



**HAL**  
open science

# Etude et Exploitation des Réseaux de Neutralité dans les Paysages Adaptatifs pour l'Optimisation Difficile

Sébastien Verel

► **To cite this version:**

Sébastien Verel. Etude et Exploitation des Réseaux de Neutralité dans les Paysages Adaptatifs pour l'Optimisation Difficile. Autre [cs.OH]. Université Nice Sophia Antipolis, 2005. Français. NNT : . tel-00159727

**HAL Id: tel-00159727**

**<https://theses.hal.science/tel-00159727>**

Submitted on 3 Jul 2007

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

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

# Étude et Exploitation des Réseaux de Neutralité dans les Paysages Adaptatifs pour l'Optimisation Difficile

## THÈSE

présentée et soutenue publiquement le 12 décembre 2005

pour obtenir le titre de

**Docteur ès Sciences**  
**de l'Université de Nice-Sophia Antipolis**

**spécialité informatique**

par

Sébastien VEREL

### Composition du jury

<i>Président :</i>	M. Enrico FORMENTI	Professeur à l'Université de Nice-Sophia Antipolis
<i>Rapporteurs :</i>	M. Cyril FONLUPT	Professeur à l'Université du Littoral - Côte d'Opale
	M. El-Ghazali TALBI	Professeur à l'Université de Lille
	M. Marco TOMASSINI	Professeur à l'Université de Lausanne
<i>Examineur :</i>	M. Manuel CLERGUE	Maître de conférences à l'Université de Nice-Sophia Antipolis
<i>Directeur :</i>	M. Philippe COLLARD	Professeur à l'Université de Nice-Sophia Antipolis



# Remerciements

Après ces longs mois de frappe de thèse, je désire écrire une thèse de remerciements.

*Chapitre 1 - Introduction* – La thèse est un parcours d’une personne au milieu d’un environnement favorable. Je voudrais maintenant à travers ces quelques lignes que les personnes indispensables à ce travail soient remerciées.

*Chapitre 2 - État de l’art* – J’ai lu tous les remerciements depuis 1976, aucun ne contient les personnes que j’aimerais remercier. Cette contribution est donc entièrement innovante s’appuyant sur des idées et des sentiments tout personnels.

*Chapitre 3 - Équipe* – Bien que l’expérience ne soit pas reproductible et les effectifs trop faibles pour comparer à d’autres équipes, je remercie l’équipe T.E.A. de son haut degré d’accueil, d’énergie et de science. Les relations avec un directeur de thèse sont toujours uniques et comment remercier Philippe Collard, toujours à l’écoute, disponible, et le petit sourire... Merci Philippe. Je remercie Manuel Clergue, nos discussions m’ont fait progresser humainement et scientifiquement. Merci à Cathy Escazut dont l’attention égale sa gentillesse. Je tiens Sincèrement à remercier Michaël qui est un ami maintenant, avec qui j’ai partagé un énorme paquet de sciences, de rêves et d’amitié. Merci aux futurs docteurs de l’équipe William et David!

Aussi très rapidement est apparue dans mon environnement une “co”-équipe à Lausanne, je remercie Marco Tomassini et Leonardo Vanneschi pour leur collaboration scientifique et sûrement plus.

*Chapitre 4 - Lieux* – L’IS ne se résume pas à une prise électrique, un câble réseau et une climatisation. C’est un lieu de vie où l’on parle de grands concepts scientifiques et des futilités du quotidien. Je remercie tous les itroissessiens et en particuliers les doctorants Stéphane, Éric, Karim, non j’arrête la liste tout de suite, je vais en oublier et remplir des chapitres entiers. Le mieux est de consulter le site web [adstic.free.fr](http://adstic.free.fr).

Je conclus ce chapitre lieux en remerciant tous ces cafés qui m’ont accueilli pour écrire, merci à ces débits.

*Chapitre 5 - Extras* – Je remercie mes parents qui m’ont soutenu, permis d’étudier pendant toutes ces années et qui m’ont apporté tout ce dont j’avais rêvé.

Hors du cadre et formidables, j’aimerais particulièrement mes amis, Rémi et Mélanie qui ont fait plus que m’accueillir au cours des derniers mois, Florence, Anne-thé, Jean-Mi, Julie, Franck indissociables du parcours pré et in thèse. Merci à Manu qui m’a permis de passer mon avant dernière étape de thèse.

Je remercie *Très Sincèrement* celle dont le prénom sonne comme des vers, l’alchimie et l’énergie qu’elle m’a apportées, ou non, à imprimer tout mon parcours de thèse.

*Chapitre 6 - Conclusion* – Je remercie Cyril Fonlupt, El-Gazhali Talbi et Marco Tomassini qui ont bien voulu émettre un avis favorable sur mes travaux. Mes remercie-

ments vont également à Enrico Formenti, président du Jury. Leurs remarques m'ont aidé à conclure ce travail de thèse et continue à orienter mes recherches.

Enfin, merci à tous ceux qui m'ont appris quelque chose ou éveillé ma curiosité.

# Table des matières

Table des figures	vii
Liste des tableaux	xiii
Introduction	1
<b>Chapitre 1 Paysages adaptatifs et Métaheuristiques</b>	<b>3</b>
1.1 Paysages adaptatifs et optimisation . . . . .	4
1.2 Exploration aléatoire du paysage adaptatif . . . . .	5
1.2.1 Recherche aléatoire . . . . .	5
1.2.2 Marche aléatoire . . . . .	6
1.3 Paysage multimodal et rugosité . . . . .	6
1.3.1 Multimodalité . . . . .	7
1.3.2 Rugosité . . . . .	8
1.4 Neutralité . . . . .	11
1.4.1 Origine du concept . . . . .	11
1.4.2 Neutralité dans les problèmes réels . . . . .	12
1.4.3 Paysages adaptatifs neutres académiques . . . . .	14
1.4.4 Influence sur la conception de métaheuristiques . . . . .	16
1.4.5 Pourquoi exploiter la neutralité? . . . . .	17
1.4.6 Neutralité synthétique . . . . .	18
1.5 Synthèse du chapitre . . . . .	20
<b>Chapitre 2 Ensemble de Neutralité : Nuage Adaptatif</b>	<b>21</b>
2.1 Définition et propriétés . . . . .	21
2.2 Modèle analytique relatif à une marche aléatoire . . . . .	23
2.2.1 Famille des paysages embarqués . . . . .	24
2.2.2 Expression analytique du NA sur la famille des <i>PEU</i> . . . . .	26
2.3 Généralisation à d'autres opérateurs . . . . .	35
2.3.1 Hill-Climbing . . . . .	35
2.3.2 Recuit Simulé . . . . .	42

2.4	Coefficient de pente négative . . . . .	48
2.4.1	Avantages / Inconvénients du NA . . . . .	48
2.4.2	Définition . . . . .	48
2.4.3	Résultats expérimentaux . . . . .	49
2.4.4	Amélioration : méthode de la bisection . . . . .	53
2.4.5	Synthèse du chapitre . . . . .	54
<b>Chapitre 3 Réseaux de Neutralité</b>		<b>57</b>
3.1	Mesures des paysages adaptatifs neutres . . . . .	57
3.1.1	Mesures existantes . . . . .	58
3.1.2	Nouvelle mesure : autocorrélation de l'évolvabilité . . . . .	59
3.2	Réseaux de neutralité sur les variantes des paysages NK . . . . .	60
3.2.1	Distribution du degré de neutralité . . . . .	61
3.2.2	Taille des réseaux de neutralité . . . . .	74
3.2.3	Nombre de réseaux de neutralité . . . . .	84
3.2.4	Taux d'innovation . . . . .	88
3.2.5	Autocorrélation de l'évolvabilité maximale . . . . .	96
3.3	Autres paysages adaptatifs neutres . . . . .	102
3.3.1	MAX-SAT . . . . .	102
3.3.2	Routes épistatiques . . . . .	108
3.4	Synthèse du chapitre . . . . .	113
<b>Chapitre 4 Dynamique et Métaheuristiques dans les problèmes neutres</b>		<b>115</b>
4.1	Dynamique des algorithmes évolutionnaires . . . . .	115
4.2	Métaheuristique dans les paysages neutres : Recherche Périscopique . . . . .	118
4.2.1	Algorithme . . . . .	118
4.2.2	Algorithmes de comparaisons . . . . .	119
4.2.3	Résultats expérimentaux . . . . .	122
4.3	Problème massivement neutre : AC du problème de majorité . . . . .	130
4.3.1	Automate cellulaire et problème de majorité . . . . .	131
4.3.2	Analyse du paysage adaptatif du problème de majorité . . . . .	133
4.3.3	Olympe . . . . .	142
4.3.4	Analyse de l'Olympe . . . . .	144
4.3.5	Algorithmes évolutionnaires sur l'Olympe . . . . .	149
4.3.6	Synthèse . . . . .	151
4.4	Synthèse du chapitre . . . . .	152
<b>Conclusion</b>		<b>155</b>







# Table des figures

1.1	Représentation d'un paysage adaptatif de dimension deux. . . . .	3
1.2	Données originales de Galton sur la taille des pois de senteur. . . . .	8
1.3	Représentation classique d'un paysage adaptatif neutre de dimension deux. . . . .	12
2.1	Nuage adaptatif d'un paysage NK de paramètres $N = 25$ et $K = 20$ relativement à l'opérateur local de recherche aléatoire. . . . .	22
2.2	Exemple d'évolvabilité pour la valeur d'adaptation $\varphi = 0.6$ . . . . .	23
2.3	Contour et courbe moyenne du nuage adaptatif d'un paysage NK de paramètre $N = 25$ et $K = 20$ relativement à l'opérateur local de recherche aléatoire. . . . .	24
2.4	Nombre moyen $E(\gamma^d)$ de contributions affectées en fonction du nombre $d$ de bits changeant de valeur pour un paysage NK avec $N = 25$ et différentes valeurs de $K$ . . . . .	30
2.5	Distribution de probabilité $\gamma^d(n)$ pour un paysage NK avec $d = 1$ , $N = 25$ , $K = 5$ (a) et $K = 20$ (b). . . . .	31
2.6	Densités conditionnelles théoriques pour différentes valeurs de $\varphi$ pour un paysage NK avec $N = 32$ , $K = 8$ et $d = 1$ . . . . .	32
2.7	Contour du NA pour le paysage NK avec $N = 25$ , $K = 5$ (a) et $K = 20$ (b). . . . .	33
2.8	Écart-types expérimentaux et théoriques pour différents paysages NK. . . . .	34
2.9	Contour du NA pour le paysage Max-3-SAT avec $k = 3$ , $n = 20$ , $m = 91$ (a) et $n = 50$ , $m = 218$ (b). . . . .	34
2.10	Contour du NA avec l'opérateur HC pour un paysage NK avec $N = 25$ , $K = 5$ (a) et $K = 20$ (b). . . . .	36
2.11	Trajectoire moyenne (avec l'écart-type) de l'itération de l'opérateur HC pour différents paysages NK et courbe calculée à partir de la courbe moyenne du NA. . . . .	39
2.12	Contour du NA avec l'opérateur HC pour les paysages MAX-SAT. . . . .	40
2.13	Trajectoire moyenne (avec l'écart-type) de l'itération de l'opérateur HC pour différents paysages MAX-3-SAT et courbe calculée à partir de la courbe moyenne du NA. . . . .	41
2.14	Contour du NA relativement à l'opérateur RS aux trois températures $T = 0.1$ , $T = 0.05$ et $T = 0.01$ (de haut en bas) pour le paysage NK avec $N = 32$ , $K = 4$ (colonne de gauche) et $K = 8$ (colonne de droite). . . . .	44
2.15	Trajectoire moyenne (avec l'écart-type) de l'itération de l'opérateur RS et courbe attendue grâce à la courbe moyenne du NA aux trois températures $T = 0.1$ , $T = 0.05$ et $T = 0.01$ (de haut en bas) pour le paysage NK avec $N = 32$ , $K = 4$ (colonne de gauche) et $K = 8$ (colonne de droite). . . . .	45
2.16	Contour du NA relativement à l'opérateur RS aux trois températures $T = 2.6$ , $T = 1.3$ et $T = 0.75$ (de haut en bas) pour le paysage Max-3-SAT avec $N = 50$ , $m = 218$ (colonne de gauche) et $N = 100$ , $m = 430$ (colonne de droite). . . . .	46

