [Abbassene 2002], l'application des principes de l'émergence et des systèmes complexes, pour la résolution du problème de la segmentation des images, est réalisée en utilisant les algorithmes cellulaires. Un Algorithme Evolutionnaire (Algorithme Génétique) est utilisé pour la recherche de la règle d'automate cellulaire adéquate à la réalisation de la tâche de segmentation.

Liu et al. [Liu 1997] présentent une nouvelle approche utilisant les agents autonomes évolutionnaires qui peuvent s'auto-reproduit, diffuser, et cesser d'exister pendant l'interaction avec l'environnement qui est l'image numérique. Les systèmes évolutionnaires d'agents autonomes (EAA : Evolutionary autonomous agent) pour des techniques de traitement d'images est un domaine de recherche récemment exploré qui étudie le comportement émergent de l'interaction des agents et de l'image numérique selon un ensemble de règles comportementales. L'approche EAA offre une nouvelle technique alternative à la détection de dispositif d'image qui est robuste et adaptative.

Zhou et Franklin [Zhou 1994] proposent un ensemble d'agents pour la reconnaissance de caractères qui " doivent trouver la nourriture dans un environnement artificiel prédéfini ". Cet environnement se compose d'une matrice de 24x24 Pixel d'une image de caractère, dans laquelle un agent doit apprendre à trouver la nourriture en suivant un chemin le long d'un caractère. Le but de Zhou et Franklin est de réduire de manière significative la quantité de prétraitement, en supprimant presque le besoin d'extraction de dispositif, pendant l'exécution de la reconnaissance des caractères.

Les algorithmes génétiques AGs sont intensivement employés dans tout le traitement d'images [John 1999]. Mitra et al. [Mitra 1998] proposent une nouvelle méthode pour la compression fractale d'images utilisant un algorithme génétique avec le modèle d'élite. La méthode proposée utilise un système de classification plus simple pour des blocs de gamme. Les algorithmes génétiques avec le modèle d'élite sont employés en trouvant le bloc approprié de domaine comme la transformation appropriée pour chaque bloc de gamme.

L'approche de Poli [Poli 1996] consiste à utiliser la programmation génétique pour l'analyse d'images. Cette nouvelle approche est basée sur l'idée que la plupart des tâches de bas niveau d'analyse d'images, à savoir le réhaussement d'images, détection de dispositif et segmentation d'images, peuvent être considérées comme problèmes de filtrage d'images et que la PG peut être employée pour découvrir les filtres optimaux qui résolvent de tels problèmes. Poli essaye d'obtenir des filtres en utilisant la PG qui peut effectuer la segmentation et la détection optimales dans des images réelles complexes en niveaux de gris.

Munteanu et Rosa [Munteanu 2000] ont proposé une nouvelle technique automatique de réhaussement d'images basée sur les algorithmes génétiques (AGs) avec le codage réel. La



tâche de l'AG est d'adapter les paramètres d'une nouvelle extension à une technique locale de réhaussement semblable à la graduation statistique, quant à augmenter le contraste et le détail dans l'image selon un critère objectif de fitness. Ils ont comparé leur méthode à d'autres techniques automatiques de réhaussement, comme les méthodes d'égalisation d'histogramme et d'étirage de contraste. Les résultats obtenus, en termes d'évaluation subjective et objective, montrent la supériorité de leur méthode, ce que nous motive à baser notre travail sur leur approche.

5. *Le traitement d'images et les algorithmes co-évolutionnaires*

Ces dernières années, les chercheurs commencent à s'orienter vers l'application de la nouvelle extension des algorithmes évolutionnaires : les algorithmes co-évolutionnaires, dans le domaine de traitement d'images. Dans cette section, quelques travaux dans ce domaine sont présentés.

Roberts et Claridge [Roberts 2004] ont montré une approche qui recherche à exploiter un type de synergie en utilisant des techniques co-évolutionnaires coopératives pour co-évoluer simultanément un ensemble de constructeurs de dispositif, et un classificateur qui utilise ces dispositifs. Cette approche était réussie sur une variété de niveaux et de problèmes de rotations invariables. Ces techniques n'exigent aucune connaissance spécifique au sujet des objets qu'elles essayent de détecter, et peuvent être employées par des experts en matière de domaine (plutôt que des experts en matière de traitement d'images) comme elles doivent simplement marquer les points de cible pour la formation.

Ils ont montré dans un autre travail [Roberts 2005], comment l'utilisation de l'idée de l'extraction co-évolutionnaire de dispositif en conjonction à l'approche à plusieurs étages peut réduire les exigences informatiques par au moins deux ordres de grandeur, permettant aux gains impressionnants d'exécution de cette technique d'être aisément appliqués à beaucoup de problèmes réels du monde.

Grasemann et Miikkulainen [Grasemann 2005] proposent une méthode basée sur l'algorithme génétique co-évolutionnaire pour évoluer les ondelettes spécialisés pour la compression d'images, en utilisant des images d'empreinte digitale comme domaine d'essai. Ces ondelettes sont comparés aux ondelettes main-conçues actuellement employé par le FBI pour compresser des empreintes digitales. Les résultats prouvent que les ondelettes évolués surpassent uniformément les ondelettes main-conçues. L'utilisation de l'évolution pour adapter des ondelettes aux classes des images peut donc, de manière significative, augmenter la qualité des images compressées.

Veenman et al. [Veenman 2003] ont proposé un algorithme co-évolutionnaire cellulaire pour l'optimisation du modèle. Ils présentent un modèle pour le problème de cluster qui n'a pas besoin de la connaissance au sujet du nombre de clusters a priori. Dans leur projet des agents multiples sont placés dans une grille 2-D régulière représentant l'image, qui impose des relations de voisinage entre eux. Les agents considèrent coopérativement la migration de Pixel d'un agent à l'autre afin d'améliorer l'homogénéité de l'ensemble des régions d'image qu'ils représentent. Si l'union des régions des agents voisins est homogène, les agents forment des alliances. D'autre part, si un agent découvre un sujet déviant, il isole le sujet. Ils ont spécialisé le modèle de cluster pour le problème de segmentation d'images, et ils ont présenté un algorithme co-évolutionnaire cellulaire (CCA) pour l'optimiser d'une manière distribuée.



6. Conclusion

Ce chapitre est centré sur tout ce qui concerne le domaine de traitement d'images numériques. Ce dernier est le processus qui prend une image en entrée et produit une autre image, qui est le résultat d'une telle opération de transformation qui vise à satisfaire la vision humaine. On a essayé aussi de donner un aperçu sur le principe de fonctionnement des opérations de traitement d'images les plus utilisées, tel que : la restauration, la segmentation, le filtrage, le réhaussement de contrastes et d'autres. La dernière section de ce chapitre est consacrée à présenter le chevauchement entre le traitement d'images et les deux types d'algorithmes : évolutionnaire et co-évolutionnaire.





Chapitre 4 : Application Du PSO Et PPO Au Réhaussement D'images

1. Introduction

La production des images numériques avec le bon contraste et détail est une condition forte dans plusieurs secteurs comme la télédétection, analyse d'images biomédicales, détection de défaut, ...etc. Des méthodes qui traitent l'image donnée de sorte que le résultat soit visuellement plus approprié que l'original, s'appellent les techniques de réhaussement d'images [Munteanu 2000]. Les techniques utilisées pour le réhaussement de contraste sont variées, et notre but est d'enrichir ces techniques. Dans ce chapitre, on va présenter l'application de l'algorithme évolutionnaire PSO ainsi que l'algorithme co-évolutionnaires compétitive PPO dans le domaine de traitement d'images, et plus précisément le réhaussement de contraste local dans les images en niveaux de gris.

Pour implémenter ces deux algorithmes et les appliquer au réhaussement de contraste, on a choisi d'utiliser l'environnement de programmation Matlab parce qu'il offre un Toolbox complet de traitement d'images qui permet de faciliter à l'utilisateur de traiter ses images. Les deux algorithmes PPO et PSO sont utilisés comme des techniques de réhaussement de contraste local ayant les objectifs suivants : a) utiliser une technique locale de réhaussement de contraste basée sur une variation de la méthode statistique de graduation [Jain 1991]. b) ne pas utiliser l'interaction avec l'utilisateur, pendant les étapes courantes de l'algorithme, donc la méthode est automatique [Munteanu 2000]. c) utiliser un critère objectif de fitness sans des paramètres externes additionnels [Munteanu 2000].

2. Formulation du problème

Le réhaussement d'images améliore la qualité (clarté) des images pour le visionnement humain. Retirant le flou et le bruit, augmentant le contraste, et découvrant les détails sont des exemples des opérations de réhaussement. Par exemple, une image pourrait être prise d'une cellule endothéliale, qui pourrait être de bas contraste et un peu brouillée. La réduction du bruit et du flou, et l'augmentation de la gamme de contraste ont pu augmenter l'image. Une



image pourrait avoir des secteurs de très haute et très basse intensité, qui masquent des détails [Rapp 1996].

Pour quelques images, une méthode globale de réhaussement, qui est une méthode dans laquelle les pixels dans l'image sont modifiés par une fonction de transformation basée sur la distribution des niveaux de gris dans l'image entière, souvent ne produit pas des résultats satisfaisants [Gonzales 1987]. Dans ces cas, *un procédé local* qui augmente différemment dans différents secteurs de l'image est recommandable.

Les méthodes locales de réhaussement appliquent les fonctions de transformation qui sont basées sur la distribution des niveaux de gris, ou d'autres propriétés, dans le voisinage de chaque Pixel dans une image donnée [Gonzales 1987]. Un exemple d'une méthode locale de réhaussement de contraste est l'égalisation adaptative d'histogramme où chaque Pixel est transformé selon une égalisation d'histogramme exécuté dans *le voisinage de $n*n$* de ce Pixel.

Les deux algorithmes, PPO et PSO, sont des modèles inspirés des écosystèmes naturels, et utilisés pour les problèmes d'optimisation de telles fonctions objectives. Comme il est déjà mentionné dans le chapitre 2, PPO est la nouvelle extension du PSO qui consiste à introduire une nouvelle particule jouant le rôle d'un prédateur, qui a pour but de poursuivre le meilleur élément dans la population, qui a trouvé la meilleure solution pour éviter la convergence prématurée aux solutions sous-optimales locales. Pour appliquer les deux algorithmes PSO et PPO pour le réhaussement de contraste, on doit définir d'abords ce qui suit :

- ▶ Une fonction de transformation de l'image originale doit être précisée.
- ▶ Pour évaluer la qualité de chaque image transformée, on doit trouver une telle fonction objective qui respecte quelques critères.
- ▶ On doit préciser les paramètres utilisés par L'algorithme PSO et l'algorithme PPO afin de mettre à jours les vitesses et les positions des particules et celles du prédateur.

2.1 La fonction de transformation

En traitant une image, il y a une variété de méthodes disponibles pour obtenir les résultats désirés. Généralement, elles peuvent être décomposés en opérations d'histogramme, opérations arithmétiques et logiques, opérations de convolution, opérations dérivées et opérations morphologiques [Bridge 1999].

Les opérations d'histogramme permettent l'équilibre de couleur ou l'intensité dans une image à être réorganisé. Un histogramme d'un certain nombre de Pixel avec leurs intensité est produit pour une image (ceci décrit, en forme graphique, la fonction de distribution de probabilité (PDF) de l'image). Une fonction f est alors générée pour décrire le traçage entre le PDF actuel de l'image et le PDF désiré. Chaque Pixel est transformé par f pour produire une nouvelle image. En utilisant cette méthode, le contraste peut 'être étiré' pour augmenter la définition des dispositifs dans une image. [Bridge 1999].

Les méthodes de réhaussement local appliquent les fonctions de transformation qui sont basées sur la distribution des niveaux de gris, ou d'autres propriétés, dans le voisinage de chaque Pixel dans une image donnée [Gonzales 1987]. Notre application est basée sur la fonction de transformation employée par Munteanu et Rosa [Munteanu 2000], ils ont choisi d'employer une méthode moins longue semblable à la graduation statistique présentée dans



[Gonzales 1987]. La méthode applique à chaque Pixel (x, y) la fonction de transformation suivante:

$$g(x, y) = \left(\kappa \frac{M}{\sigma(x, y) + b} \right) \cdot [f(x, y) - c \cdot m(x, y)] + m(x, y)^a \tag{1}$$

$$0.5 < \kappa < 1.5; a \in \Psi_1, b \in \Psi_2, c \in \Psi_3 \text{ avec } \Psi_1, \Psi_2, \Psi_3 \subset R_+$$

Dans l'équation (1):

- $m(x, y)$ et $\sigma(x, y)$ sont la moyenne des niveaux de gris et l'écart type calculé dans un voisinage centré à (x, y) et avoir n*n pixels,
- M est la moyenne globale de l'image,
- $f(x, y)$ est l'intensité de niveau de gris du pixel (x, y) dans l'image d'entrée,
- $g(x, y)$ est la valeur d'intensité de niveau de gris résultat du pixel, au même endroit.
- $b \neq 0$ tient compte de l'écart type nul dans le voisinage,
- $c \neq 0$ tient compte seulement d'une fraction de la moyenne $m(x, y)$ pour être soustraite du niveau de gris original de Pixel,
- Le dernier terme, dans l'équation, peut avoir un effet d'éclairage et de lissage sur l'image.

Les meilleurs intervalles dans l'équation (1) sont : $\Psi_1 = [0, 1.5]$, $\Psi_2 = [0, 0.5]$ et $\Psi_3 = [0, 1]$ (prouvé dans [Munteanu 2000]).

Dans notre travail, la tâche de PPO et de PSO est de trouver la meilleure image augmentée selon la meilleure combinaison des quatre paramètres: a, b, c, κ . Pour atteindre ce but, la position de chaque particule dans l'essaim, est représenté par un vecteur X de 4 dimensions correspondant aux 4 paramètres.

➤ **Calcul de la moyenne et l'écart type dans le voisinage**

Pour calculer la moyenne et l'écart type des niveaux de gris dans le voisinage de chaque pixel, plusieurs tailles du voisinage ont été définit (voir la section 2.9.1).

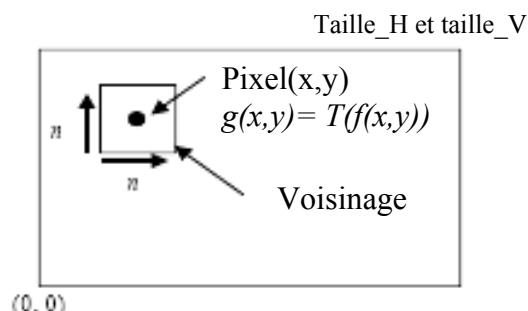


Figure 4.1. Appliquant à chaque Pixel dans l'image l'opération $T(f(x,y))$ dans le voisinage



Dans notre cas, on a choisi le voisinage de 3*3 pixels (c'est-à-dire 8 pixels autour du pixel considéré).

2.2 La fonction de fitness

Afin d'appliquer PPO et PSO, pour le réhaussement de contraste dans des images en niveaux de gris, un critère objectif devrait être précisé. Un bon choix d'un critère objectif est la fonction de fitness suivante (à être maximisée) décrite dans [Munteanu 2000]:

$$F(x) = \log(\log(E(I(x)))) \cdot \frac{n_edgels(I(x))}{H_size \times V_size} \cdot H(I(x)) \tag{2}$$

Dans cette équation :

- $F(x)$ est la fonction de fitness,
- $I(x)$ est l'image originale I avec la transformation T appliquée selon l'équation (1), où les paramètres respectifs : a, b, c , et κ sont donnés par la particule $\mathbf{X} = (a, b, c, \kappa)$.
- $E(I(x))$ est l'intensité des bords détectés avec un détecteur de bords 'Sobel' [Rosin 1997 b], où le détecteur est appliqué à l'image transformée $I(x)$.
- n_edgels dénote le nombre de Pixel de bords détecté avec le détecteur de bord 'Sobel',
- $H(I(x))$ est une mesure de l'entropie dans l'image $I(x)$,
- H_size, V_size sont les tailles respectivement : horizontale et verticale (nombre de Pixel dans chaque direction) de l'image.

Le détecteur de bord Sobel utilisé est un détecteur automatique de seuil [Ray 1991]. La somme d'intensités des bords $E(I)$ inclus dans l'image augmentée est calculée par l'expression suivante [Rosin 1997 b] :

$$E(I) = \sum_x \sum_y \sqrt{\delta h_1(x, y)^2 + \delta v_1(x, y)^2}$$

$$\delta h_1(x, y) = g_1(x+1, y-1) + 2g_1(x+1, y) + g_1(x+1, y+1) - g_1(x-1, y-1) - 2g_1(x-1, y) - g_1(x-1, y+1)$$

$$\delta v_1(x, y) = g_1(x-1, y+1) + 2g_1(x, y+1) + g_1(x+1, y+1) - g_1(x-1, y-1) + 2g_1(x, y-1) + g_1(x+1, y-1)$$

2.2.1 Le détecteur de bords 'Sobel'

Le principe du détecteur de bords 'Sobel' est d'utiliser deux masques (noyaux) de convolution : h_x et h_y (Figure 4.2), qui permettent de détecter respectivement les bords horizontalement et verticalement [Rosin 1997 b].



$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Figure 4.2. Le détecteur de bords 'Sobel'

L'algorithme de Sobel apporte une meilleure insensibilité au bruit [Cochard 1998].

2.2.2 Calcul de l'entropie

Soit une image comportant n lignes et p colonne. On peut interpréter l'histogramme d'une image en termes de probabilités : si f_i est la fréquence d'apparition du niveau i, la probabilité d'apparition de ce niveau dans l'image est donnée par [Cochard 1998] :

$$p_i = f_i / n.p$$

Supposant que l'image est codée sur N niveaux de couleur, on appelle quantité d'information apportée par un niveau donné la grandeur suivante [Cochard 1998]:

$$Q_i = -\log_2(p_i)$$

On peut, sur une image, calculer la quantité d'information moyenne en effectuant une moyenne arithmétique pondérée des quantités d'informations apportées par chaque niveau (avec des coefficients p_i). Le résultat est appelé "*l'entropie*" de l'image (en fait l'entropie au premier ordre car la corrélation entre les pixels n'est pas prise ici en compte) [Cochard 1998]:

$$E = \sum_{i=1}^{N=1} E_i = \sum_{i=1}^{N=1} p_i Q_i = - \sum_{i=1}^{N=1} p_i \log_2(p_i)$$

3. L'application de PSO et PPO pour le réhaussement de contraste

On se rappelle des formules utilisées pour la mise a jours des valeurs des position et celles des vitesses des particules dans l'espace de recherche dans le cas du PSO :

$$\begin{cases} v_{ij}(t) = wv_{ij}(t-1) + \varphi_{1ij}(p_{ij} - x_{ij}(t-1)) + \varphi_{2ij}(p_{gi} - x_{ij}(t-1)) \\ x_{ij}(t) = x_{ij}(t-1) + v_{ij}(t) \end{cases} \quad (3)$$

Où :

$v(t)$: est la vitesse courante ;

$x(t)$: est la position courante ;

$p(t)$: est sa meilleure position précédente ;



$g(t)$: est la meilleure position trouvée par ces voisins ;

$\varphi_{1ij} = c_1 * r_{1ij}$ et $\varphi_{2ij} = c_2 * r_{2ij}$: c_1 et c_2 sont les constantes: cognitive et sociale d'accélération respectivement, et r_{1ij} et r_{2ij} sont des nombres aléatoires distribués entre 0 et 1

w est le poids d'inertie.

Les formules utilisées pour la mise à jour des positions des particules (proies), dans le cas de l'algorithme PPO, sont comme suit :

$$\begin{cases} v_{ij}(t) = wv_{ij}(t-1) + \varphi_{1ij}(p_{ij} - x_{ij}(t-1)) + \varphi_{2ij}(p_{gi} - x_{ij}(t-1)) + \varphi_{3ij}D(d) \\ x_{ij}(t) = x_{ij}(t-1) + v_{ij}(t) \end{cases} \quad (4)$$

Le quatrième terme, dans la nouvelle équation de mise à jour de la vitesse, mesure l'influence répulsive du prédateur en ajoutant à la vitesse une valeur qui est une fonction de la différence entre la position du prédateur et la particule. La fonction $D(x)$ est calculée par la formule suivante :

$$D(x) = ae^{-bx}$$

a représente l'amplitude maximale de l'effet du prédateur sur une proie et b permet de contrôler la distance à laquelle l'effet est encore significatif.

Le prédateur, aussi possède une position dans l'espace de recherche et une vitesse qui lui permet de poursuivre la meilleure particule dans l'essaim, les équations de mise à jour sont :

$$\begin{cases} V_p(t) = \varphi_4(X_g(t-1) - X_p(t-1)) \\ X_p(t) = X_p(t-1) + V_p(t) \end{cases} \quad (5)$$

φ_4 est un nombre aléatoire distribué entre 0 et une limite supérieure

X_p est la position du prédateur

X_g est la position actuelle de la meilleure particule dans l'essaim.

3.1 Choix des paramètres

Pour appliquer les deux modèles d'optimisation PPO et PSO au réhaussement de contraste, on doit indiquer les valeurs des paramètres utilisés pour la mise à jour des positions et des vitesses des particules dans l'essaim, ainsi que ceux du prédateur dans le cas du PPO. Comme on a mentionné précédemment (chapitre 2), PPO combine les idées de PSO avec la stratégie inspirée de l'écosystème prédateur-proie, donc, on doit indiquer les valeurs des paramètres employés par PSO (voir le tableau 4.), ensuite on doit préciser celles ajoutées par la nouvelle technique PPO (voir le tableau 5.). On a choisi ces valeurs des paramètres comme elles sont employées par Silva et al. [Silva 2003], avec un poids d'inertie w qui décroît linéairement entre w_{max} et w_{min} .