2.17	Trajectoire moyenne (avec l'écart-type) de l'itération de l'opérateur RS et courbe attendue grâce à la courbe moyenne du NA aux trois températures $T = 2.6$ , $T = 1.3$ et $T = 0.75$ (de haut en bas) pour le paysage Max-3-SAT avec $N = 50$ , $m = 218$ (colonne de gauche) et $N = 100$ , $m = 430$ (colonne de droite). . . . .	47
2.18	Nuage adaptatif et segments moyens pour le problème binomial-3 pour différentes valeurs de $a_{\mathcal{R}}$ . (a) : $a_{\mathcal{R}} = 1$ , (b) : $a_{\mathcal{R}} = 10$ , (c) : $a_{\mathcal{R}} = 10^2$ et (d) : $a_{\mathcal{R}} = 10^3$ . . . . .	50
2.19	CPN et nuage adaptatif pour le problème de parité paire pour différent nombre de variables : (a) $k = 3$ , (b) $k = 5$ , (a) $k = 7$ , (b) $k = 9$ . . . . .	51
2.20	CPN et nuage adaptatif pour le problème de la fourmi artificielle pour deux profondeurs d'arbre (a) : profondeur 10, (b) : profondeur 6. . . . .	52
3.1	Représentation sous forme de graphe d'un paysage adaptatif neutre. . . . .	59
3.2	Exemple de moyenne de distribution du degré de neutralité pour différentes valeurs des paramètres (bâton) et distribution binomiale de même moyenne (ligne). . . . .	63
3.3	Degré de neutralité moyen en fonction des paramètres des paysages. On trouve de haut en bas la famille des paysages $NK_q$ , $NK_M$ et $NK_p$ . . . . .	68
3.4	Fonctions d'autocorrélation des degrés de neutralité pour différentes valeurs du paramètre $K$ et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages $NK_q$ . . . . .	69
3.5	Autocorrélation des degrés de neutralité pour différentes valeurs du paramètre $K$ et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages $NK_M$ . . . . .	70
3.6	Autocorrélation des degrés de neutralité pour différentes valeurs du paramètre $K$ et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages $NK_p$ . . . . .	71
3.7	Coefficient d'autocorrélation d'ordre 1 des degrés de neutralité pour les paysages $NK_q$ , $NK_M$ et $NK_p$ avec $N = 64$ . . . . .	72
3.8	Échelle empirique de corrélation des degrés de neutralité en fonction du coefficient d'autocorrélation d'ordre 1. . . . .	73
3.9	Profils rang-taille des RN pour les paysages $NK_q$ . Le paramètre $q$ prend les valeurs 2, 3, 4, 10 de haut en bas, et le paramètre $K$ les valeurs 1, 2, 3 de gauche à droite. . . . .	76
3.10	Profils rang-taille des RN pour les paysages $NK_q$ . Le paramètre $q$ prend les valeurs 2, 3, 4, 10 de haut en bas, et le paramètre $K$ les valeurs 5, 8 de gauche à droite. . . . .	77
3.11	Profils rang-taille pour les paysages $NK_M$ . Le paramètre $M$ prend les valeurs 16, 32, 48, 160 de haut en bas, et le paramètre $K$ les valeurs 1, 2, 3 de gauche à droite. . . . .	78
3.12	Profils rang-taille pour les paysages $NK_M$ . Le paramètre $M$ prend les valeurs 16, 32, 48, 160 de haut en bas, et le paramètre $K$ les valeurs 5, 8 de gauche à droite. . . . .	79
3.13	Profils rang-taille pour les paysages $NK_p$ . Le paramètre $p$ prend les valeurs 0.5, 0.8, 0.9, 0.95, 0.99 de haut en bas, et le paramètre $K$ les valeurs 1, 2, 3 de gauche à droite. . . . .	80
3.14	Profils rang-taille pour les paysages $NK_p$ . Le paramètre $p$ prend les valeurs 0.5, 0.8, 0.9, 0.95, 0.99 de haut en bas, et le paramètre $K$ les valeurs 5, 8 de gauche à droite. . . . .	81
3.15	Taille moyenne sur les 50 instances de paysages du plus grand réseau de neutralité normalisée par la taille de l'espace de recherche de $2^{16}$ . On trouve de haut en bas la famille des paysages $NK_q$ , $NK_M$ et $NK_p$ . . . . .	82

3.16	Rapport de l'écart-type par la moyenne de la taille du plus grand réseau de neutralité sur les 50 instances de paysages. On trouve de haut en bas la famille des paysages $NK_q$ , $NK_M$ et $NK_p$ . . . . .	83
3.17	Nombre de réseaux de neutralité par valeur de performance (les quatre graphiques supérieurs) et densité des états (graphique inférieur) pour différentes valeurs des paramètres pour les paysages $NK_q$ . . . . .	85
3.18	Nombre de réseaux de neutralité par valeur de performance (les quatre graphiques supérieurs) et densité des états (graphique inférieur) pour différentes valeurs des paramètres pour les paysages $NK_M$ . . . . .	86
3.19	Nombre de réseaux de neutralité par valeur de performance (les quatre graphiques supérieurs et en bas à gauche) et densité des états (graphique en bas à droite) pour différentes valeurs des paramètres pour les paysages $NK_p$ . . . . .	87
3.20	Exemple de courbes d'innovation cumulative lors d'une marche neutre sur un RN ( $C_n$ ), et lors d'une marche aléatoire dans l'ensemble du paysage ( $C_a$ ) pour un paysage $NK_q$ avec $K = 2$ et $q = 2$ (a) et pour un paysage $NK_M$ avec $K = 2$ et $M = 16$ (b). . . . .	88
3.21	Fonctions d'autocorrélation de l'évolvabilité maximale pour différentes valeurs du paramètre $K$ et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages $NK_q$ avec $N = 16$ . . . . .	97
3.22	Fonctions d'autocorrélation de l'évolvabilité maximale pour différentes valeurs du paramètre $K$ et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages $NK_M$ avec $N = 16$ . . . . .	98
3.23	Fonctions d'autocorrélation de l'évolvabilité maximale pour différentes valeurs du paramètre $K$ et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages $NK_p$ avec $N = 16$ . . . . .	99
3.24	Coefficient d'autocorrélation d'ordre 1 de l'évolvabilité maximale pour les paysages $NK_q$ , $NK_M$ et $NK_p$ avec $N = 64$ . . . . .	101
3.25	Distribution des degrés de neutralité pour $m = 69$ (a) et moyenne et écart-type des distributions de degré de neutralité (b) pour les paysages MAX-3-SAT pour $N = 16$ . . . . .	103
3.26	Fonctions d'autocorrélation des degrés de neutralité pour différentes valeurs du paramètre $m$ pour les paysages MAX-3-SAT pour $N = 16$ . . . . .	103
3.27	Profils rang-taille des RN pour les paysages MAX-3-SAT pour $N = 16$ et différentes valeur de $m$ . . . . .	105
3.28	Moyenne (a) et écart-type (b) du plus grand RN pour les paysages MAX-3-SAT pour $N = 16$ . . . . .	106
3.29	Fonctions d'autocorrélation de l'évolvabilité maximale pour différentes valeurs du paramètre $m$ pour les paysages MAX-3-SAT pour $N = 16$ . . . . .	107
3.30	Distribution des degrés de neutralité pour différentes valeurs des paramètres des paysages RE. . . . .	109
3.31	Fonctions d'autocorrélation et coefficient d'autocorrélation d'ordre 1 (en bas à droite) des degrés de neutralité pour différentes valeurs des paramètres des paysages RE. . . . .	110
3.32	Fonctions d'autocorrélation de l'évolvabilité maximale pour différentes valeurs du paramètre $K$ sur les paysages RE avec $N = 16$ et $b = 2$ . . . . .	112
3.33	Coefficient d'autocorrélation de l'évolvabilité maximale d'ordre 1 pour différentes valeurs des paramètres des paysages RE. . . . .	112

4.1	Exemple d'évolution de performance d'une population lors d'une dynamique d'équilibres ponctuels. . . . .	116
4.2	Illustration de la recherche périscopique : la recherche périscopique alterne une phase de mouvements neutres jusqu'à trouver un maximum local neutre avec un saut qualitatif de performance. . . . .	120
4.3	Illustration de la recherche périscopique : pendant la phase de mouvements neutres, l'algorithme sélectionne la solution voisine du réseau de neutralité dont l'évolvabilité est la plus grande. . . . .	120
4.4	Performances moyennes pour les différentes métaheuristiques pour les paysages $NK_q$ avec $N = 16$ . . . . .	124
4.5	Performances moyennes pour les différentes métaheuristiques pour les paysages $NK_M$ avec $N = 16$ . . . . .	125
4.6	Performances moyennes pour les différentes métaheuristiques pour les paysages $NK_p$ avec $N = 16$ . . . . .	126
4.7	Performances moyennes pour les différentes métaheuristiques pour les paysages $NK_q$ avec $N = 64$ . . . . .	127
4.8	Performances moyennes pour les différentes métaheuristiques pour les paysages $NK_M$ avec $N = 64$ . . . . .	128
4.9	Performances moyennes pour les différentes métaheuristiques pour les paysages $NK_p$ avec $N = 64$ . . . . .	129
4.10	Diagramme espace-temps pour la règle GLK. La densité de 0 est 0.476 pour (a) et 0.536 pour (b). L'état 0 est représenté en blanc et le 1 en noir. . . . .	131
4.11	Erreur de l'évaluation de la performance standard donné par un t-test pour un échantillon de taille $n = 10^4$ . $isNeutral(s, s')$ est vrai si la différence de performance entre les deux solutions est en dessous la courbe. . . . .	134
4.12	D.O.S. obtenus par échantillonnage équiprobable de l'espace de recherche (a) et en utilisant l'algorithme de Métropolis-Hastings (b). . . . .	134
4.13	Nuage adaptatif et segments utilisés pour calculer le CPN. L'algorithme de Métropolis-Hastings a été utilisé pour créer l'échantillon de solutions. . . . .	135
4.14	Distribution du degré de neutralité au cours des marches neutres sur $RN_{0.5}$ (a) et $RN_{0.76}$ (b). . . . .	136
4.15	Estimation de la fonction d'autocorrélation des degrés de neutralité pour les marches neutres pour $RN_{0.5}$ (a) et pour $RN_{0.76}$ (b). . . . .	136
4.16	Innovation cumulative neutre au cours des marches neutres pour $RN_{0.5}$ (a) et $RN_{0.76}$ (b). . . . .	137
4.17	Centroïde $C$ de six $moc$ . Les carrés indiquent la fréquence de la valeur 1 pour les six $moc$ en fonction de la position du bit. La colonne de droite indique le nombre de bit de $C$ parmi les 128 qui ont la même fréquence de 1 indiquée par la colonne de gauche. . . . .	139
4.18	Profil d'évolvabilité pour les six meilleures optima locaux connus Pour chaque optima, la ligne pointillé indique sa performance. La colonne $r$ et la pente $m$ (voir texte) sont reportées sous chaque figure. . . . .	140
4.19	Centroïde des $moc'$ . Les carrés indiquent la fréquence de la valeur 1 pour les six $moc$ en fonction de la position du bit. La colonne de droite indique le nombre de bit de $C$ parmi les 128 qui ont la même fréquence de 1 indiquée par la colonne de gauche. . . . .	144
4.20	Distance entre les $moc$ et le centroïde $C$ (a) et entre le centroïde $C'$ et les $moc'$ (b). . . . .	144