Paramètre	Valeur
W_{min}	0.4
W_{max}	0.9
Coefficient de constriction χ	1
Constante cognitive c_1	1.3
Constante sociale c_2	1.3

Tableau 4. Les paramètres de l'algorithme PSO

Paramètre	Valeur
a	1
b	1
p_f	0.02

Tableau 5. Les paramètres de l'algorithme PPO

3.2 Les étapes de l'algorithme PSO

Chaque particule a une vitesse et une position qui sont représentées respectivement par les deux vecteurs: X et Y avec 4 dimensions. Chaque particule X est un vecteur de 4-dimensions représentant une combinaison des variables : a, b, c, k (figure 4.3). Par conséquent, le but de PSO est de trouver la meilleure combinaison des 4 paramètres, de l'équation (1), pour trouver la meilleure transformation de l'image originale.

$$g(x, y) = T(f(x, y); \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{\kappa})$$

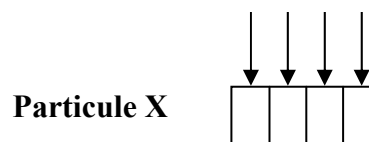


Figure 4.3. Représentation d'une particule

Les étapes de notre application de l'algorithme PSO pour le réhaussement de contraste sont:

1. Créer la population initiale avec des positions aléatoires des particules.
2. Initialiser la meilleure position personnelle (p_i) de chaque particule à sa position initiale.
3. Pour chaque itération :
 - a- Transformer l'image originale selon des valeurs de a, b, c et k données par chaque particule i en utilisant l'équation (1)



- b- évaluer la fitness de chaque particule (basée sur l'image transformée correspondante) utilisant la fonction objective décrite dans l'équation (2)
 - c- trouver la meilleure position globale 'p_g' dans l'essaim, correspondant à la meilleure image réhaussée,
 - d- mettre à jour la position et la vitesse de chaque particule en utilisant les équations (3)
4. Trouver la meilleure position dans l'essaim, qui donne la meilleure image réhaussée, puis l'afficher

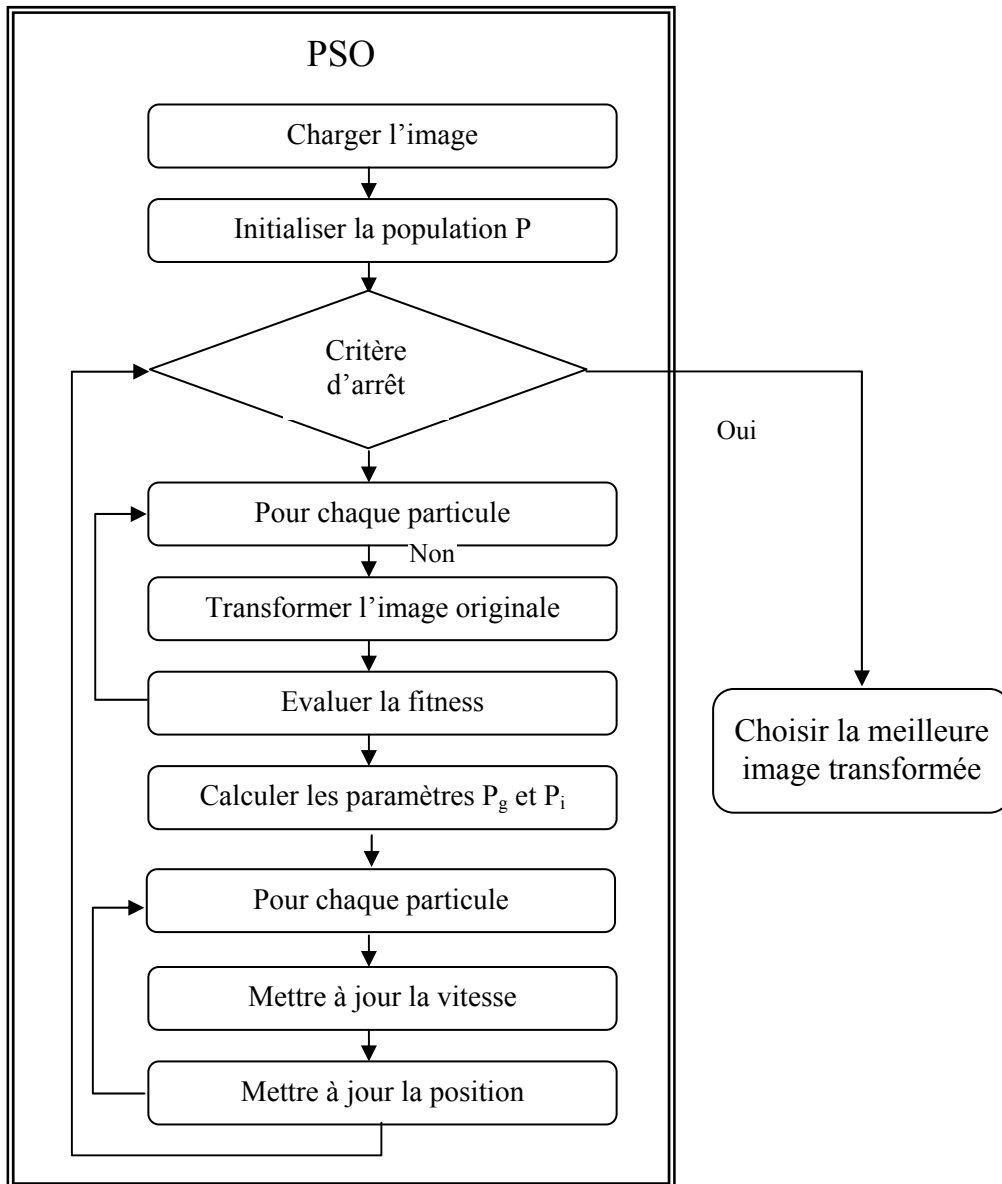


Figure 4.4. Les étapes de l'algorithme PSO

Concernant le voisinage dans l'essaim, différentes définitions de voisinage ont été définies (Section 6.1.4) ; dans notre cas, nous supposons que chaque particule est un voisin à chaque autre particule dans l'essaim (Gbest).



Voici un exemple de l'application de l'algorithme PSO au réhaussement de contraste (figure 4.5)

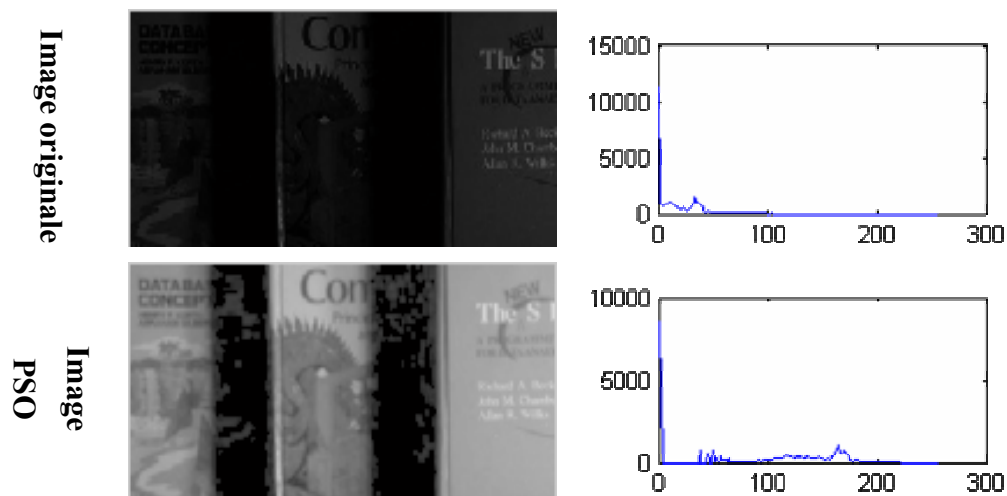


Figure 4.5. Application de PSO pour le réhaussement d'images

3.3 Les étapes de l'algorithme PPO

Chaque particule (proie ou prédateur) a une vitesse et une position qui sont représentées respectivement par les deux vecteurs: X et Y ; avec 4 dimensions. Par conséquent, le but de PPO est de trouver la meilleure combinaison des 4 paramètres, de l'équation (1), pour trouver la meilleure transformation de l'image originale.

Les étapes de l'algorithme PPO pour le réhaussement d'images en niveaux de gris sont:

1. Créer la population initiale avec des positions aléatoires des particules (les proies), ainsi qu'une position aléatoire du prédateur. Chaque particule X est un vecteur de 4-dimensions représentant une combinaison des variables (a , b , c , k).
2. Initialiser la meilleure position personnelle (p_i) de chaque particule à sa position initiale.
3. Pour chaque itération :
 - a- Transformer l'image originale selon les valeurs de : a , b , c et k données par chaque particule i en utilisant l'équation (1)
 - b- évaluer la fitness de chaque particule (basée sur l'image transformée correspondante) utilisant la fonction objective décrite dans l'équation (2)
 - c- trouver la meilleure position globale ' p_g ' dans l'essaim, correspondant à la meilleure image réhaussée,
 - d- mettre à jour la vitesse du prédateur et sa position selon les équations (5).
 - e- mettre à jour la position et la vitesse de chaque particule en utilisant les équations (4)



4. Trouver la meilleure position dans l'essaim, qui donne la meilleure image augmentée et afficher l'image correspondant.