4.21	Moyenne et écart-type de l'évolvabilité par bit pour les <i>moc</i> (a) et pour les <i>moc'</i> (b). Les barres verticales en dessous des figures indiquent les bits fixés des schéma <i>S</i> (a) et <i>S'</i> (b). . . . .	145
4.22	Densité des états (a) et degré de neutralité des solutions en fonction de leur performance (b) pour l'Olympe. $10^3$ solutions ont été échantillonnées et ont été évaluées sur un échantillon de CI de taille $10^4$ . . . . .	145
4.23	Nuage de point du CPD calculé avec la distance euclidienne du centroïde $C'$ . Deux échantillons de solutions de taille $10^4$ sont générés : <i>Osample</i> (a) et <i>Csample</i> (b). . . . .	146
4.24	Fonction d'autocorrelation (a) et d'autocorrelation partielle (b) d'une marche aléatoire sur l'Olympe. . . . .	147
4.25	Fonction d'autocorrelation des résidus (a) et p-valeur de la statistique Ljung-Box (b) pour le modèle <i>ARMA</i> (2, 1). . . . .	148
4.26	Nuage adaptatif et segments utilisés pour calculer le CPN sur l'Olympe. . . . .	149
4.27	Pourcentage d'exécutions (a) et nombre de générations nécessaires (b) à l'émergence d'une solution dont la performance est supérieure ou égale au seuil de performance reporté en abscisse. . . . .	150
4.28	Distance de Hamming moyenne entre les solutions de la population en fonction des générations. . . . .	151



# Liste des tableaux

2.1	Moyenne et écart-type sur 300 instances des paramètres de la droite moyenne du NA $\tilde{E}(\varphi) = b + a(\varphi - b)$ pour les paysages NK avec $d = 1$ . . . . .	31
2.2	Moyenne et écart-type sur 100 instances des paramètres de la droite moyenne du NA $\tilde{E}(\varphi) = b + a(\varphi - b)$ pour le paysage Max-3-SAT avec $d = 1$ . . . . .	33
2.3	Résultats expérimentaux de la courbe moyenne $\tilde{E}(\varphi) = a\varphi + b$ relative à l'opérateur HC sur les paysages NK pour différentes valeurs de $N$ et $K$ . . . . .	35
2.4	Valeur moyenne et écart-type du point d'intersection $\beta$ du NA comparée avec la valeur moyenne du point de convergence $\beta^*$ de l'itération de l'opérateur HC pour les paysages NK. . . . .	37
2.5	Résultats expérimentaux de la courbe moyenne $\tilde{E}(\varphi) = a\varphi + b$ relative à l'opérateur HC sur les paysages MAX-SAT pour différentes valeurs de $N$ et $m$ . . . . .	37
2.6	Valeur moyenne et écart-type du point d'intersection $\beta$ du NA comparée avec la valeur moyenne du point de convergence $\beta^*$ de l'itération de l'opérateur HC pour les paysages MAX-3-SAT. . . . .	38
2.7	Résultats expérimentaux (moyenne et écart-type sur 100 instances de paysage) de la droite de régression $\tilde{E}(\varphi) = a\varphi + b$ relative à l'opérateur RS des paysages MAX-SAT pour différentes valeurs de $N$ et $m$ et de température $T$ . . . . .	43
2.8	CPN pour le problème binomial-3 pour différentes valeurs de $a_{\mathcal{R}}$ . . . . .	51
2.9	CPN pour le problème binomial-3 pour différentes valeurs de $a_{\mathcal{R}}$ . . . . .	52
2.10	Valeur du CPN pour les trois problèmes : multiplexeurs, Spirales enroulées, et arbres royaux (AR). . . . .	53
3.1	Moyenne p-valeur du test du chi2 et nombre de tests vérifiés au seuil de 5% pour les distributions du degré de neutralité des familles de paysages $NK_q$ et $NK_M$ . . . . .	62
3.2	Espérance ( $E$ ) et écart-type ( $\sigma$ ) des distributions du degré de neutralité pour les familles de paysages $NK_q$ , $NK_M$ et $NK_p$ pour $N = 16$ . . . . .	64
3.3	Résultats pour les paysages $NK_q$ , $NK_M$ et $NK_p$ de la régression de la forme $D = a/x + b$ où $D$ est le degré de neutralité moyen du paysage et $x$ l'un des paramètres de neutralité $q$ , $M$ ou $p$ . . . . .	64
3.4	Espérance ( $E$ ) et écart-type ( $\sigma$ ) des distributions du degré de neutralité pour les familles de paysages $NK_q$ , $NK_M$ et $NK_p$ pour $N = 64$ . . . . .	66
3.5	Régressions linéaires des innovations cumulatives neutre et aléatoire $\log(C_n(t)) = a_n \log(t) + b_n$ et $\log(C_a(t)) = a_a \log(t) + b_a$ pour les paysages $NK_q$ avec $N = 16$ . Le coefficient de corrélation est noté $\rho$ . . . . .	89
3.6	Régressions linéaires des innovations cumulatives neutre et aléatoire $\log(C_n(t)) = a_n \log(t) + b_n$ et $\log(C_a(t)) = a_a \log(t) + b_a$ pour les paysages $NK_M$ avec $N = 16$ . Le coefficient de corrélation est noté $\rho$ . . . . .	90



3.7	Régressions linéaires des innovations cumulatives neutre et aléatoire $\log(C_n(t)) = a_n \log(t) + b_n$ et $\log(C_a(t)) = a_a \log(t) + b_a$ pour les paysages $NK_p$ avec $N = 16$ . Le coefficient de corrélation est noté $\rho$ .	91
3.8	Régressions linéaires des innovations cumulatives neutre et aléatoire $\log(C_n(t)) = a_n \log(t) + b_n$ et $\log(C_a(t)) = a_a \log(t) + b_a$ pour les paysages $NK_q$ avec $N = 64$ . Le coefficient de corrélation est noté $\rho$ .	92
3.9	Régressions linéaires des innovations cumulatives neutre et aléatoire $\log(C_n(t)) = a_n \log(t) + b_n$ et $\log(C_a(t)) = a_a \log(t) + b_a$ pour les paysages $NK_M$ avec $N = 64$ . Le coefficient de corrélation est noté $\rho$ .	93
3.10	Régressions linéaires des innovations cumulatives neutre et aléatoire $\log(C_n(t)) = a_n \log(t) + b_n$ et $\log(C_a(t)) = a_a \log(t) + b_a$ pour les paysages $NK_p$ avec $N = 64$ . Le coefficient de corrélation est noté $\rho$ .	94
3.11	Valeur du coefficient d'autocorrélation d'ordre 1 pour les paysages MAX-3-SAT pour $N = 16$ .	103
3.12	Régressions linéaires des innovations cumulatives neutre et aléatoire $\log(C_n(t)) = a_n \log(t) + b_n$ et $\log(C_a(t)) = a_a \log(t) + b_a$ pour les paysages MAX-3-SAT. Le coefficient de corrélation est noté $\rho$ .	104
3.13	Valeur des coefficient d'autocorrélation d'ordre 1 de l'évolvabilité maximale pour les paysages MAX-3-SAT pour $N = 16$ .	106
3.14	Espérance ( $E$ ) et écart-type ( $\sigma$ ) des distributions du degré de neutralité pour différentes valeurs des paramètres des paysages RE.	109
3.15	Régressions linéaires des innovations cumulatives neutre et aléatoire $\log(C_n(t)) = a_n \log(t) + b_n$ et $\log(C_a(t)) = a_a \log(t) + b_a$ pour les paysages RE. Le coefficient de corrélation est noté $\rho$ .	111
4.1	CPD pour les six meilleurs optima connus, calculés à partir d'un échantillon de taille $4 \cdot 10^3$ en utilisant l'algorithme de Métropolis-Hastings.	135
4.2	Description et performance standard des six meilleures règles connues ( $moc$ ) calculées sur un échantillon de CI de $10^4$ .	138
4.3	Distances entre les six meilleurs optima locaux connus	138
4.4	Description des six symétriques des meilleurs optima locaux connus ( $moc'$ ).	143
4.5	Distances entre les symétriques des meilleurs optima locaux connus ( $moc'$ )	143
4.6	CPD où la distance est calculée à partir de l'un des $moc'$ , ou le plus proche des $moc'$ , ou à partir du centroïde $C'$ . Deux échantillons de solutions de taille $10^4$ sont générées : $Osample$ et $Csample$ .	146
4.7	Performances de AE calculés sur un échantillon de CI de taille $10^4$ .	150
A.1	Moyenne et écart-type des performances des différents algorithmes sur les paysages $NK_q$ pour $N = 16$ .	158
A.2	Moyenne et écart-type des performances des différents algorithmes sur les paysages $NK_q$ pour $N = 64$ .	159
A.3	Moyenne et écart-type des performances des différents algorithmes sur les paysages $NK_M$ pour $N = 16$ .	160
A.4	Moyenne et écart-type des performances des différents algorithmes sur les paysages $NK_M$ pour $N = 64$ .	161
A.5	Moyenne et écart-type des performances des différents algorithmes sur les paysages $NK_p$ pour $N = 16$ .	162
A.6	Moyenne et écart-type des performances des différents algorithmes sur les paysages $NK_p$ pour $N = 64$ .	163

# Introduction

Cette thèse trouve naturellement sa première source d’inspiration dans les travaux traitant des paysages adaptatifs, des algorithmes évolutionnaires et des métaheuristiques pour l’optimisation combinatoire. Mais au delà de ces références précises, elle est une contribution au vaste domaine des réseaux et de la complexité.

Depuis plusieurs dizaines d’années, les réseaux apparaissent comme des universaux pour différentes sciences. Dans plusieurs domaines, on observe des structures communes basées sur des entités de même nature dont les interactions font émerger des comportements complexes. C’est par exemple le cas des réseaux immunitaires en biologie, du réseau internet en informatique, des réseaux de villes en géographie, des réseaux de réactions chimiques, des réseaux sémantiques en linguistique, etc.

Le concept de *paysage adaptatif* a été introduit par S. Wright dans le domaine de la biologie de l’évolution dans les années 1930. Il consiste à représenter l’ensemble de tous les organismes potentiels à l’aide d’un espace abstrait muni d’une relation de voisinage. Les organismes potentiels sont les points du paysage et la relation de voisinage définit les déplacements possibles sur celui-ci. De plus, à chaque organisme potentiel est associé une “hauteur” indiquant son taux de réplication appelé sa valeur d’adaptation ou fitness. L’évolution d’une population s’envisage alors comme une marche adaptative d’une population vers les plus hauts sommets c’est-à-dire les grandes valeurs d’adaptation. On décrit alors le paysage à l’aide de pics, crêtes, vallées, etc.

Le paysage adaptatif est l’un des concepts les plus pertinents pour modéliser l’évolution d’une population d’organismes. Cette notion s’est aussi imposée dans d’autres domaines scientifiques tels que la biologie moléculaire, la physique statistique, ou l’optimisation combinatoire pour modéliser des systèmes dynamiques complexes. Dans le domaine de l’optimisation combinatoire qui nous concerne dans ce mémoire, l’intérêt même de cette métaphore est de lier la description géométrique avec la dynamique des algorithmes de recherche.

Au cours des années 60, grâce à la naissance et aux progrès de la biologie moléculaire, Kimura a mis en évidence que la majorité des mutations sont neutres ou bien léthales ; dans ce dernier cas, elles conduisent les organismes à disparaître avant de pouvoir se reproduire. Les mutations neutres n’apportent pas d’avantage sélectif à l’organisme ; elles sont uniquement dirigées par le hasard et leur taux d’apparition serait constant au cours de l’évolution. L’existence des mutations neutres permet d’expliquer le polymorphisme, c’est-à-dire l’importante variabilité génétique observée au sein des populations de nombreuses espèces. À la différence du modèle néodarwinien où de meilleurs variants sont continuellement découverts et où une population s’adapte progressivement à son environnement, le modèle d’évolution neutre suppose que pendant de longues périodes de temps l’évolution n’est plus guidée par l’adaptation mais seulement par des variations aléatoires.

Cette théorie neutraliste de l’évolution<sup>1</sup> a modifié la perception du paysage adaptatif. On décrit alors la géométrie du paysage à l’aide de “plateaux” où la recherche “dérive” aléatoirement. Ces paysages avec de nombreux plateaux où sont observés ces dynamiques sont appelées *paysages*

---

<sup>1</sup>qu’il préfère nommer théorie de la mutation et de la dérive aléatoire

*adaptatifs neutres.*

Les réseaux s'avèrent alors pertinentes dans l'étude des paysages adaptatifs neutres. On définit les *réseaux de neutralité* comme les graphes des points du paysage de même performance où les arcs sont définis par la relation de voisinage induite par les mutations neutres. Les réseaux de neutralité correspondent aux plateaux du paysage. L'objet de ce mémoire est d'étudier les paysages adaptatifs neutres et en particulier les réseaux de neutralité. On utilisera les outils issus de diverses disciplines comme l'autocorrélation des degrés du graphe, les profils rang-taille des réseaux, le taux d'innovation, etc.

Le premier chapitre exposera l'origine du concept de paysage adaptatif et ces deux principales géométries, multimodale et neutre. Nous donnerons les principales définitions et mesures usuelles des paysages adaptatifs liées à la multimodalité. Ensuite, nous montrerons que les paysages adaptatifs neutres se rencontrent dans le domaine de l'optimisation et nous exposerons les principaux travaux relatifs à cette géométrie qui soit tentent d'"introduire" de la neutralité dans le codage du problème afin de faciliter la recherche, soit conçoivent des algorithmes dédiés.

Dans le deuxième chapitre, nous proposerons un nouvel outil, le *Nuage Adaptatif* (NA), pour étudier les ensembles de neutralité, *i.e.* l'ensemble des solutions de même performance. Le nuage adaptatif qui exprime la corrélation de performance entre solutions voisines relativement à un opérateur local permet l'analyse de l'effet d'un opérateur sur les ensembles de neutralité. Nous donnerons l'expression analytique du nuage adaptatif relativement à l'opérateur de recherche aléatoire sur une large famille de paysages, les paysages *embarqués uniformes*, et nous validerons expérimentalement cette expression. Nous verrons comment le nuage adaptatif permet la prédiction de l'évolution de performance lors de l'itération d'un opérateur local. Enfin, nous présenterons une mesure de difficulté déduite du NA, le *Coefficient de pente négative*, particulièrement adapté au domaine de la programmation génétique.

Dans le troisième chapitre, nous proposerons une étude expérimentale des réseaux de neutralité de paysages académiques de neutralité ajustable représentant différentes manières d'obtenir de la neutralité dans un paysage additif. L'étude utilisera des mesures existantes de réseaux mais aussi une nouvelle mesure, l'*autocorrélation de l'évolvabilité*. Cette mesure complète la description des paysages en décrivant ces interactions entre les différents réseaux de neutralité.

Cette nouvelle mesure mettra en évidence de nouvelles caractéristiques des paysages adaptatifs neutres ce qui permettra de concevoir un nouvel algorithme de recherche, la *recherche périscopique*. Le quatrième chapitre exposera les performances de cet algorithme sur les paysages neutres académiques étudiés au chapitre précédent. Enfin, le chapitre se termine par l'analyse d'un paysage neutre relatif au *problème de majorité*, qui est un problème d'apprentissage d'une tâche par un automate cellulaire. L'étude de ce problème difficile et la prise en compte de sa géométrie neutre permet de découvrir un sous-espace sur lequel l'optimisation est facilitée.

Ce travail d'informaticien n'a été possible que par l'utilisation massive de la puissance de calcul des ordinateurs. C'est une démarche expérimentale qui a exigé un travail approfondi dans le choix des expériences à mener et de leurs interprétations. Mais c'est aussi une démarche théorique qui a nécessité l'utilisation d'outils mathématiques comme les statistiques, les probabilités ou les graphes. Cette thèse bien que relevant principalement de l'informatique se veut un travail pluridisciplinaire dans ses motivations et le choix de ses références, dans sa démarche et ses outils et dans ses résultats qui, nous l'espérons, pourront être exploités dans différents domaines.

# Chapitre 1

## Paysages adaptatifs et Métaheuristiques

Le concept de *paysage adaptatif* (ou paysage de fitness) a été introduit par S. Wright dans le domaine de la biologie de l'évolution dans les années 1930 [146]. Il consiste à représenter l'ensemble de tous les organismes potentiels à l'aide d'un espace abstrait muni d'une relation de voisinage. Les organismes potentiels sont les points du paysage et la relation de voisinage définit les déplacements possibles sur celui-ci. Pour compléter cette image, à chaque organisme potentiel est associé une "hauteur" indiquant son taux de réplication appelé sa valeur d'adaptation ou fitness. la figure 1.1 est la représentation classique d'un paysage adaptatif avec des pics et des vallées.

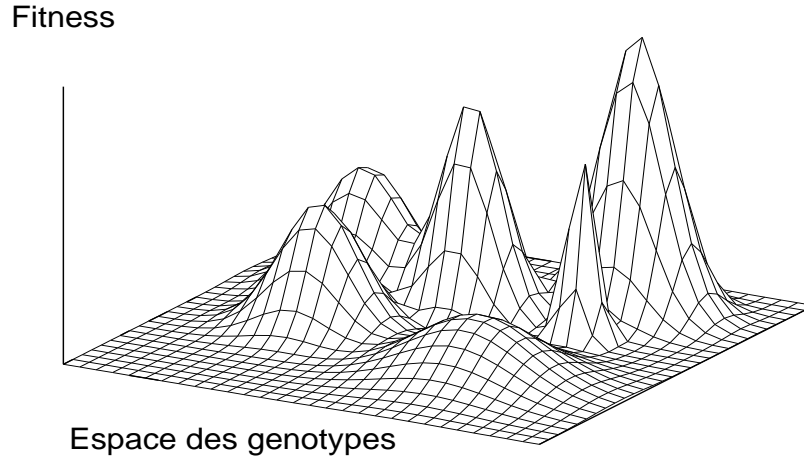


FIG. 1.1 – Représentation d'un paysage adaptatif de dimension deux.

Le paysage adaptatif est l'un des concepts les plus pertinents pour modéliser l'évolution d'une population d'organismes. Cette notion s'est aussi imposée dans d'autres domaines de la science tels que la biologie moléculaire, la physique statistique [69, 99, 39], ou l'optimisation combina-

toire pour modéliser des systèmes dynamiques complexes. Dans le domaine de l’optimisation combinatoire, l’intérêt même de cette métaphore est de lier la description géométrique avec la dynamique des algorithmes de recherche et ainsi d’extraire des informations sur la difficulté à optimiser un problème donné.

## 1.1 Intérêt de la métaphore des paysages adaptatifs pour l’optimisation

Dans le domaine de l’optimisation combinatoire à l’aide de métaheuristiques comme par exemple le recuit simulé (RS), la recherche taboue (RT) ou les algorithmes évolutionnaires (AE), on utilise également le concept de paysage adaptatif en l’associant à un problème d’optimisation. Dans ce cas, l’ensemble des organismes potentiels correspond à l’ensemble des solutions potentielles du problème ; deux solutions sont dites voisines lorsqu’il existe un opérateur local permettant de passer de l’une à l’autre ; enfin, la valeur d’adaptation représente l’objectif à maximiser (ou le coût à minimiser). La valeur de la fonction à optimiser est parfois, notamment dans le cas du recuit simulé, assimilée à une énergie que l’on cherche à minimiser. Cependant dans la suite de ce mémoire, nous considérerons que toutes les fonctions objectifs sont à maximiser, nous nommerons les “solutions potentielles” plus simplement “solutions” et la valeur d’adaptation d’une solution sera aussi appelée performance d’une solution.

Formellement, nous utiliserons la définition et les notations usuelles [125, 106, 124] :

**Définition:** Un *paysage adaptatif* est un triplet  $(\mathcal{S}, \mathcal{V}, f)$  tel que :

- $\mathcal{S}$  est un ensemble de solutions potentielles,
- $\mathcal{V} : \mathcal{S} \rightarrow 2^{\mathcal{S}}$  est un voisinage qui associe à chaque solution  $s \in \mathcal{S}$  un ensemble de solution voisines  $\mathcal{V}(s) \subset \mathcal{S}$ ,
- $f : \mathcal{S} \rightarrow \mathbb{R}$  une fonction d’adaptation qui à chaque solution associe un nombre réel.

Dans le cadre des métaheuristiques par recherche locale, le ou les opérateurs locaux permettent de définir le voisinage  $\mathcal{V}$ . Si la métaheuristique utilise un seul opérateur local  $op$ , le voisinage d’une solution  $x$  est souvent défini comme  $\mathcal{V}(x) = \{y \in \mathcal{S} \mid y = op(x)\}$ . Dans le cas où plusieurs opérateurs locaux sont utilisés, on peut alors soit associer un paysage adaptatif à chaque opérateur ou alors construire l’ensemble des voisins comme l’ensemble des solutions atteignables par application d’un des opérateurs locaux. Un voisinage peut être associé à une distance ; par exemple, dans le contexte des algorithmes génétiques, quand l’ensemble des solutions est l’ensemble des chaînes binaires de longueur fixe, l’opérateur qui change la valeur d’un bit définit la relation de voisinage. Ainsi, deux solutions sont voisines lorsque leur distance de Hamming est égale à 1.

Les métaheuristiques “voient” un paysage adaptatif comme un substrat sur lequel les solutions courantes se déplacent. Le concept de paysage adaptatif a ainsi permis d’étudier la dynamique d’évolution des solutions, la convergence des métaheuristiques, ou la capacité des algorithmes à optimiser un problème. En formulant des hypothèses sur le paysage, la modélisation par chaînes de Markov permet d’étudier la convergence des métaheuristiques telles que le recuit simulé [1, 52], ou les algorithmes évolutionnaires [95, 29, 102]. Il est également possible de décrire un paysage indépendamment d’une métaheuristique particulière mais seulement en tenant compte des opérateurs de variation locaux. Les outils mathématiques qui permettent ce type d’étude sont alors, par exemple, le graphe  $\Gamma = (\mathcal{S}, \mathcal{V})$  induit par le paysage [106, 124] ou la décomposition en série de Walsh [68, 71]. Un paysage adaptatif peut être analysé de façon “géographique” ; on détaille alors ses caractéristiques : montagne massive, vallée, plateau, pic, crête... dans des espaces de dimension supérieure à trois. La description des paysages a été largement influencée par les théories de l’évolution. Deux types de géométries de paysage qui correspondent à deux dynamiques ont été mises en avant : la *marche adaptative* inspirée par la

théorie néo-darwinienne sur les paysages avec de nombreux pics et la *dérive aléatoire* inspirée par la théorie neutraliste sur les paysages avec de nombreux plateaux. Ces deux types de paysages ne sont évidemment pas exclusifs, et il existe une grande variété de géométries combinant les deux aspects.

Un autre intérêt des paysages adaptatifs est d'aider à mieux décrire un problème. Wolpert et Macready ont démontré le théorème “no free lunch” [144, 145] aux conséquences pratiques importantes. Il énonce que tous les algorithmes d'optimisation ont des performances moyennes équivalentes sur l'ensemble  $\mathcal{F}$  des fonctions d'optimisation de  $\mathcal{X}$  sur  $\mathcal{Y}$  où  $\mathcal{X}$  et  $\mathcal{Y}$  sont des ensembles finis. En conséquence, s'il existe un algorithme d'optimisation  $a_1$  dont les performances sont meilleures qu'un autre algorithme d'optimisation  $a_2$  sur un sous-ensemble de fonctions de  $\mathcal{F}$ , alors il existe un autre sous-ensemble de  $\mathcal{F}$  où l'algorithme  $a_1$  obtient de moins bonnes performances que l'algorithme  $a_2$ . Les conséquences pratiques de ce théorème font toujours l'objet de débat. Les travaux de Radcliffe *et al* et de Schumacher *et al* [104, 110] caractérisent les sous-ensembles de fonctions (sous-ensembles fermés par permutations) pour lesquels tous les algorithmes ont les mêmes performances moyennes. Igel et Toussaint [63] montrent que ces classes de fonctions sont rares parmi l'ensemble des classes possibles de fonctions et lorsque qu'il existe une notion de voisinage sur l'espace de recherche  $\mathcal{X}$ , aucune classe de fonctions qui respectent ce voisinage n'est fermée par permutation. Weinberg et Talbi [139] restreignent aussi le domaine d'application du théorème, le théorème n'est plus vérifié lorsque la classe de problèmes est structurée comme par exemple l'ensemble des problèmes de coloration de graphe. Toutefois, ce théorème renforce l'idée que l'élaboration d'un algorithme d'optimisation adapté nécessite de connaître la classe du problème que l'on optimise.