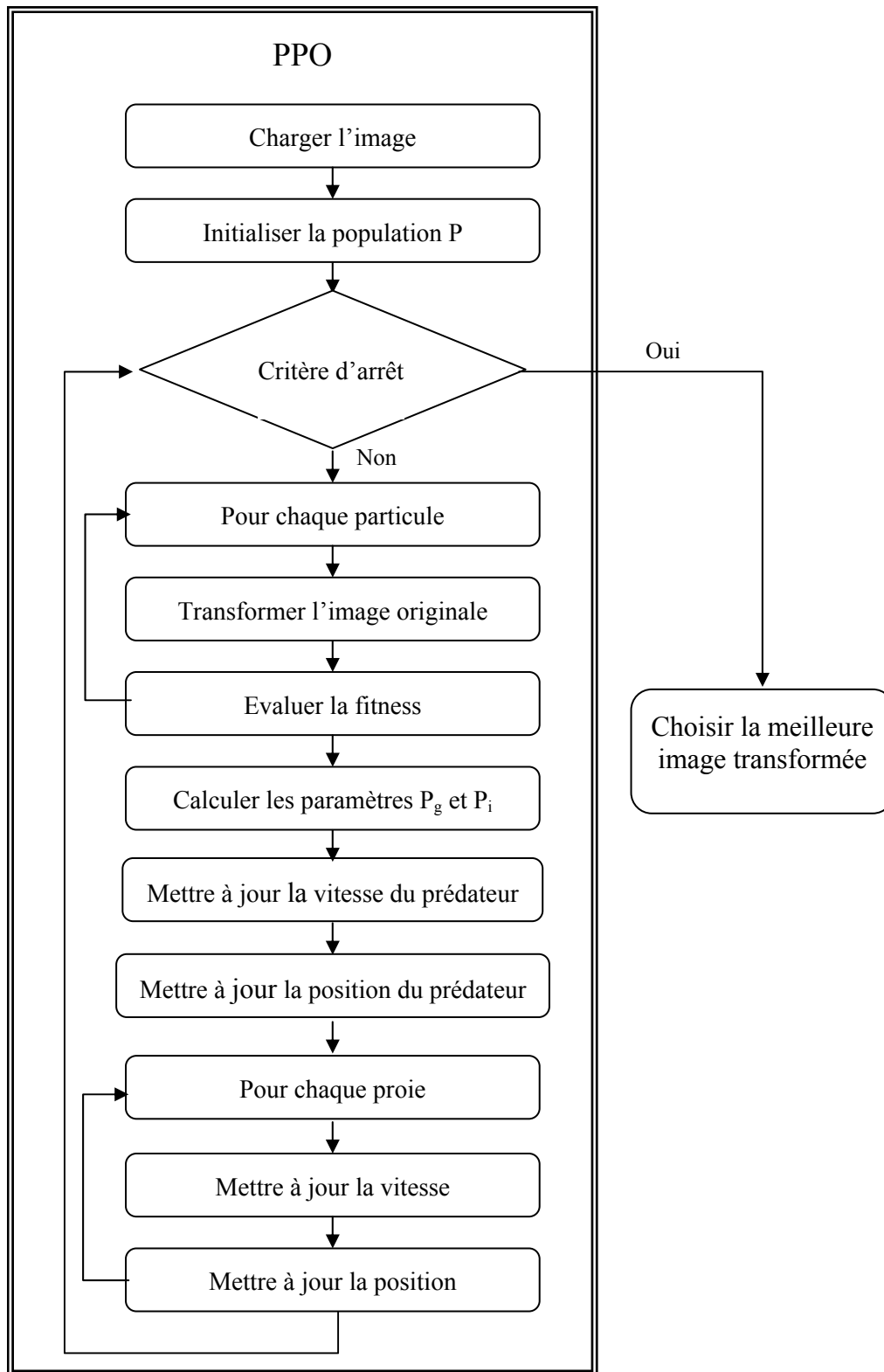


Figure 4.6. Les étapes de l'algorithme PPO



Voici un exemple de l'application de l'algorithme PPO au réhaussement de contraste (figure 4.7)

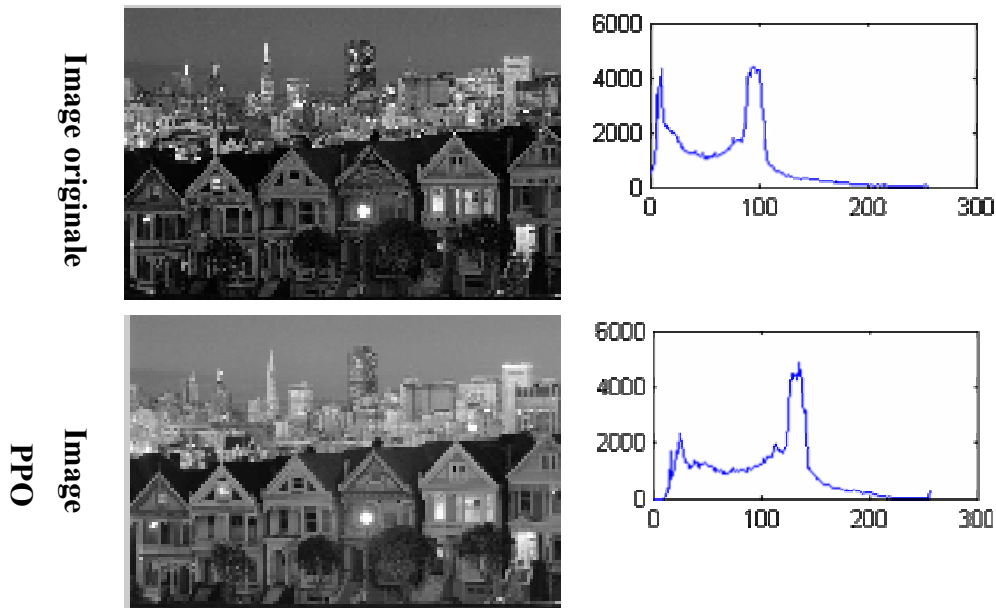


Figure 4.7. Résultat de l'application PPO

4. Résultats expérimentaux et discussion

Les résultats de l'application des deux algorithmes PPO et PSO pour le réhaussement de contraste dans les images en niveau de gris, sont comparés à ceux obtenus par la technique d'égalisation d'histogramme. Pour obtenir un histogramme il faut calculer le nombre de chaque niveau de gris présent dans l'image. La technique d'égalisation d'histogramme est l'une des premières techniques de réhaussement de contraste qui consiste à réarranger les valeurs des niveaux de gris de telle façon à obtenir une distribution uniforme des niveaux de gris dans l'image, (comme il est présenté dans la figure 4.8)

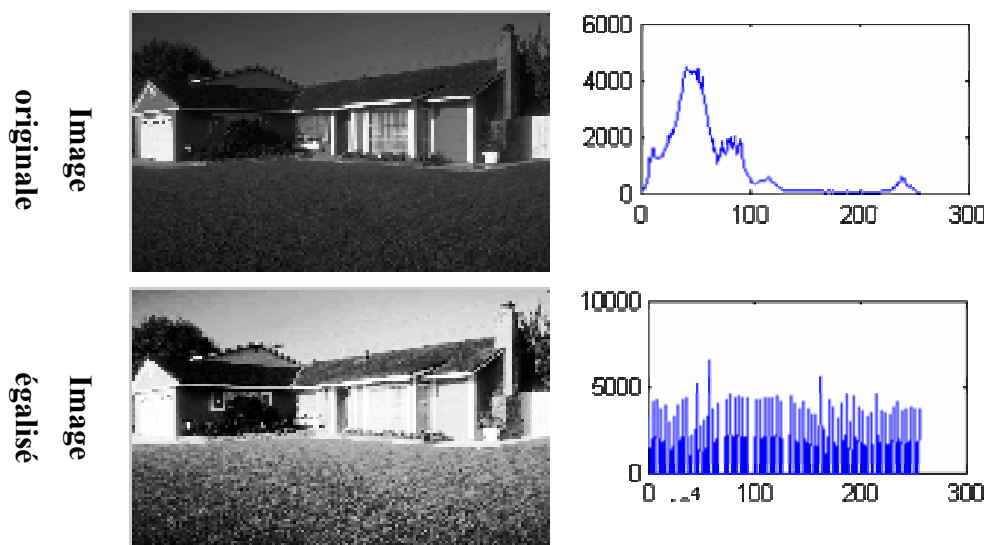


Figure 4.8. La technique d'égalisation d'histogramme.



A partir de l'histogramme on peut obtenir des renseignements rapides sur une image. On peut notamment faire la distinction entre une image trop foncée (les niveaux en majorité sont près de 0) et une image trop claire (les niveaux en majorité sont près de 255) comme il est indiqué dans la figure suivante.

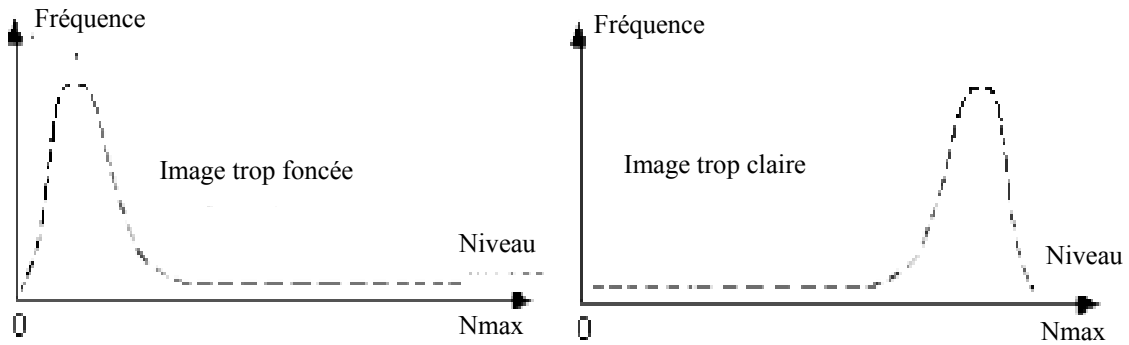
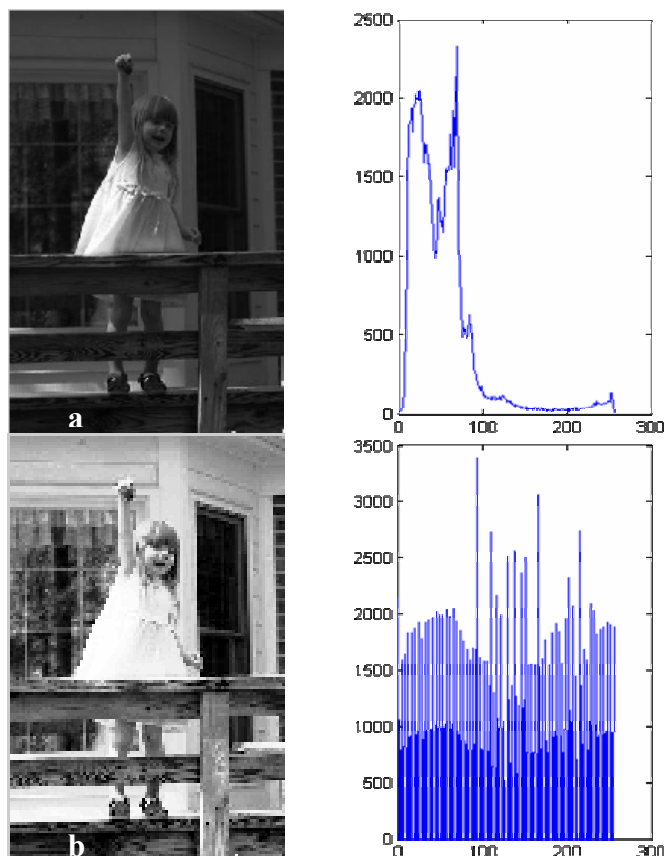


Figure 4.9. Histogramme d'une image trop foncée et celui d'une image trop claire.