Pour un problème donné, il existe souvent plusieurs façons de représenter les solutions (vecteurs, listes, etc), de définir une relation de voisinage entre solutions, ou plusieurs choix pour la fonction d'adaptation. Chacun de ces choix précisent la nature du problème et surtout définissent un paysage adaptatif particulier avec ses propres caractéristiques. Une description statistique du paysage permet de discriminer les mauvais choix où peu d'algorithmes pourront optimiser correctement le problème, ou mieux de sélectionner le ou les meilleurs choix possibles qui seront favorables à une optimisation efficace par métaheuristique. Le paysage adaptatif est un moyen de caractériser un problème et de là, d'identifier l'algorithme qui exploite au mieux ces caractéristiques.

L'image d'un “paysage” est séduisante pour rendre compte de la dynamique de la recherche. Elle permet de visualiser des solutions traversant vallées, crêtes et plateaux vers des optima locaux ou globaux. Bien que cette métaphore soit une source féconde pour imaginer de nouveaux algorithmes ou pour se représenter leurs dynamiques, elle peut être trompeuse pour rendre compte des dynamiques dans des espaces comportant un grand nombre de dimensions. Il est nécessaire d'utiliser des outils mathématiques, notamment statistiques, pour caractériser la géométrie du paysage. Dans la suite, après avoir exposé les outils liés à la recherche aléatoire, nous allons présenter les géométries prépondérantes qui ont influencé la conception des métaheuristiques de recherche locale.

## 1.2 Exploration aléatoire du paysage adaptatif

### 1.2.1 Recherche aléatoire

La recherche aléatoire (RA) consiste à sélectionner une solution de manière équiprobable dans l'espace de recherche  $\mathcal{S}$ . L'étude de la RA présente deux intérêts. La RA sert d'algorithme de référence pour évaluer les performances d'un algorithme d'optimisation. En effet, une conséquence du théorème “no free lunch” est que tout algorithme d'optimisation a les mêmes

performances moyennes que la recherche aléatoire sur l'ensemble des problèmes. Sur un problème particulier, il est donc pertinent d'évaluer un algorithme par rapport à la recherche. La RA est aussi une méthode couramment utilisée pour fournir les solutions initiales de métaheuristiques telles que le recuit simulé, la recherche taboue, etc. L'analyse de la RA permet d'évaluer les performances initiales de ces métaheuristiques.

Nous présentons une première mesure liée à la recherche aléatoire. La densité des états, noté D.O.S pour *Density Of States*, a été introduite par Rosé *et al* [108] dans le domaine de l'optimisation. Cette mesure, issue de la physique des solides, correspond à la fréquence d'apparition d'une valeur d'adaptation (fitness) dans l'ensemble de l'espace de recherche  $\mathcal{S}$ . Elle donne ainsi la densité de probabilité d'obtenir une valeur d'adaptation donnée lors d'une RA. Expérimentalement, la densité des états est estimée efficacement à l'aide d'un échantillonnage obtenu par l'algorithme de Métropolis [82]. De nombreux travaux ont utilisé cette méthode, Crisan et Muhlenbein l'ont appliquée au problème d'affectation de fréquences [24], Bresina *et al* au problème de planification de tâches pour un télescope [16], Collins au problème de parité [23], Bélaidouni et Hao aux problèmes plus théoriques SAT et MAX-CSP [11, 12]. La D.O.S. est un indicateur de performance d'une RA ou d'une méthode d'initialisation uniforme sur l'espace de recherche. La forme de la décroissance (exponentielle, linéaire, etc) de la queue de la distribution est également un indice de difficulté pour optimiser : plus la décroissance est rapide, plus les bonnes solutions sont rares et l'optimisation difficile. Toutefois, la D.O.S. n'est pas suffisante comme indicateur puisqu'il est facile de concevoir des problèmes où les bonnes solutions sont rares mais où les métaheuristiques trouvent facilement une suite de solutions qui mène à l'optimum.

## 1.2.2 Marche aléatoire

La recherche aléatoire utilise uniquement l'ensemble des solutions potentielles  $\mathcal{S}$  du paysage adaptatif et ignore la notion de voisinage du paysage. Une autre façon de rechercher de façon aléatoire en utilisant la structure de voisinage  $\mathcal{V}$  est la *marche aléatoire* (MA). La MA est une suite de solutions voisines où la solution suivante est sélectionnée aléatoirement de manière uniforme dans le voisinage de la solution courante.

**Définition:** Pour tout paysage adaptatif  $(\mathcal{S}, \mathcal{V}, f)$ , une *marche*  $W$  de  $s$  à  $s'$  est une suite  $(s_0, s_1, \dots, s_m)$  de solution de  $\mathcal{S}$  telle que  $s_0 = s$  et  $s_m = s'$ , et  $s_{i+1} \in \mathcal{V}(s_i)$  pour tout  $0 \leq i < m$ . Une *marche aléatoire*  $W_{alea}$  est une marche de  $s$  à  $s'$  où pour tout  $i \geq 0$ ,  $s_{i+1}$  est sélectionné uniformément dans l'ensemble  $\mathcal{V}(s_i)$ .

La marche aléatoire n'a pas une meilleure performance moyenne que la recherche aléatoire sur un problème donné. Elle permet seulement l'étude du voisinage du point de vue de la performance comme nous le verrons plus précisément dans la partie 1.3.2. La MA donne une idée du "profil" du paysage. Lorsque ce profil est "accidenté", les performances des solutions voisines sont dispersées, au contraire, lorsque le profil est "lisse", ces performances sont proches. La continuité du voisinage a une influence sur les capacités de recherche des métaheuristiques basée sur ce voisinage.

## 1.3 Paysage multimodal et rugosité

Lorsque S. Wright proposa le concept de paysage adaptatif, il envisageait l'évolution comme une marche adaptative, *i.e.* où la solution suivante de la marche a une meilleure performance, d'une population vers les plus hauts sommets c'est-à-dire les grandes valeurs d'adaptation [146, 70]. Cette image est aussi celle qui a dominé dans la conception des métaheuristiques. Les solutions se déplacent progressivement vers les points du paysage de plus grandes adaptations,

par exemple au moyen d'un algorithme qui sélectionne toujours la solution voisine de meilleure performance. La convergence est obtenue lorsque les solutions sont bloquées sur des optima locaux ou globaux. La difficulté d'optimisation du problème est donc liée à l'existence d'optima locaux : leur nombre, leur densité spatiale, la taille des bassins d'attraction, etc. Nous allons tout d'abord présenter quelques mesures de paysage adaptatif relatives à ce critère. Ensuite, nous définirons la notion de rugosité liée à la corrélation locale et nous verrons le lien entre ces deux notions.

### 1.3.1 Multimodalité

Les optima locaux d'un paysage sont les solutions qui n'ont pas de solutions voisines de meilleure performance. On peut définir formellement cette propriété dans le cas de problème de maximisation :

**Définition:** Pour tout paysage adaptatif  $(\mathcal{S}, \mathcal{V}, f)$ , une solution  $s^*$  est un *optimum local* ssi le prédicat  $isOpt(s^*, f, \mathcal{V})$  est vrai où  $isOpt$  est défini par  $isOpt(s^*, V, g) = (\forall s \in V(s^*), g(s) \leq g(s^*))$ .

De plus, si  $\forall s \in \mathcal{S}, g(s) \leq g(s^*)$ , l'optimum est global.

Un paysage est dit *multimodal* suivant le nombre d'optima locaux, noté  $M$ . Palmer [97] définit un paysage fortement multimodal lorsque l'ordre de grandeur de  $M$  est exponentiellement croissant avec la "dimension" de l'espace de recherche <sup>2</sup>. Cependant cette taille  $M$  est le plus souvent accessible sans une énumération exhaustive de l'espace de recherche et peut-être estimé par des méthodes statistiques [45, 48]. Ces méthodes sont toutes basées sur la notion de marche adaptative qui converge nécessairement vers un optimum local. En effet, une *marche adaptative* est une suite de solutions voisines où la solution suivante est la solution voisine de meilleure performance.

**Définition:** Pour tout paysage adaptatif  $(\mathcal{S}, \mathcal{V}, f)$ , une *marche adaptative*  $(s_0, s_1, \dots, s_L)$  est une marche telle que  $\forall i \in \{0, \dots, L-1\}, f(s_i) < f(s_{i+1})$  et  $s_{i+1} = \underset{s \in \mathcal{V}(s_i)}{\operatorname{argmax}} f$  et telle que  $s_L$  est un optima local. La notation *Argmax* est l'argument maximum et représente la valeur de la variable pour laquelle la fonction atteint son maximum.

$L$  est alors la longueur de la marche adaptative. La définition d'une marche adaptative est ambiguë lorsqu'il existe plusieurs solutions dans le voisinage  $\mathcal{V}(s_i)$  de performance maximale. Une marche adaptative commençant par la même solution initiale peut aboutir à des optima locaux différents, voire même finir sur un plateau d'optima locaux et non plus sur un optimum local "isolé". Nous exposerons les modifications possibles de cette définition dans la partie sur les réseaux de neutralité. Dans la suite de ce paragraphe, afin de garantir la correction des définitions, nous supposerons qu'une marche adaptative commençant sur une même solution se termine toujours sur le même optimum, même dans le cas où le paysage contient beaucoup de solutions voisines de performance égale.

À chaque optimum local  $s^*$  est associé un bassin d'attraction  $\mathcal{B}(s^*)$ . Une solution  $s_0$  appartient à  $\mathcal{B}(s^*)$  si et seulement si la marche adaptative commençant par la solution  $s_0$  se termine sur l'optima local  $s^*$ . L'ensemble des bassins d'attractions forme alors une partition de l'espace de recherche. La distribution des tailles des bassins est un facteur important pour les performances des métaheuristiques [47]. Garnier *et al* [48] propose une méthode pour estimer les tailles des bassins d'attractions et le nombre d'optima locaux  $M$ . La méthode consiste à réaliser une marche adaptative à partir de  $N$  points choisis aléatoirement dans l'espace de recherche. On suppose que chaque marche converge vers un unique optimum local. Le résultat de ces  $N$  marches permet d'obtenir les nombres  $\beta_j$  pour  $1 \leq j \leq N$  correspondant au nombre d'optima locaux détectés à partir de  $j$  solutions. Ensuite, il s'agit de proposer une famille de densité de

---

<sup>2</sup>par exemple, pour le problème du voyage de commerce, la dimension du problème est le nombre de villes.



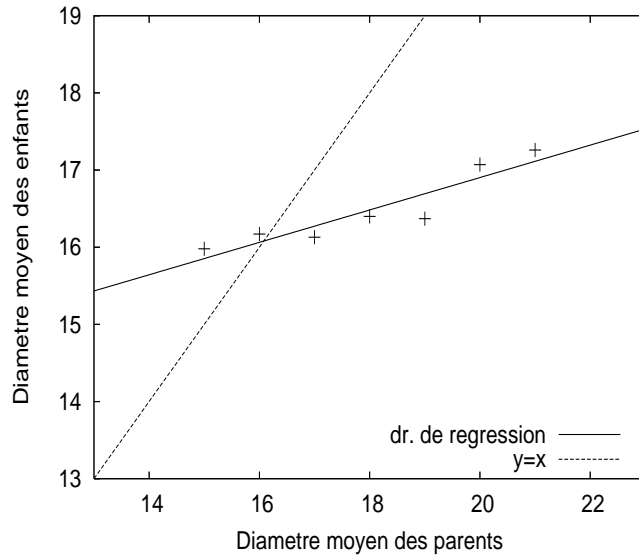


FIG. 1.2 – Données originales de Galton sur la taille des pois de senteur.

probabilité  $H_\gamma$ , paramétrée par un nombre réel  $\gamma$ , dont l’une de ces densités est la densité de probabilité d’avoir une taille donnée de bassin d’attraction. Ainsi, à partir de ces densités  $H_\gamma$ , il est possible de calculer le nombre moyen  $\beta_{j,\gamma}$  d’optima locaux détectés à partir de  $j$  solutions aléatoirement choisies dans l’espace de recherche. Enfin, un test du  $\chi^2$  qui consiste à calculer pour tout  $\gamma$ ,  $T_\gamma = \sum_{j>0} \frac{(\beta_j - \beta_{j,\gamma})^2}{\beta_{j,\gamma}}$  permet de connaître la distribution  $H_\gamma$  la plus probable; et en déduire une estimation du nombre d’optima locaux  $M$  et de la distribution des tailles des bassins d’attraction  $\mathcal{B}(s^*)$ .