L'objectif de notre travail est de trouver la meilleure image transformée, qui est plus claire et plus satisfaisante pour les préférences de la vision humaine. Pour prouver l'efficacité de l'application de des algorithmes d'optimisation : PPO ainsi que PSO à rehausser le contraste des images, nous avons testé son exécution sur un ensemble d'images en niveau de gris, le premier exemple d'image est illustré dans la figure suivante (la figure 4.10):



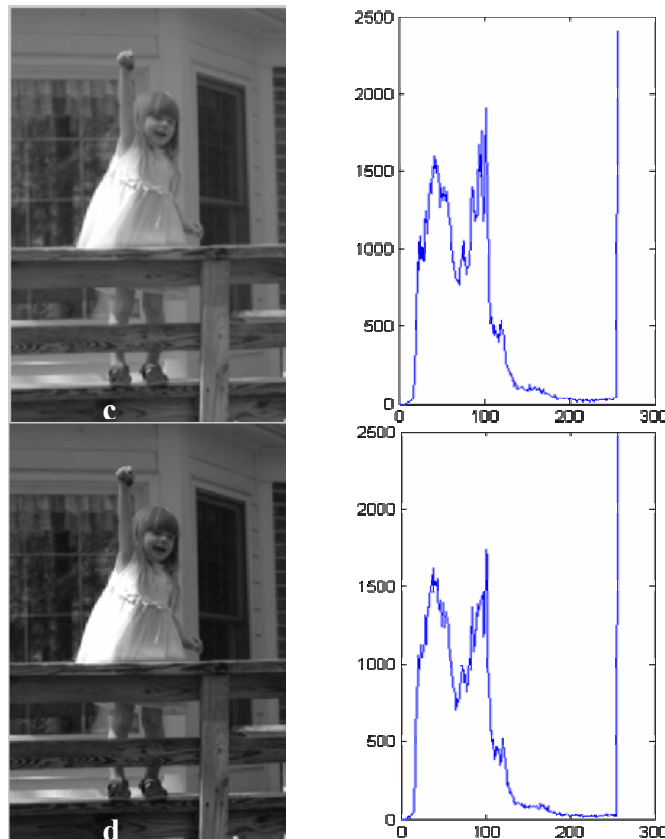


Figure 4.10. Gauche: (a): image originale, (b): image égalisée, (c): image PPO. (d) image PSO. Droite: leurs histogrammes correspondants.

Dans la figure 4.10, on montre une comparaison entre le résultat obtenu en appliquant, sur l'image originale, la méthode classique de réhaussement de contraste, qui est la technique d'égalisation d'histogramme et ceux obtenus par l'application des deux algorithmes d'optimisation PPO et PSO. Dans cette figure l'effet de l'algorithme PPO et celui de PSO sont très clairs. Les algorithmes PPO et PSO augmentent le contraste local de l'image originale mieux que la technique d'égalisation d'histogramme. On peut observer aussi que PPO augmente le contraste de l'image mieux que PSO.

On a examiné également l'application sur un ensemble d'images en niveaux de gris, comme il est illustré dans la figure 4.11, et d'ailleurs démontre l'efficacité de notre méthode par rapport à la méthode d'égalisation d'histogramme. Les caractéristiques de ces images sont données dans le Tableau 6. Ces images sont des images couleurs, mais on a utilisé une fonction Matlab (**Rgb2gray**) qui permet de transformer une image couleur à une image en niveaux de gris.

L'image	Format	Hauteur	Largeur
Image 1	tif	400	129
Image 2	jpg	432	283
Image 3	tif	199	141
Image 4	jpg	432	283
Image 5	tif	688	534
Image 6	jpg	300	196

Tableau 6. Les Caractéristiques des images réhaussées par PPO et PSO

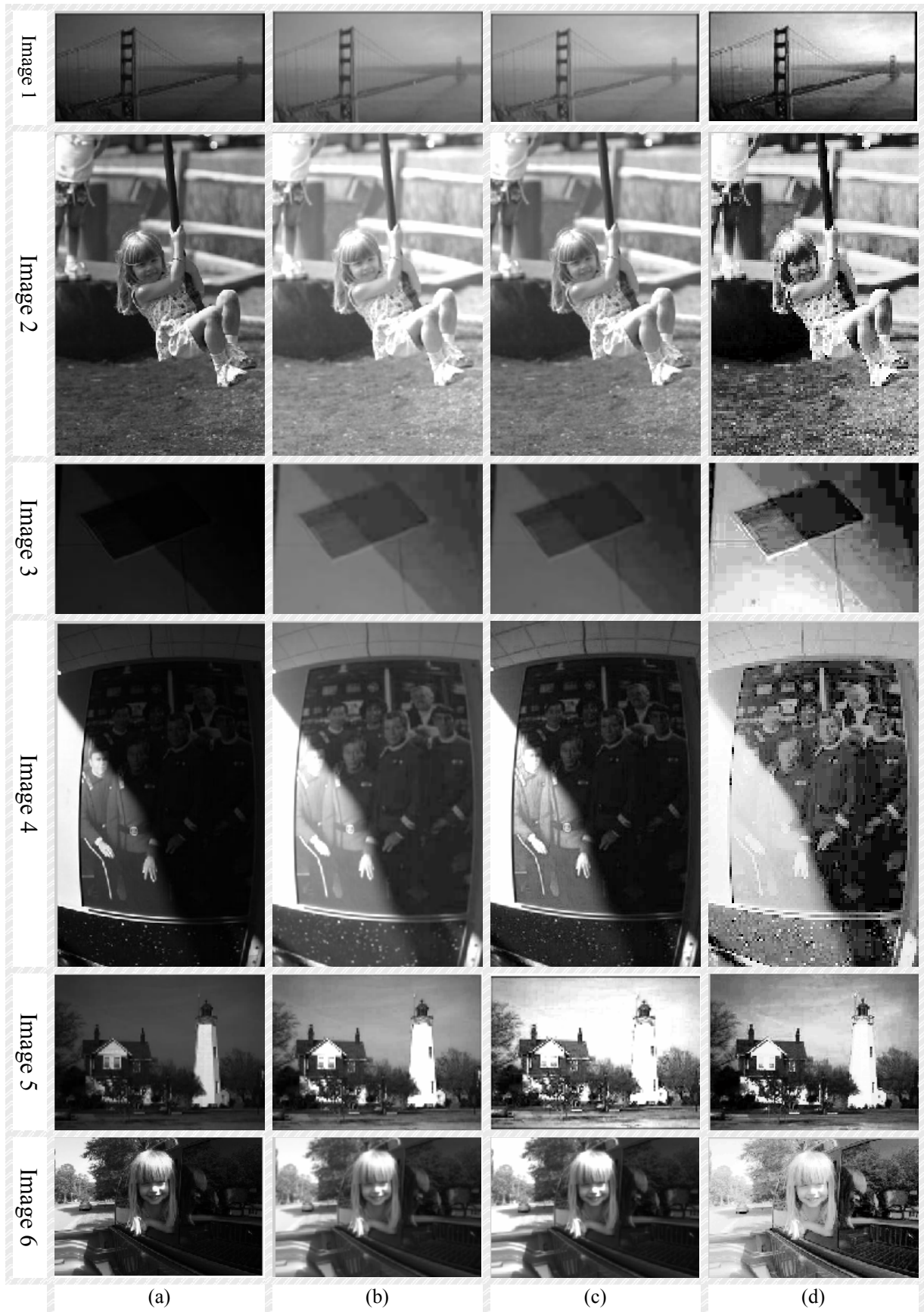


Figure.4.11. (a) Image originale, (b) Image PPO, (c) Image PSO, (d) Image égalisée



On a essayé aussi d'appliquer les deux algorithmes sur des images bruitées, et de comparer ces résultats aussi avec ceux obtenus en appliquant la méthode classique d'égalisation d'histogramme. On a ajouté le bruit aux images en utilisant la fonction de Matlab : `imnoise`, qui demande deux paramètres à spécifier :

- » L'image à laquelle on veut ajouter le bruit
- » La méthode utiliser pour ajouter le bruit

Il existe quatre méthodes pour ajouter le bruit :

- » Gaussien : bruit blanc gaussien avec les constantes : la moyenne M et la variance V , s'ils ne sont pas spécifiés, elles prennent par défaut : 0 et 0.01 respectivement.
- » 'localvar': "Zero-mean Gaussian white noise with an intensity-dependent variance"
- » 'poisson': bruit de Poisson
- » 'salt & pepper' "On and Off" pixels
- » 'speckle' : bruit multiplicatif.

On a choisi la méthode Gaussienne pour ajouter le bruit à une image en niveau de gris.

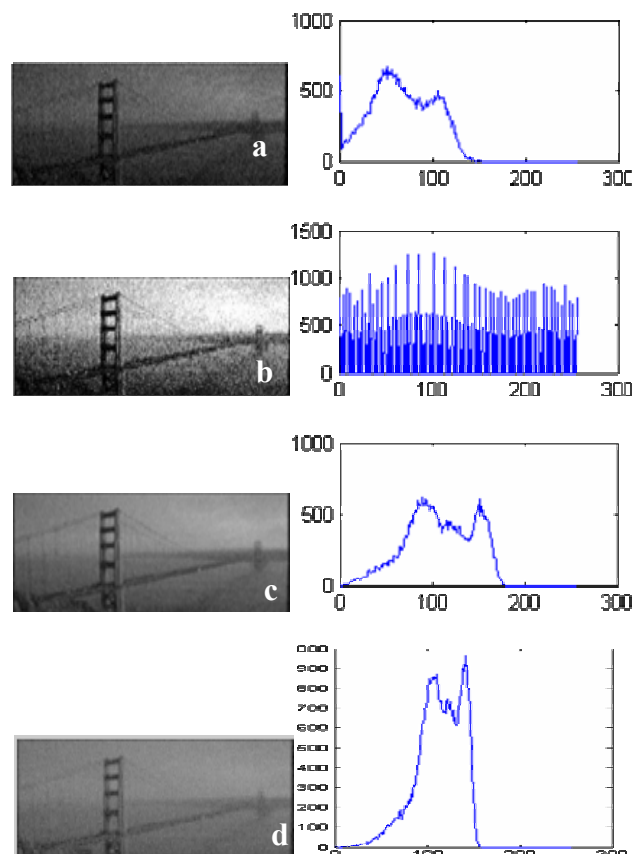


Figure 4.12. (a) Image bruitée, (b) Image égalisée, (c) Image PPO, (d) Image PSO.

Dans la figure ci-dessus, après l'ajout du bruit à l'image originale, les résultats obtenus en appliquant les méthodes PPO et PSO est clairement meilleur que ceux obtenus par la méthode d'égalisation d'histogramme. En plus de l'opération de réhaussement sur cette image qui mène à un bon résultat, PPO et PSO diminuent aussi le taux de bruit dans l'image, et toujours PPO donne de bons résultats par rapport à PSO.

Conclusion Et Perspectives

Le travail présenté dans ce mémoire vise à enrichir les méthodes et les techniques appliquées au réhaussement d'images. Après l'application de plusieurs algorithmes évolutionnaires pour le réhaussement de contraste dans les images par plusieurs auteurs, nous avons essayé d'utiliser un autre genre d'algorithmes évolutionnaires qui est les algorithmes co-évolutionnaires (inspirés des relations de co-évolution entre les individus au sein d'un écosystème). Ces algorithmes sont classés en deux types: algorithmes coopératifs et algorithmes concurrentiels. Dans notre travail, nous avons appliqué l'algorithme PSO (Particle Swarm Optimization) et l'algorithme co-évolutionnaire concurrentiel d'optimisation PPO (Predator Prey Optimization), qui est inspiré des relations prédateur-proie dans un écosystème.

Un ensemble d'images en niveaux de gris a été employé, avec différents formats (JIF, TIF, JPG, ... etc.) pour prouver l'efficacité de notre application pour le réhaussement de contraste de ces images. Comme on a démontré dans le dernier chapitre, les résultats obtenus en appliquant PPO et PSO sont comparés à ceux obtenus par la technique classique de réhaussement de contraste qui est la méthode d'égalisation d'histogramme, et il était clair que les résultats obtenus par PPO et PSO sont meilleurs par rapport à ceux de la technique classique d'égalisation d'histogramme. L'algorithme PPO prouve sa performance par rapport à l'algorithme PSO et augmente le contraste mieux que PSO. Les deux algorithmes PPO et PSO sont aussi appliqués aux images bruitées, et donnent de bons résultats en augmentant le contraste et en réduisant le bruit.

Après l'efficacité démontrée par l'application de PPO par Silva et al. [Silva 2003], par rapport à PSO, sur un ensemble de fonctions multimodales généralement utilisées en tant que problèmes de repère d'optimisation dans le calcul évolutionnaire, notre application de PPO trouve également de meilleures solutions dans le cas du réhaussement de contraste dans les images en niveaux de gris, ce qui encourage la future recherche dans les domaines de la résolution de problèmes complexes d'utiliser cette nouvelle extension (PPO). L'algorithme PPO peut être étendu d'une manière à utiliser plusieurs prédateurs dans l'essaim, pour produire plus de perturbation dans l'essaim de particules.



Bibliographie

- [**Abbassene 2002**] A. Abbassene, S. Meshoul, M. Batouche, «*Segmentation d'Images par Emergence, Basée sur les Automates Cellulaires Évolutionnaires (EvCA)*», Groupe Vision et Infographie, Laboratoire LIRE, Université Mentouri de Constantine.
- [**Allen 1982**] Allen, T. F. H. and T. B. Starr (1982). «*Hierarchy: perspectives for ecological complexity*», University of Chicago Press.
- [**Allen 1992**] Allen, T. F. H. and T. W. Hokstra (1992). «*Toward a unified ecology*», Columbia University Press.
- [**Angeline 1998**] Angeline, P. J. (1998). «*Evolutionary optimisation versus particle swarm optimisation: philosophy and performance differences*», The Seventh Annual Conf. on Evolutionary Programming.
- [**Antony 2004**] Antony W. Iorio and Xiaodong Li (2004). «*A Cooperative Co-évolutive Multiobjective Algorithm Using Non-dominated Sorting*», School of Computer Science and Information Technology, Royal Melbourne Institute of Technology University, Melbourne, Vic. 3001, Australia.
- [**Armetta 2004**] Frédéric Armetta, Salima Hassas, Simone Pimont (2004). «*Une approche auto-organisationnelle pour la gestion dynamique des flux dans les chaînes de production*».
- [**Awcock 1995**] Awcock, G.J. and Thomas, R. (1995). «*Applied Image Processing*». Basingstoke: Macmillan.
- [**Bach 1996**] Bach, T. (1996). «*Evolutionary Algorithms in Theory and Practice: evolution strategies, evolutionary programming, genetic algorithms*». New York Oxford: Oxford University Press.
- [**Baillie 2003**] Jean-Christophe Baillie (2003). «*Segmentation* » ES322-Traitement d'Image et Vision Artificielle.
- [**Ballet 1998**] Ballet P., Rodin V., and J. Tisseau (1998). «*A multiagent system to simulate in-vitro experimentation*», Proceedings of the 1998 IIIE Conference on Systemics, Cybernetics and Informatics (SCI '98), Orlando (USA).
- [**Bichot 2005**] Charles-Edmond Bichot Jean-Marc Alliot (2005). «*Optimisation par colonies de fourmis appliqué au découpage de l'espace aérien européen en zones de qualification*», LOG (Laboratoire d'Optimisation Globale) CENA/ENAC, Toulouse, France.
- [**Bonabeau 1999**] Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999). «*Swarm Intelligence: From Natural to Artificial Systems*», Oxford University Press.

- [**Bonnet 2004**] Noël Bonnet (2004). « *Concepts Fondamentaux de L'image Numerique* », INSERM U514 et LERI, Université de Reims.
- [**Borcard 2005**] Daniel Borcard (2005). « *Les communautés Structure des communautés* », Département de sciences biologiques, Université de Montréal.
- [**Bourquin 2004**] Yvan Bourquin (2004). « *Self-Organization of Locomotion in Modular Robots* », University of Sussex, Brighton, United Kingdom.
- [**Bousquet 2001**] Bousquet F. (2001). « *Modélisation d'accompagnement Simulations multi-agents et gestion des ressources naturelles et renouvelables* », l'Université de Lyon 1.
- [**Bousquet 1998**] Bousquet F., I. Bakam, et al. (1998). « *Cormas : Common-pool Resources and Multi-Agent Systems* ». Lecture Notes in Computer Science 1416, pp. 826-838. Springer.
- [**Bousquet 2002**] François Bousquet, Christophe Le Page, Jean-Pierre Müller (2002). « *Modélisation et simulation multi-agent* ».
- [**Bridge 2003**] Richard Bridge (2003). « *Computer image processing* », Issue V1.R2.M0, Copyright © Tessella.
- [**Brits 2002**] R. Brits, A. P. Engelbrecht, and F. van den Bergh (2002). « *A niching particle swarm optimizer* », Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, pp. 692-696.
- [**Bucci 2002**] A. Bucci and J.B. Pollack (2002). « *A Mathematical Framework for the Study of Co-évolution* », Foundations of Genetic Algorithms VII (FOGA'2002).
- [**Burel 2001**] Gilles Burel (2001): « *Introduction au traitement d'images* », simulation sous matlab.
- [**Carlisle 2000**] A. Carlisle and G. Dozier (2000). « *Adapting particle swarm optimization to dynamic environments* ». In International Conference on Artificial Intelligence; 1:429--434. 24 J.F. SCHUTTE ET AL.
- [**Catalao 2002**] Vera Lessa Catalao (2002). « *Alphabétisation écologique* ».
- [**Chauvet 2002**] Erica Chauvet et al. (2002). « *A Lotka-Volterra Three-species Food Chain* », VOL. 75, NO. 4.
- [**Clerc 2002**] Clerc, M. and Kennedy, J. (2002). « *The particle swarm-explosion, stability, and convergence in a multidimensional complex space* ». IEEE Transactions on Evolutionary Computation, Vol. 6, No. 1, pp. 58-73.
- [**Cochard 1998**] Gérard-Michel Cochard (1998). « *Notions de traitement et d'analyse d'image* », Université de Picardie Jules Verne.
- [**Coello 2003**] Coello, C. A. C., Sierra, M. R. (2003). « *A Co-évolutive Multi-objective Evolutionary Algorithm* ». In: Sarker, R., Reynolds, R., Abbass, H., Tan, K. C., McKay, B., Essam, D., Gedeon, T. (eds.): Proc. 2003 Congress on Evolutionary Computation (CEC'03). IEEE Press, Piscataway NJ (2003) 482-489.
- [**Davi 2002**] H.Davi et A.Deredec (2002). « *Introduction à la modélisation en biologie des populations et des écosystèmes* », UV modélisation en écologie
- [**David 1999**] David S. Boukal, Vlastimil Křivan (1999). « *Lyapunov functions for Lotka+Volterra predator+prey models with optimal foraging behaviour* », J. Math. Biol. 39: 493-517

- [**Dawkins 1986**] Richard Dawkins (1986). «*The blind Watchmaker*». Harlow : Longman Scientific and Technical.
- [**De Castro 2003**] De Castro, L. and Timmis, J. (2003). «*Artificial immune systems as a novel soft computing paradigm*», *Soft Computing Journal* 7(7).
- [**De Meyer 2004**] Kris De Meyer (2004). «*Foundations Of Stochastic Diffusion Search*».
- [**DeAngelis 1979**] DeAngelis D. L., D. C. Cox, et al. (1979). «*Cannibalism and size dispersal in the young-of-the year largemouth bass: experiments and model*». *Ecologica Modelling*, 8: 133-148.
- [**Déléage 1978**] Déléage J.P. (1978). «*Une histoire de l'écologie* ». Edition la découverte.
- [**Deneubourg 1989**] Deneubourg J. L. and S. Goss (1989). «*Collective patterns and decision making*». *Ethol. Ecol. & Evol*, 1: 295-311.
- [**Desachy 2001**] Jacky Desachy (2001). «*Analyse D'images* ». Université des Antilles et de la Guyane.
- [**Drogoul 1996**] Drogoul A. (1996). «*De la simulation multi-agent à la résolution collective de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multi-agent*». Thèse de l'Université Paris 6.
- [**Duboz 2004**] Raphaël Duboz (2004). «*Intégration de modèles hétérogènes pour la modélisation et la simulation de systèmes complexes, Application à la modélisation multi-échelles en écologie marine* », Laboratoire d'Informatique du Littoral - UPRES JE-2335.
- [**Duch 1998**] Włodzisław Duch, Jerzy Korczak (1998). «*Optimization and global minimization methods suitable for neural networks*». *Neural Computing Surveys* 2, XXX-YYY.
- [**Durand 2004**] Nicolas Durand (2004). «*Algorithmes génétiques et autres outils d'optimisation appliqués à la gestion de trafic aérien* ».
- [**Eberhart 1998**] Eberhart, R. and Shi, Y. (1998). «*Comparison between Genetic Algorithms and Particle Swarm Optimization*». The 7th Annual Conference on Evolutionary Programming, San Diego, USA.
- [**Egli 2002**] Simon Egli et Ivano Brunner (2002). «*Les mycorhizes, Une fascinante biocénose en forêt* ». Institut fédéral de recherches WSL, CH-8903 Birmensdorf, © WSL Birmensdorf,
- [**Elton 1942**] C. Elton and M. Nicholson (1942). «*The ten-year cycle in numbers of the lynx in Canada*». *Journal of Animal Ecology*, 11 (1942), 215–244.
- [**Emmeche 1994**] Emmeche, C. (1994). *The garden in the Machine: the emerging science of artificial life*. Translated by Steven Sampson. Princeton, N.J.: Princeton university press.
- [**Eriksson 1997**] R. Eriksson and B. Olsson. (1997). «*Cooperative coevolution in inventory control optimisation*». In G. D. Smith, N. C. Steele, and R. F. Albrecht, editors, In *Proceedings of 3rd International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA 1997)*, Norwich, UK,. Springer.
- [**Ferber 1994**] Jacques Ferber (1994). «*Coopération réactive et émergence* », *Intellectica*, pp. 19-52, Université Paris 6.