### 1.3.2 Rugosité

La *rugosité* d’un paysage adaptatif est relative à sa structure locale plus ou moins accidentée. Lorsque le paysage est irrégulier, on dit qu’il est rugueux, dans le cas contraire, on le dit peu rugueux, continue ou lisse. La quantification de la rugosité s’effectue en mesurant la corrélation entre les valeurs de performance de solutions voisines.

### Corrélation Parents/Enfants et Évolutivité

L’étude de la relation entre les performances de solutions voisines n’est pas nouvelle, du moins dans son interprétation biologique. En effet, dès la fin du XIX siècle, Galton [43, 44] réalisa les premières études de l’hérédité des caractères quantitatifs en étudiant l’hérédité de la taille des pois de senteur. Il fit pousser sept lots différents de graines tel que dans chaque lot la taille des graines est homogène. Après la récolte, il mesura la taille des graines enfants obtenues. Il disposa sur un graphique la moyenne des tailles des pois de senteur enfants en fonction de la moyenne des tailles de leurs parents (voir figure 1.2), puis traça une droite, dite depuis *droite de régression*, pour décrire la relation entre la moyenne des tailles des enfants et la moyenne des tailles des parents. Galton utilise cette droite pour prédire l’évolution de la taille des pois au cours des générations. À chaque génération, la *droite de régression* permet d’estimer la taille à la génération suivante et ainsi de suite jusqu’au point “moyen”. Galton commet une erreur qui est relevée et corrigée par Pearson [98] : il considère dans son raisonnement qu’il n’y a pas évolution de l’hérédité des caractères quantitatifs, c’est-à-dire de la droite de corrélation.

La corrélation parents/enfant rejoint en optimisation la notion d'*évolvabilité*. Altenberg [2, 41, 3] définit l'évolvabilité comme "la capacité de l'opérateur de variation locale à produire des solutions de meilleure performance que les solutions initiales". L'auteur précise que l'évolvabilité est une mesure de performance au niveau local qui s'intéresse à la distribution de performance des solutions produites par un opérateur.

Comme souligné par Turney [131], la notion d'évolvabilité est difficile à définir. Il tente de définir intuitivement l'évolvabilité par :

si deux solutions  $s$  et  $s'$  sont de même performance,  $s$  est plus "évolvable" que  $s'$   
si la meilleure solution voisine de  $s$  est plus grande que celle de  $s'$ .

Nous choisirons une définition de l'évolvabilité plus neutre permettant de prendre en compte un grand nombre de situations et nous introduirons des mesures d'évolvabilité déduite de la définition de l'évolvabilité. Nous définissons donc l'évolvabilité d'une solution  $s$  relative à un opérateur local  $op$  comme la distribution de probabilité des performances des solutions obtenues par l'opérateur.

**Définition:** Soient un paysage adaptatif  $(\mathcal{S}, \mathcal{V}, f)$  et  $op : \mathcal{S} \rightarrow \mathcal{S}$  un opérateur<sup>3</sup> local agissant sur  $\mathcal{S}$  tel que pour tout  $s \in \mathcal{S}$ ,  $op(s) \in \mathcal{V}(s)$ . Notons  $Y : \mathcal{S} \rightarrow \mathbb{R}$  la v.a.  $Y(s) = f(op(s))$ . L'*évolvabilité* de  $s$  relative à l'opérateur  $op$  est la densité de probabilité de  $Y(s)$ .

Plusieurs mesures d'évolvabilité peuvent être naturellement déduites de cette définition : l'évolvabilité moyenne par  $E(Y(s))$ , la probabilité d'amélioration  $P(Y(s) \geq f(s))$  etc.

D'autres auteurs ont utilisé la corrélation de performance entre solutions voisines ; citons par exemple Manderick *et al* [83] et Greffenstette [51] qui définissent la "distribution de fitness d'un opérateur" afin de prédire l'efficacité d'un algorithme génétique. Fogel *et al* [37] utilisent la même idée pour déterminer, au cours de la recherche, l'opérateur le plus efficace. Bornholdt [14] analyse la distribution de performance des solutions voisines à l'aide de la technique des cumulants. Igel *et al* [61, 62] utilisent la probabilité d'amélioration et la performance moyenne des solutions voisines pour déterminer l'opérateur le plus adapté au problème d'optimisation, enfin Smith *et al* [119] utilisent plusieurs mesures de l'évolvabilité basées sur la distribution de performance et montrent que ces mesures sont capables de rendre compte de la difficulté due à la multimodalité et à la neutralité des problèmes.

## Autocorrélation

La rugosité d'un paysage peut-être aussi décrite à l'aide de la corrélation locale de performance où l'on cherche à connaître la corrélation de performance entre solutions proches. La *fonction d'autocorrélation* et la *longueur de corrélation*, introduites par Weinberger [141, 140] sont de loin les indicateurs de rugosité les plus accessibles au calcul numérique.

Étant donnée une marche aléatoire  $(s_t, s_{t+1}, \dots)$ , la fonction d'autocorrélation  $\rho$  d'une fonction de performance  $f$  est l'autocorrélation de la série temporelle  $(f(s_t), f(s_{t+1}), \dots)$  :

$$\rho(n) = \frac{E[f(s_t)f(s_{t+n})] - E[f(s_t)]E[f(s_{t+n})]}{var(f(s_t))}$$

où  $E[f(s_t)]$  et  $var(f(s_t))$  sont respectivement l'espérance et la variance de  $f(s_t)$ . Des estimés  $r(n)$  des coefficients d'autocorrélation  $\rho(n)$  peuvent être obtenus à l'aide d'une série temporelle  $(s_1, s_2, \dots, s_L)$  de longueur  $L$  :

$$r(n) = \frac{\sum_{t=1}^{L-n} (f(s_t) - \bar{f})(f(s_{t+n}) - \bar{f})}{\sum_{t=1}^L (f(s_t) - \bar{f})^2}$$

---

<sup>3</sup>cet opérateur peut-être stochastique

où  $\bar{f} = \frac{1}{T} \sum_{t=1}^L f(s_t)$ , et  $L \gg 0$ .

Une marche aléatoire est représentative de l'ensemble du paysage lorsque le paysage est statistiquement isotropique. Dans ce cas, quelque soit le point de départ de la marche aléatoire et le voisin sélectionné durant cette marche, l'estimé de  $r(n)$  converge vers la valeur  $\rho(s)$  [126]. L'erreur d'estimation diminue avec la longueur de la marche.

La longueur de corrélation  $\tau$  [38, 113] mesure la décroissance de la fonction d'autocorrélation et la rugosité du paysage :  $\tau = \sum_{s=0}^{\infty} \rho(s)$ . Le paysage est d'autant plus lisse que la longueur de corrélation est grande.

Cette définition de la longueur de corrélation est pertinente lorsque la décroissance de la fonction d'autocorrélation est exponentielle. En effet, dans ce cas, la fonction d'autocorrélation s'écrit  $\rho(s) = e^{-\frac{s}{\tau}}$ . Il est possible d'utiliser une définition plus générale, proposée par Hordijk [58], venant de l'analyse des séries temporelles par la méthode de Box et Jenkins [15]. Dans cette méthode, la série temporelle des valeurs d'adaptation est approchée par un *modèle autorégressif à moyenne mobile*, le modèle ARMA. Dans un modèle ARMA( $p, q$ ), la valeur courante dépend linéairement des  $p$  valeurs précédentes et des  $q$  bruits blancs précédents :

$$f(s_t) = c + \sum_{i=1}^p \alpha_i f(s_{t-i}) + \epsilon_t + \sum_{i=1}^q \beta_i \epsilon_{t-i}$$

où  $\epsilon_t$  sont des bruits blancs.

L'approche consiste à itérer trois étapes [15]. L'*étape d'identification* détermine les valeurs de  $p$  et  $q$  en utilisant la fonction d'autocorrélation (ACF) et la fonction d'autocorrélation partielle (PACF) de la série temporelle. L'*étape d'estimation* calcule les valeurs  $c$ ,  $\alpha_i$  et  $\beta_i$  en utilisant la PACF. La validité de ces valeurs est déterminée à l'aide d'un t-test : elles ne sont pas significatives lorsque le t-test donne une valeur inférieure à 2. L'*étape de diagnostique* est composée de deux parties. La première vérifie l'adéquation entre les données réelles et les données produites par le modèle par la mesure de corrélation  $R^2$  et par le *critère d'information d'Akaike (AIC)* :

$$AIC(p, q) = \log(\hat{\sigma}^2) + 2(p + q)/L$$

où  $\hat{\sigma}^2 = L^{-1} \sum_{j=1}^L (y_j - \hat{y}_j)^2$

La seconde partie vérifie l'hypothèse de bruit blanc des résidus qui est la différence entre les données observées et les valeurs estimées. Pour cela, l'autocorrélation des résidus et les p-valeurs du test Ljung-Box sont calculées.

La longueur de corrélation  $\tau$  est alors définie par :  $\forall i \geq \tau, |r(\tau)| \leq 2/\sqrt{T}$  et  $|r(\tau)| > 2/\sqrt{T}$ .

Les notions d'optima locaux et de rugosité sont liées. Lorsque la longueur d'autocorrélation est faible, il y a peu de corrélation entre solutions voisines et il existe de nombreux optima locaux ; Les métaheuristiques basées sur le voisinage seraient peu efficace. À l'inverse, lorsque la longueur d'autocorrélation est grande, il existe peu d'optima locaux. Les cas intermédiaires avec une longueur intermédiaire sont les plus fréquents. Dans ce cas, les coefficients  $\alpha_i$  et  $\beta_i$  renseignent sur la nature du paysage. Par exemple, Hordijk [56] émet l'hypothèse que pour un paysage adaptatif qui s'approche par un modèle AR(2), un voisinage de taille deux est nécessaire pour avoir des informations pertinentes et utiles sur le paysage. Stadler et Garcia ont formulé une conjecture [127, 45] qui permet de relier le nombre d'optima locaux et la longueur de corrélation :

$$M \approx |\mathcal{S}|/|\mathcal{S}(x_0, \tau)|$$

où  $\mathcal{S}(x_0, \tau)$  est l'ensemble des solutions de  $\mathcal{S}$  que l'on peut atteindre au cours d'une marche de longueur  $\tau$  sur le graphe  $\Gamma = (\mathcal{S}, \mathcal{V})$ . Cette relation est vérifiée sur les paysages classiques d'optimisation combinatoire tels que le problème du voyageur de commerce.

## 1.4 Neutralité

Alors que la géométrie multimodale des paysages est liée à la présence d’optima locaux, nous allons présenter dans cette section une autre géométrie liée à la présence de plateau. La géométrie des *paysages adaptatifs neutres* a été mise en avant par la théorie de la neutralité en évolution moléculaire de Motoo Kimura [73]. Dans le domaine de l’optimisation combinatoire, nous montrerons que ce type de paysage existe. Les espaces de recherche sont alors redondants, un grand nombre de solutions sont équivalentes et ont la même performance. Les paysages adaptatifs contiennent des plateaux lorsque le nombre de solutions voisines de performance égale est important. La dynamique des métaheuristiques sur les paysages adaptatifs neutres est spécifique, elle est appelée dynamique des équilibres ponctués. L’évolution est une alternance de phases de dérive aléatoire sur les plateaux et de phases de changement rapide de performance. Ensuite, nous verrons comment les métaheuristiques peuvent exploiter les caractéristiques de cette géométrie afin d’optimiser plus efficacement les problèmes. Enfin, nous exposerons les principales méthodes d’introduction avantageuse de la neutralité dans un paysage, basé sur l’espoir de remplacer un paysage multimodale difficile à optimiser par un paysage avec des plateaux qui permettent d’éviter les optima locaux.

### 1.4.1 Origine du concept

La théorie neutraliste est une théorie alternative à la théorie synthétique de l’évolution proposée par le biologiste M. Kimura [72, 73]. La théorie synthétique n’invoque que la sélection positive, *i.e.* la sélection d’un gène ayant un avantage sélectif, pour expliquer les différences biologiques entre espèces, voire même les différences les plus infimes [36]. Selon cette théorie, le taux et la direction de l’évolution sont surtout déterminés par sélection positive, les mutations ne jouant seulement qu’un rôle mineur. On pensait que même si les mutations cessaient complètement, la variabilité génétique entraînée par recombinaison serait en générale suffisante. Selon Fisher [35], les taux d’évolution sont avant tout conditionnés par les facteurs sélectifs plutôt que par les taux de mutation.