- [Fogel 1966] Fogel, L.J., Owens, A.J., and Walsh, M.J. (1966). «*Artificial intelligence through simulated evolution*». New York : Wiley.
- [Fogel 2000] Fogel, D.B. (2000). «*Evolutionary Computation: toward a new philosophy of machine intelligence*». New York: IEEE Press, 2nd ed.
- [Fonseca 1993] Fonseca, C. M., Fleming, P. J. (1993). «*Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion, and Generalization*». In: Forrest S. (ed.): Proc. of the Fifth International Conference on Genetic Algorithms, Vol. 26. Morgan Kaufmann, Los Altos CA 30-45.
- [Francisco 2002] Dominique Francisco (2002). «*Algorithmes Evolutionnaires Et Optimisation Multi-Objective En Data Mining*», I3S/RR-2002-12-FR.
- [Francour 2005] P. Francour (2005). «*Les relations Proies-Prédateurs*».
- [Franken 2004] Nelis Franken, Andries P Engelbrecht (2004). «*PSO approaches to co-evolve IPD strategies*», University of Pretoria, South Africa,
- [Frontier 1995] S. Frontier et D. Pinchod-Viale (1995). «*Ecosystèmes. Structure-fonctionnement évolution* », Masson.
- [Gagnon 1999] Langis Gagnon et Marc Lalonde (1999). «*Vision artificielle et traitement d'image* » Centre de recherche informatique de Montréal ©.
- [Géorgé 2003] J.P.Géorgé (2003). «*L'Emergence*». Rapport interne IRIT/2003-12-R.
- [Goldberg 1994] David E.Goldberg (1994). «*Algorithmes génétiques: Exploration, optimisation et apprentissage automatique*», Edition Addison-Wesley France, SA.
- [Goldberg 2004] Goldberg D.E. (1989). «*Genetic Algorithms in Search, Optimization and Machine Learning*». Addison Wesley, Massachusetts.
- [Gonzales 1987] Gonzales R. and Winter P (1987). «*Digital Image Processing*», Addison Wesley.
- [Goreaud 2004] Goreaud F. (2004). «Modèle exponentiel, loi logistique, modèle proie-prédateur : des outils mathématiques et des concepts d'écologie théorique pour comprendre la dynamique des populations». Modélisation des systèmes biologiques, Fiche technique, 10 pages.
- [Grasemann 2005] Uli Grasemann, Risto Miikkulainen, (2005). «*Effective Image Compression using Evolved Wavelets*», GECCO'05, Washington, DC, USA.
- [Grimm 1999] Grimm V., T. Wyszomirski, et al. (1999). «*Individual-based modelling and ecological theory: synthesis of a workshop*». Ecological modelling, 115: 275-282.
- [Gutknecht 1997] Gutknecht O. and J. Ferber(1997). «*MadKit: Organizing heterogeneity with groups in a platform for multiple multi-agent systems*». Rapport Interne LIRMM.
- [Hawick 2004] K. A. Hawick, C. J. Scogings and H. A. James (2004). «*Defensive Spiral Emergence in a Predator-Prey Model* », Conference on Complex Systems, Proceedings of the 7th Asia-Pacific.
- [Haynes 1995] Thomas Haynes et al. (1995). «*Evolving multiagent coordination strategies with genetic programming*». Technical Report UTULSA-MCS-95-04, The University of Tulsa.

- [**Hillis 1990**] Hillis, D. (1990). «*Coevolving Parasites Improves Simulated Evolution as an Optimization Procedure*». In: Langton, C. G., Taylor, C., Farmer, J. D., Rasmussen, S. (eds.): *Artificial Life II - Proc. of the Workshop on the Synthesis and Simulation of Living Systems*. Addison Wesley, Redwood City CA 313-324.
- [**Holland 1975**] Holland J.H. (1975). «*Adaptation in Natural and Artificial Systems*». Ann Arbor : The University of Michigan Press.
- [**Holling 1959**] Holling, C.S. (1959). «*The components of predation as revealed by a study of small mammal predation of the European pine sawfly*». *Canad. Entomol.* 91: 293-320.
- [**Holling 1987**] Holling. (1987). «*Simplifying the complex: the paradigms of ecological function and structure*». *European Journal of Operational Research* 30: 139-146.
- [**Hutzler 2000**] Guillaume Hutzler (2000). «*Du Jardin des Hasards aux Jardins de Données: une approche artistique et multi-agent des interfaces homme / systèmes complexes*», Université Paris Vi - Pierre Et Marie Curie.
- [**Jain 1991**] Jain A. K. (1991). «*Fundamentals of Digital Image Processing*», Prentice Hall.
- [**Janson 2003**] S. Janson and M. Middendorf, (2003) «*A hierarchical particle swarm optimizer*» *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 770-776.
- [**John 1999**] John P. Cartlidge (1999). «*An Analysis of Evolutionary Computation used in Image Processing Techniques*» BSc Artificial Intelligence and Mathematics.
- [**Juille 1990**] H. Juille and J.B. Pollack (1990). «*Co-évolutive Learning: a Case Study*», in *Proceedings of the 15th International Conference on Machine Learning*.
- [**Kaewkamnerdpon 2005**] Boonserm Kaewkamnerdpong and Peter J. Bentley (2005). «*Perceptive Particle Swarm Optimisation: An Investigation*», University College London, UK.
- [**Keerativuttitumrong 2002**] Keerativuttitumrong, N., Chaiyaratana, N., Varavithya V. (2002): «*Multi-objective Cooperative Co-évolution ary Genetic Algorithm*». In: Merelo Guervs, J.J., Adamidis, P., Beyer, H.-G., Fernandez-Villacaas, J.-L., Schwefel, H.-P. (eds.): *Proc. of Parallel Problem Solving From Nature VII (PPSN'02)*. Lecture Notes in Computer Science, Vol 2439. Springer-Verlag, Berlin Germany 288-297.
- [**Kennedy 1995**] Kennedy, J. and Eberhart, R. (1995). «*Particle Swarm Optimization*», *IEEE International Conference on Neural Networks (Perth, Australia)*, IEEE Service Center, Piscataway.
- [**Kennedy 1997**] J. Kennedy and R. C. Eberhart (1997). «*A discrete binary version of the particle swarm algorithm*». *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, pp. 4104-4109.
- [**Kennedy 2001**] Kennedy, J., Eberhart, R. and Shi, Y. (2001). «*Swarm Intelligence*», Morgan Kaufmann.
- [**Kennedy 2002**] J. Kennedy and R. Mendes (2002). «*Population structure and particle swarm performance*». In *Proceedings of Congress on Evolutionary Computing (CEC 2002)*, Honolulu, Hawaii.
- [**Kirkpatrick 1983**] S. Kirkpatrick, C.D. Gellat Jr. and M.P. Vecchi (1983). «*Optimization by simulated annealing*», *Science*, 220: 671-680.

- [Kleber 1997] Kleber X. S. de Souza, Mario A. Nascimento (1997). «*Using JAVA to implement agents in artificial ecosystems*», First European Conference for Information Technology in Agriculture, Copenhagen,
- [Koza 1999] John R. Koza, et al. (1999). «*Genetic Programming 3: Darwinian Invention and Problem Solving*. Morgan Kaufman»
- [Krempp 2005] Yannick Krempp (2005). «*Manipulation d'images*», Cellular Imaging Facility.
- [Krohling 2002] R. A. Krohling, L. S. Coelho, and Y. Shi (2002). «*Cooperative particle swarm optimization for robust control system design*» Proceedings of the 7th Online World Conference on Soft Computing in Industrial Applications.
- [Krohling 2004] Renato A. Krohling, Frank Hoffmann (2004). «*Co-evolutionary Particle Swarm Optimization for Min-Max Problems using Gaussian Distribution*».
- [Langton 1997] Langton, C.G., ed. (1997). «*Artificial Life: an overview*». Cambridge, Mass. London: MIT.
- [Le Moigne 1989] Le Moigne, J. L. (1989). «*La modélisation des systèmes complexes*», Dunod.
- [Li 2002] Xiaodong Li and Stuart Sutherland. (2002). «*A Cellular Genetic Algorithm Simulating Predator-Prey Interactions*», School of Computer Science and Information Technology, Australia.
- [Lina 2004] J.M. Lina (2004). «*Traitement des Images* », LOG 620-Analyse D'algorithmes.
- [Liu 1997] Liu, J., Tang, Y.Y., and Cao, T.C. (1997), «*An Evolutionary Autonomous Agents Approach to Image Feature Extraction*». IEEE Transactions on Evolutionary Computation, Vol. 1, No. 2, pages 141-158.
- [Løvbjerg 2002] Morten Løvbjerg (2002). «*Improving Particle Swarm Optimization by hybridization of stochastic search heuristics and Self-Organized Criticality*», University of Aarhus,
- [Lutton 2004] Evelyne Lutton (2004). «*Darwinisme artificiel : une vue d'ensemble*», INRIA - Rocquencourt - Equipe COMPLEX - Projet FRACTALES,
- [Maniezzo 1996] V. Maniezzo M. Dorigo and A. Colomi (1996). «*The ant system: Optimization by a colony of cooperating agents*». IEEE Transactions on Systems, Man, and Cybernetics-Part B,, 26(1) :29–41.
- [Mao 2001] Mao, J., Hirasawa, K., Hu, J., Murata, J. (2001). «*Genetic Symbiosis Algorithm for Multiobjective Optimization Problems*». In: Beyer, H., Cantu-Paz, E., Goldberg, D., Parmee, I., Spector, L., Whitley, D. (eds.): Proc. 2001 Genetic and Evolutionary Computation Congress (GECCO'01). Morgan Kaufmann,
- [Marcenac 1997] Marcenac P. (1997). «*Modélisation de systèmes complexes par agents*», TSI, 16(8): 1013-1038.
- [Michelle 2003] Michelle Galea (2003). «*Fuzzy Rules from ANT-Inspired Computation*», PhD Proposal School of Informatics; University of Edinburgh.
- [Millonas 1994] Millonas, M. M. (1994). «*Swarm, phase transitions, and collective intelligence*», In C. C. Langton, Ed. Artificial Life III. Addison Wesley, Reading, MA.

- [**Minar 1996**] Minar N., R. Burkhart, et al (1996). «*The swarm simulation system: a toolkit for building multi-agent simulations*», <http://www.santafe.edu/project/swarm>.
- [**Mitchell 2002**] Melanie Mitchell and Mark Newman (2002). «*Complex Systems Theory and Evolution*», In Encyclopedia of Evolution (M. Pagel, editor), New York: Oxford University Press.
- [**Mitra 1998**] Mitra, S.K., Murthy, C.A., and Kundu, M.K. (1998). «*Techniques for Fractal Image Compression Using Genetic Algorithm*». IEEE Transactions on Image Processing, Vol. 7, No. 4, April 1998, pages 586-593.
- [**Monson 2004**] C. K. Monson and K. D. Seppi (2004). «*The Kalman Swarm: A new approach to particle motion in swarm optimization*». Proceedings of the Genetic and Evolutionary Computation Conference, pp. 140-150.
- [**Moriarty 1997**] D. E. Moriarty (1997). «*Symbiotic Evolution of Neural Networks in Sequential Decision Tasks*». PhD thesis, University of Texas, Austin.
- [**Munteanu 2000**] Munteanu C. Rosa A. (2000) «*Towards Automatic Image Enhancement Using Genetic Algorithms*», Universidade Técnica de Lisboa, Portugal.
- [**Olson 1995**] Olson, R. L. & R. A. Sequeira (1995). «*Emergent Computation and the Modeling and Management of Ecological Systems*». Computers and Electronics in Agriculture. 12:183-209.
- [**Panait 2003**] Liviu Panait, Sean Luke (2003). «*Cooperative Multi-Agent Learning: The State of the Art*», Technical Report GMU-CS-TR-2003-1.
- [**Papacostantis 2005**] Evangelos Papacostantis, Andries P. Engelbrecht, Nelis Franken (2005). «*Coevolving Probabilistic Game Playing Agents Using Particle Swarm Optimization Algorithms* », University of Pretoria.
- [**Paredis 1995**] J. Paredis (1995). «*Coevolutionary computation. Artificial Life*», 2(4):355-375.
- [**Paredis 1997**] J. Paredis (1997). «*Coevolutionary process control*». In G. D. Smith, editor, Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA97), Vienna, Springer.
- [**Pearson 1991**] Pearson, D. (1991). «*Image Processing*». London: McGraw-Hill.
- [**Peña-Reyes 2001**] C.-A. Peña-Reyes and M. Sipper (2001). «*Fuzzy CoCo: A cooperative coevolutionary approach to fuzzy modeling*». IEEE Transactions on Fuzzy Systems, 9(5):727-737.
- [**Potter 1995**] Potter, M. A., De Jong K. (1995). «*A Cooperative Co-évolutive Approach to Function Optimization*». In: Proc. of Parallel Problem Solving From Nature III (PPSN III), Springer-Verlag, Berlin Germany 249-257.
- [**Potter 1997**] M. Potter (1997). «*The Design and Analysis of a Computational Model of Cooperative Co-évolutive*», PhD thesis, George Mason University, Fairfax, Virginia.
- [**Potter 2000**] M.A. Potter and K.A. DeJong (2000). «*Cooperative Co-evolution: an Architecture for Evolving Coadapted Subcomponents*», Evolutionary Computation, vol. 8, pp. 1-29.
- [**Qureshi 2001**] Mohammad Adil Qureshi (2001). «*The Evolution of Agents*», University of London.

- [**Rapp 1996**] Carl Steven Rapp (1996). «*Image Processing and Image Enhancement*», East Tennessee State University
- [**Ray 1991**] Ray, T.S. (1991). «*An Approach to the Synthesis of Life*». In Langton et al. (1991), pages 371-408.
- [**Reynolds 2003**] R. G. Reynolds, B. Peng, and J. Brewster (2003). «*Cultural swarms II: Virtual algorithm emergence*». Proceedings of IEEE Congress on Evolutionary Computation, pp. 1972-1979,
- [**Roberts 2004**] Roberts, M.E., Claridge, E. (2004). «*Cooperative coevolution of image feature construction and object detection*». In et. al., X.Y., eds.: *Parallel Problem Solving from Nature - PPSN VIII*. Volume 3242 of LNCS., Birmingham, UK, Springer 899–908
- [**Roberts 2005**] Roberts, M.E., Claridge, E. (2005): «*A Multistage Approach to Cooperatively Coevolving Feature Construction and Object Detection*».
- [**Rosin 1997 a**] C.D. Rosin and R.K. Belew (1997). «*New Methods for Competitive Co-évolution*», *Evolutionary Computation*, vol. 5, pp. 1-29.
- [**Rosin 1997 b**] Rosin P. L. (1997). «*Edges: saliency measures and automatic thresholding*», *Machine Vision and Applications*, Springer Verlag, vol 9, pp. 139 – 159.
- [**Schmitt 2004**] Karlheinz Schmitt, Jörn Mehnen and Thomas Michelitsch (2004). «*A Predator-Prey Approach for Pareto-Optimization*», No. CI-184/04; Technical Report ISSN 1433-3325, UNIVERSITY OF DORTMUND.
- [**Schoeman 2005**] I. L. Schoeman, A. P. Engelbrecht (2005). «*A Parallel Vector-Based Particle Swarm Optimizer*», University of Pretoria, South Africa,
- [**Schoenauer 2003**] Marc Schoenauer (2003). «*Les algorithmes évolutionnaires : état de l'art et enjeux* », *Projet Fractales*, INRIA Roquencourt (France), InRIA, pp.113,118.
- [**Schwefel 1984**] Schwefel H.-P., *Evolution Strategies* (1984). «*A Family of Non-Linear Optimization Techniques Based on Imitating Some Principles of Organic Evolution*», *Annals of Operations Research*, vol. 1, pp. 165-167.
- [**Setiono 2000**] R. Setiono, (2000). «*Generating Concise and Accurate Classification Rules for Breast Cancer Diagnosis*», *Artificial Intelligence in Medicine*, vol. 18, pp. 205-219.
- [**Shi 1998**] Shi, Y. and Eberhart, R. C. (1998). “Parameter selection in particle swarm optimisation” *Evolutionary Programming VII: Proc. EP 98*. New York, pp. 591-600.
- [**Shi 1999**] Y. Shi and R. C. Eberhart, (1999). «*Empirical Study of Particle Swarm Optimization*» *Proceedings of the 1999 Congress of Evolutionary Computation*, vol. 3, pp. 1945–1950, IEEE Press.
- [**Silva 2003**] Arlindo Silva, Ana Neves, Ernesto Costa, (2003). «*An Empirical Comparison of Particle Swarm and Predator Prey Optimisation*».
- [**Skarbek 1994**] Wladyslaw Skarbek and Andreas Koschan. (1994). «*Colour image segmentation ñ a survey ñ*. Technical report, Institute for Technical Informatics, Technical University of Berlin, October.
- [**Sonka 1993**] Sonka, M., Hlavac, V. and Boyle, R. (1993). «*Image Processing, Analysis, and Machine Vision*». London: Chapman and hall.

- [Spalanzani 1999] Anne Spalanzani (1999). « *Algorithmes évolutionnaires pour l'étude de la robustesse des systèmes de reconnaissance automatique de la parole* » ; l'Université Joseph Fourier - Grenoble I.
- [Tabisa 1998] Tabisa Mohammed (1998). « *Filtre LP : Etude des propriétés et Application en Traitement d'image* », l'université de SAVOIE,
- [Taillard 2001] Taillard, E., Gambardella, L., Gendreau, M. and Potvin, J. (2001). « *Adaptive memory programming: a unified view of metaheuristics* », European Journal of Operational Research 135, 1–16.
- [Théraulaz 1994] Théraulaz G. (1994). « *Du super organisme à l'intelligence en essaim : modèles et représentations du fonctionnement des sociétés d'insectes* ». Dans E. Bonabeau and G. Théraulaz, éditeurs, *Intelligence collective*, pp. 29-109. Hermès.
- [Veenman 2003] C.J. Veenman, M.J.T. Reinders, and E. Backer (2003). « *A Cellular Coevolutionary Algorithm for Image Segmentation* », IEEE Transactions on Image Processing, Vol. 12, No. 3, pp. 304-316.
- [Volterra 1931] Volterra, V. (1931). « *Leçons sur la théorie mathématique de la lutte pour la vie* », Gauthier-Villar, Paris.
- [Watson 1999] Watson, R. A. and Pollack, J. B. (1999). « *How Symbiosis Can Guide Evolution* ». Proceedings of the 5th European Conference on Advances in Artificial Life (ECAL 1999) pp. 29-38.
- [Watson 2001] R. Watson and J. Pollack (2001). « *Co-évolution ary Dynamics in a Minimal Substrate* », in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), L.Spector, E. Goodman, A. Wu, W.B. Langdon, H.-. Voigt, M. Gen, S. Sen, M. Dorigo, S.Pezeshk, M. Garzon and E. Burke Eds. Morgan Kaufmann.
- [Young 1998] Ian T. Young, Jan J. Gerbrands, Lucas J. van Vliet (1998). « *Fundamentals of Image Processing* ».
- [Zaritsky 2004] Assaf Zaritsky, Moshe Sipper (2004). « *Coevolving Solutions to the Shortest Common Superstring Problem* », Department of Computer Science, Ben-Gurion University, P.O. Box 653, Beer-Sheva 84105, Israel.
- [Zhou 1994] Zhou, L., and Franklin, S. (1994). « *Character Recognition Agents* ». In Brooks and Maes, pages 301-306.

Liste des liens

- [Web 1] http://www.geocities.com/boss_be_99/notion_ecolo.htm
- [Web 2] « Glossaire D'écologie », (Révision BTS GPN)
http://www.protection-nature.org/glossaire_ecologie.htm
- [Web 3] <http://www.cvm.qc.ca/yderepen/écologie.doc>
- [Web 4] Claude Villeneuve : « Ecologie : concepts clés » , Professeur responsable du DESS en Eco-Conseil, Département des science fondamentale, Université du Québec à Chicoutimi.
<http://dsf.uqac.ca/eco-conseil/formation/Documents/1ecc801/notiecol.pdf>
- [Web 5] Antoine BLAIS ; « Apports de la biologie en informatique », 2003
http://www.supinfo-projects.com/2003/biologie_informatique/3/

- [Web 6] http://www.geocities.com/boss_be_99/dynamique_populat.htm
- [Web 7] Shlomiya Bar-Yam, « Co-evolution »
http://necsi.org/projects/evolution/co-evolution/co-evolution_intro.html
- [Web 8] Shlomiya Bar-Yam, « Predator-Prey Relationships »
http://necsi.org/projects/evolution/co-evolution/pred-prey/co-evolution_predator.html
- [Web 9] Shlomiya Bar-Yam, « Symbiotic Relationships »
http://necsi.org/projects/evolution/co-evolution/symbiosis/co-evolution_symbiosis.html
- [Web 10] Nils FERRAND, « De l'apport potentiel de la sociologie pour l'ingénierie des systèmes sociaux artificiels », Laboratoire LEIBNIZ – IMAG,
<http://www.lisc.clermont.cemagref.fr/Labo/MembresEtPagesIntermediaires/pagesperso/Anciensmembres/ferrand/publications/roche-final.html#Heading1>
- [Web 11] «Functional and Numerical Response»
<http://www.ento.vt.edu/~sharov/PopEcol/lec10/funcresp.html>
- [Web 12] Antoine BLAIS ; « Apports de la biologie en informatique », 2003
http://www.supinfo-projects.com/2003/biologie_informatique/2/
- [Web 13] « Imagerie Numérique », Faculté de médecine de Brest; Option d'Informatique, 2004-2005 ;
http://cours.univ-brest.fr/UFR-Medecine/Pcem2/Info//C8_Images_Numeriques.pdf
- [Web 14] Armel Ménez, « Initiation au traitement d'images avec le Gimp (version 2) » Un document réalisé pour le Centre de ressources du Pays de Brest ;
<http://www.archipeldulibre.org/spip/IMG/jegimpeTugimpes-1.0.pdf>
- [Web 15] L.Delahoche – E.Brassart : « Introduction au traitement d'images , Analyse fréquentielle, Filtrage » ;
<http://dionysos.univ-lyon2.fr/~dsarrut/bib/vision/intro-image.pdf>