La théorie neutraliste que Kimura appelée aussi théorie de “la mutation et de la dérive aléatoire” s’appuie principalement sur deux disciplines. D’une part, la génétique moléculaire a permis de mesurer les taux d’évolution des substitutions des acides animés à partir de la comparaison des hémoglobines des vertébrés et ainsi d’estimer le taux de substitution des nucléotides au sein des gènes. Elle a aussi permis de mesurer la variabilité génétique intra-spécifique. D’autre part, la théorie stochastique (Fisher, Wright, Kimura), en utilisant les équations de diffusion, a permis de traiter le comportement des allèles mutants en tenant compte des changements aléatoires dus à l’échantillonnage au hasard des gamètes lors de la reproduction. Cela a permis, par exemple, de déterminer la probabilité de fixation d’un gène dans une population finie. Selon la théorie neutraliste, les changements évolutifs ainsi que la majeure partie de la variation intra-spécifique résultent principalement de la fixation aléatoire de mutants sélectivement neutres ou presque neutres et non pas d’une sélection darwinienne positive. Elle suppose que l’intensité de la pression de sélection est si faible que la pression de mutation et la dérive aléatoire [74] sont les facteurs essentiels de l’évolution moléculaire.

Bien que les paysages adaptatifs issus de problème d’optimisation ne soient certainement pas de même nature que les paysages adaptatifs “biologiques”, la théorie de la neutralité met en relief des caractéristiques peu ou pas étudiées auparavant. Dans le domaine de l’optimisation par algorithme génétique, la dérive aléatoire a fait l’objet d’études depuis la fin des années 1980 [50]. Une représentation “classique” de paysage par plateaux, dont un exemple est donné par la figure 1.3, y est associée. Le concept de neutralité dans les paysages adaptatifs avec les

principales définitions<sup>4</sup> a pour l'essentiel été formulé par Schuster dans ses travaux en évolution moléculaire [112, 105, 111] sur l'étude de la structure secondaire de l'ARN. Il définit la mesure du nombre de solutions voisines de même performance, appelé degré de neutralité, et étudie les caractéristiques des marches neutres (une *marche neutre* est une suite de solutions voisines de même performance et où la distance à la solution initiale est strictement croissante). Enfin, Schuster a défini la notion de *réseau de neutralité*, *i.e.* le graphe des solutions voisines de même performance, de la même manière que Maynard [84] avait défini les réseaux protéiques. Toutes ces notions et définitions ont permis de mettre en évidence la présence de neutralité dans les paysages adaptatifs d'optimisation ; quelques exemples sont donnés dans la suite de cette section. Il n'existe pas de définition précise de paysage adaptatif neutre, ni de critère quantitatif permettant de discriminer clairement les paysages contenant de la "neutralité" des autres. Dans toute la suite ce travail, nous appellerons *paysage adaptatif neutre* un paysage contenant un grand nombre de solutions dont le degré de neutralité est important.

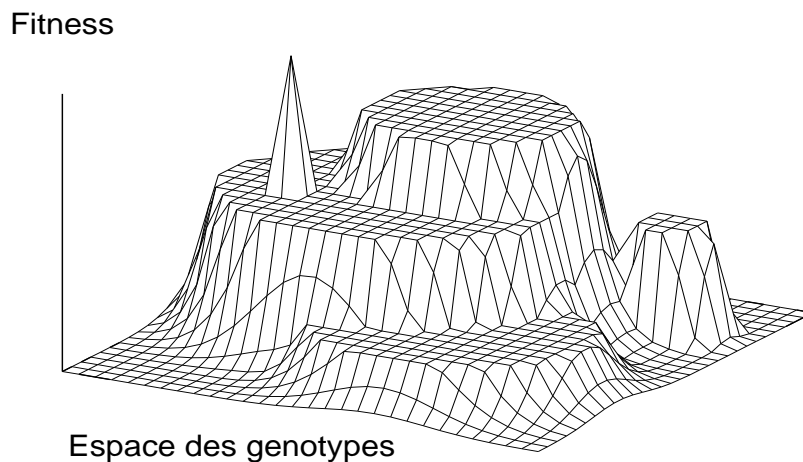


FIG. 1.3 – Représentation classique d'un paysage adaptatif neutre de dimension deux.

### 1.4.2 Neutralité dans les problèmes réels

Parmi les problèmes d'optimisation que l'on rencontre dans les applications réelles, certains correspondent à des paysages adaptatifs neutres. Nous en présentons ici quelques exemples.

#### Erreur d'évaluation et contrôleurs de robot

Husbands [59] propose d'utiliser un réseau de neurones particulier (GasNet) pour contrôler un robot. L'optimisation du réseau de neurones est réalisée à l'aide d'un algorithme génétique. L'ensemble des neurones est représenté par une chaîne d'entiers de longueur variable où chaque neurone est codé par dix-neuf paramètres contrôlant le poids d'une connexion, l'entrée sensoriel,

<sup>4</sup>ces définitions seront exposées en détail dans la partie 3

etc. Les mutations sont de trois types : insertion ou suppression de neurone et changement de valeur de paramètre. Smith *et al* [120] [121] [122] soulignent l'importance de la neutralité dans ce paysage adaptatif. En effet, les robots doivent résoudre des tâches dans des environnements bruités où nécessairement la performance ne peut être calculée précisément. C'est ce bruit d'évaluation qui induit de la neutralité dans le paysage. La dynamique d'évolution est alors influencée par la neutralité [120] : la dynamique alterne des phases où la performance moyenne de la population stagne, avec des phases où cette performance croît rapidement. Cette dynamique particulière nommée *équilibre ponctué* sera présentée en section 4.1. Durant la première phase, la population se déplace significativement sur les réseaux de neutralité et la probabilité de découvrir, dans le voisinage des solutions de la population, une solution de meilleure performance est constante.

Des travaux similaires en évolution de réseaux de neurones (CTRNN) contrôlant la marche d'un robot unijambiste mettent également en évidence la présence de neutralité [115]. Seys et Beer observent les deux mêmes phases caractéristiques, une phase d'exploration d'un réseau de neutralité et une phase de découverte de meilleures solutions. Les auteurs mesurent le temps passé sur un réseau de neutralité ainsi que les mouvements de la population entre réseaux. Ils en déduisent que la recherche peut être assimilée à une recherche aléatoire durant la phase d'exploration. Cette étude leur a permis d'adapter l'opérateur de mutation selon la phase de la recherche. L'erreur d'évaluation dans ces problèmes d'apprentissage des réseaux de neurones contrôlant un robot conduit à une neutralité inhérente au paysage adaptatif.

## Neutralité et programmation génétique

La programmation génétique (PG) [76] est la classe des algorithmes évolutionnaires adaptée au traitement de solutions exprimées sous la forme de programmes (arbre, pile, ...). Ebner [32] met en évidence que le nombre de programmes qui codent la même fonction est important. En particulier, l'existence de code mort favorisée par la taille variable des programmes est une source de redondance où des programmes différents produisent un même résultat et donc ont la même performance. La redondance est une condition nécessaire à l'existence de neutralité dans le paysage. Plus la redondance est importante, plus la probabilité que des solutions voisines aient la même performance est grande. Par nature, donc, les paysages en programmation génétique sont neutres.

La PG cartésienne permet de concevoir des circuits électroniques [86]. Dans cette variante de la PG, on configure un tableau de portes logiques. Chaque porte possède le même nombre d'entrées, une sortie et une fonction. On peut ainsi coder l'ensemble des portes par une chaîne de taille fixe qui représente le graphe indexé des portes. L'étude sur plusieurs problèmes montrent la présence et l'avantage de la neutralité. Pour la conception d'un multiplieur à 3-bit, Vassilev [136] mesure la taille des réseaux de neutralité et montre que le nombre de mutations neutres décroît moins vite que la taille des réseaux. Les mutations neutres sont donc utilisées lors de la recherche. La neutralité permet d'échapper aux optima locaux en évitant les mutations délétères. Pour la conception d'un circuit réalisant la fonction parité<sup>5</sup>, Yu *et al* [148] [149] mettent en évidence la présence de neutralité et montrent qu'en favorisant les mutations neutres, un AE améliore ses performances.

Les "Field Programmable Gate Array" (FPGA) sont des circuits électroniques reconfigurables. Ces circuits sont des matrices de portes logiques où chaque porte peut être configurée. Il est possible d'utiliser un AE afin de programmer de tels circuits [129] [54]. Thompson et Harvey ont mené une étude sur un algorithme génétique dont la tâche est de programmer un FPGA qui résout le problème de distinction de signaux de fréquences différentes. Sans toutefois en expli-

---

<sup>5</sup>Parité du nombre de 1 dans une chaîne binaire

quer l’origine, ils ont montré la présence de neutralité dans le paysage adaptatif du problème. En effet, bien que la performance du meilleur individu n’augmente plus, la population se déplace encore dans le paysage montrant ainsi qu’elle n’a pas convergé “autour” d’un optimum local.

Dans le cas de la programmation génétique, les codages sont fortement redondants et conduisent à des paysages neutres, et cela ne semble pas être un obstacle pour obtenir de bonnes performances.

### 1.4.3 Paysages adaptatifs neutres académiques

En évolution artificielle de nombreux problèmes académiques ont été proposés afin d’étudier les relations entre leurs propriétés caractéristiques et les performances des AE ; parmi ces problèmes tests, certains définissent des paysages d’adaptation qui ont une géométrie neutre.

#### Paysage contenant de la neutralité implicite

Les paysages Routes Royales (Royal-Road) [88] dépendent de deux paramètres : le nombre de blocs  $n$  et la taille  $k$  d’un bloc. Ils ont été initialement conçus pour étudier la dynamique des algorithmes génétiques. Ils permettent de décrire comment des parties de la solution optimale, appelés blocs de construction, sont combinés pour produire de meilleures solutions. Toutefois, ces paysages sont fortement neutres et le degré de neutralité peut être ajusté à l’aide du paramètre  $k$ .

Plus formellement, la fonction  $B_i$  indique si tous les bits du bloc  $i$  valent 1 :  $B_i(x) = 1$  si  $\forall j \in [ik, (i+1)k - 1], x_j = 1$  et  $B_i(x) = 0$  sinon. La fonction de fitness  $f_{RR}$  est définie sur les chaînes binaires de longueur  $nk$  par :  $\forall x \in \{0, 1\}^{nk}, f_{RR}(x) = \sum_{i=0}^{K-1} B_i(x)$ . La fitness d’une solution est modifiée soit parce que l’on mute un seul bit dans un bloc contenant  $k$  bits positionnés à 1, soit parce que l’on mute l’unique bit 0 d’un bloc. Si bien que plus  $k$  est grand, plus le degré de neutralité des solutions est important.

Le problème MAX-SAT est le problème d’optimisation relatif au problème SAT. Dans ce problème, l’espace de recherche  $\mathcal{S}$  est l’ensemble des chaînes binaires de longueur  $N$  qui représente la valeur de chaque variable  $x_i$  du problème. MAX-SAT est défini à partir de  $m$  clauses  $C_j$  pour  $j \in \{1, \dots, m\}$ . Chaque clause est une disjonction de littéraux où chaque littéral est une variable positive booléenne  $x_i$ , soit la négation d’une variable booléenne  $\neg x_i$ . Sans perdre en généralité, nous supposons que chaque clause contient au plus la variable  $x_i$  ou  $\neg x_i$ . Une clause est satisfaite lorsque la valeur *vrai* est affectée à au moins un littéral positif ou la valeur *faux* est affectée à au moins la négation d’un littéral. Le problème consiste à maximiser le nombre de clauses satisfaites. Ce problème est NP-complet [46]. Beaucoup d’études portent sur le problème SAT [85]. Il a été montré qu’il existe une transition de phase entre les problèmes solubles et insolubles en fonction du ratio  $\alpha = \frac{m}{N}$ . Sous le seuil  $\alpha_c = 4.3$ , une instance du problème admet de nombreuses solutions qui forment un grand cluster. Proche de la valeur critique  $\alpha_c$ , L’ensemble des solutions se sépare en de petits sous-ensembles et le problème SAT devient difficile à résoudre. Pour les valeurs de  $\alpha$  largement au dessus du seuil  $\alpha_c$ , il existe peu ou pas de solutions.

Les automates cellulaires sont des systèmes dynamiques discrets en espace et en temps. Ils consistent à évoluer selon une règle locale les états d’un ensemble de cellules disposés selon une géométrie régulière (appelé configuration). Par exemple, le problème de majorité est un problème de recherche de la meilleure règle, d’un automate cellulaire linéaire dont les états sont binaires, qui réalise la classification de la configuration initiale suivant la densité de l’état 1. L’analyse du paysage adaptatif massivement neutre relatif à ce problème est développée dans la section 4.3.1.

## Paysages adaptatifs de neutralité explicite

Nous allons présenter trois paysages où la neutralité est introduite de façon explicite et donc directement ajustable : les paysages NKq, les paysages NKp et les paysages technologiques. Ils sont tous trois définis à partir de la famille des paysages adaptatifs NK que nous allons préalablement présenter. Dans les paysages  $NK_q$  et  $NK_p$ , certains changements de valeur d’une variable ne modifient pas la performance des solutions. Dans les paysages technologiques, les faibles variations de la performance calculée à l’aide du paysage NK ne sont pas pris en compte.

Kauffman [69] a proposé la famille des paysages adaptatifs NK, paramétrée par des entiers  $N$  et  $K$ , afin d’explorer le lien entre l’épistasie, qui est le degré d’interaction entre les variables, et la multimodalité des paysages. L’espace de recherche est l’ensemble  $\mathcal{S} = \{0, 1\}^N$  des chaînes binaires de longueur  $N$  et le voisinage  $\mathcal{V}$  est le voisinage de Hamming de taille 1. Le paramètre  $K$  représente le nombre de liens “épistatiques” entre bits. En ajustant  $K$ , des paysages de différents degrés de rugosité peuvent être générés. Lorsque  $K = 0$ , le problème est sans épistasie, le paysage est lisse ; lorsque  $K = N - 1$ , le problème est très épistatique, le paysage est très rugueux.

Chaque bit  $i$  d’une solution apporte une contribution à la performance globale par une fonction  $f_i : \{0, 1\}^{K+1} \rightarrow [0, 1)$ . Cette contribution dépend localement du bit  $i$  mais aussi de  $K$  autres bits épistatiquement liés ( $K$  est compris entre 0 et  $N - 1$ ). La fonction d’adaptation  $f_{NK}(s)$  d’une solution  $s \in \mathcal{S}$  est la moyenne des valeurs des  $N$  fonctions de contribution  $f_i$  :

$$f_{NK}(s) = \frac{1}{N} \sum_{i=1}^N f_i(s_i; s_{i_1}, \dots, s_{i_K})$$

où  $\{i_1, \dots, i_K\} \subset \{1, \dots, i - 1, i + 1, \dots, N\}$ . Deux solutions ont été proposées pour choisir les  $K$  bits épistatiques parmi  $N$  : le choix *adjacent* ( $K$  bits dont les positions suivent celle du bit  $i$ <sup>6</sup>), ou le choix aléatoire. Chaque contribution  $f_i$  est définie en extension, par un nombre réel  $y_{i,(x_i;x_{i_1},\dots,x_{i_K})}$  de  $[0, 1)$  associé à chaque élément  $(x_i; x_{i_1}, \dots, x_{i_K})$  de  $\{0, 1\}^{K+1}$ . Ces nombres sont choisis uniformément dans l’intervalle  $[0, 1)$ .

À partir de ce modèle, trois variantes ont été proposées pour introduire de la neutralité. On peut diminuer le nombre de contributions participant à la valeur adaptative, discrétiser les valeurs des contributions ou encore discrétiser les valeurs d’adaptation.

Les *paysages adaptatifs NKp* ont été introduit par Barnett [7] ; ils consistent à particulariser les fonctions de contribution des paysages NK, à l’aide du paramètre  $p \in [0, 1]$  ajustant la neutralité du paysage. La valeur d’une contribution est nulle avec une probabilité  $p$ , *i.e.*  $P(y_{i,(x_i;x_{i_1},\dots,x_{i_K})} = 0) = p$ . La probabilité que deux solutions voisines aient la même performance augmente avec le paramètre  $p$ . On peut trouver les principaux résultats sur les paysages NKp au sixième chapitre de la thèse de L. Barnett [8]. Cette famille de paysages modélise les paysages où il existe des combinaisons d’interactions qui ne modifient pas la performance.

Les *paysages adaptatifs NKq* ont été introduit par Newman *et al* [91]. Comme dans les paysages NKp, les fonctions de contribution sont définies à l’aide d’un paramètre entier  $q$  supérieur ou égale à 2. Chaque  $y_{i,(x_i;x_{i_1},\dots,x_{i_K})}$  est l’une des fractions  $\frac{k}{q}$  où  $k$  un nombre entier choisi uniformément dans l’intervalle  $[0, q - 1]$ . La neutralité du paysage est maximale lorsque  $q$  est égale à 2, et décroît lorsque  $q$  augmente. Les auteurs montrent que cette famille de paysages permet de modéliser les propriétés d’évolution neutre d’espèces moléculaires.

Les *paysages Technologiques* ont été introduit par Lobo *et al* [81]. Ils se définissent en discrétisant en  $M$  valeurs les performances possibles de la fonction  $f_{NK}$ , soit  $f_{techn}(x) = \frac{k}{M}$  ssi  $k \leq M \cdot f_{NK}(x) < k + 1$ . Selon les auteurs, cette famille de paysages modélise l’évolution de firmes technologiques où l’évolution de la performance d’une firme n’est pas graduelle mais s’effectue par “saut technologique”.

---

<sup>6</sup>Le bit suivant du bit  $N$  est le bit 1



### 1.4.4 Influence sur la conception de métaheuristiques

La mise en évidence de paysages adaptatifs neutres dans les problèmes d'optimisation et une meilleure représentation et compréhension de leurs structures ont permis récemment de concevoir des métaheuristiques adaptées.

#### Nageur de réseau

Barnett [9] a montré qu'en présence d'un haut degré de neutralité, il peut être plus avantageux d'utiliser un seul individu "nageant" sur les réseaux de neutralité plutôt qu'une population de solutions potentielles. La métaheuristique dite du *Nageur de réseau*<sup>7</sup> (NR) s'apparente donc à un recuit simulé dont la température serait nulle. L'intérêt de la démarche vient du fait que Barnett définit une classe de paysages adaptatifs neutres sur lesquels le NR est l'algorithme évolutionnaire le plus performant. Cette classe des paysages  $\epsilon$ -corrélé regroupe une large part des paysages neutres dont la famille des paysages académiques *NKp* (voir section 1.4.3) fait parti. Un paysage est dit  $\epsilon$ -corrélé s'il vérifie les hypothèses suivantes :

1. la probabilité de mutations avantageuses est très faible comparée à celles des mutations neutres ou délétères.
2. les seules mutations avantageuses de probabilité non nulle sont celles qui conduisent au réseau de neutralité adjacent.

On suppose par ailleurs que le degré de neutralité (i.e. la probabilité qu'une mutation soit neutre) est constant dans un même réseau de neutralité. Le NR a donc été conçu pour dériver dans un paysage adaptatif neutre  $\epsilon$ -corrélé ; lors du processus de recherche, la solution courante est mutée et le remplacement a lieu uniquement quand la performance du mutant est égale ou supérieure à la performance courante. Le NR réalise ainsi un parcours aléatoire neutre sur le paysage, ponctué par des transitions vers des réseaux de meilleures qualités (algorithme 1). Barnett fournit une estimation quantitative du temps nécessaire pour découvrir de meilleures solutions et propose une variante adaptative du nageur de réseau où l'on ajuste dynamiquement le nombre de bits mutés en fonction d'informations statistiques sur le paysage collectées durant l'exécution.

---

**Algorithme 1** Nageur de réseau. StepMax est le nombre maximal d'itération de l'algorithme.

---

**Require:** stepMax > 0

step  $\leftarrow$  0

Choisir une solution initiale  $s \in \mathcal{S}$

**répéter**

  choisir  $s' \in \mathcal{V}(s)$  uniformément

**si**  $f(s) \leq f(s')$  **alors**

$s \leftarrow s'$

**fin si**

  step  $\leftarrow$  step + 1

**jusqu'à** stepMax  $\leq$  step

---

#### Extrema Sélection

On considère ici que l'on utilise un algorithme évolutionnaire (AE) pour rechercher de bonnes solutions dans un espace de recherche neutre ; dans ce contexte, la "Sélection Extrême" proposée

---

<sup>7</sup>Netcrawler en anglais

par Terry Stewart [128] a pour objectif d’accélérer l’évolution lors de la recherche. Quand la majorité des solutions dans une population a atteint le même niveau de performance on observe généralement une dérive génétique purement aléatoire ; cela correspond à une longue période durant laquelle il n’y a plus d’amélioration de la qualité des solutions trouvées par l’AE. L’idée est d’attribuer à chaque solution une performance endogène lors de la sélection qui aurait pour effet de disperser la population sur le réseau de neutralité. On peut ainsi espérer, par une exploration plus large du réseau, atteindre plus rapidement des points de meilleure performance. La performance endogène d’une solution prend en compte la distance au centroïde de la population. Pour s’assurer que la population reste dans un même réseau de neutralité, les mutations délétères sont ignorées. Ce type particulier de sélection est évalué sur la famille des paysages  $NKp$  pour lesquels le paramètre  $p$  permet de contrôler la neutralité. Les expériences menées montrent que l’amélioration des performances due à la "sélection extrême", comparée à une sélection standard (roulette-wheel), est corrélée au degré de neutralité.

#### 1.4.5 Pourquoi exploiter la neutralité ?

De façon générale, une métaheuristique de recherche locale ne peut être performante que si l’on choisit avec soin le codage du problème et les opérateurs. Ces choix permettent d’introduire une connaissance spécifique au problème. Des codages et des opérateurs différents engendrent des espaces de recherche de tailles variées et des structures de paysage adaptatif plus ou moins rugueux, ou plus ou moins neutre. Dès lors on peut se demander si un codage augmentant la neutralité permet une optimisation plus efficace. Les mesures statistiques sur les paysages adaptatifs permettent de répondre en partie à cette question.

Nimwegen et Crutchfield [93] ont utilisé la notion de *barrière* pour éclairer cette question. Une dynamique de recherche couramment rencontrée en évolution biologique ou optimisation artificielle, comme nous avons pu voir dans les problèmes précédents, est la dynamique des *équilibres ponctués* [34]. Celle-ci consiste en l’alternance de longues périodes de stabilité sur une valeur adaptative avec des périodes de transition rapide vers une meilleure valeur adaptative. Deux interprétations sont possibles pour expliquer cette dynamique. Wright [147] dans sa théorie "shifting balance" propose que la population reste bloquée autour d’un optimum local durant la première période jusqu’à ce que de rares mutants traversent une "vallée" vers un meilleur optimum local. On doit alors franchir une *barrière de performance* et accepter de dégrader temporairement la performance. Une autre interprétation inspirée par la théorie de la neutralité propose que la population se diffuse sur un réseau de neutralité jusqu’à trouver une solution "*porte*" qui permette d’accéder à une solution voisine de meilleure performance. Lors de la diffusion, la population n’est guidée par aucune information liée à la performance. Les auteurs Nimwegen et Crutchfield comparent la phase de diffusion au franchissement d’une *barrière d’entropie*. La question est donc de savoir s’il est plus facile de franchir une barrière de performance plutôt qu’une barrière d’entropie.

Les travaux d’Izquierdo-Torres [64] illustrent cette dualité. Comme dans le cas de l’apprentissage de la marche (section 1.4.2), il s’agit de déterminer les paramètres d’un réseau de neurones dynamique (CTRNN) afin de réaliser des portes logiques. La performance est mesurée par la différence entre les activations finales et attendues<sup>8</sup>. Un algorithme évolutionnaire (1 + 1), équivalent au Nageur de Réseaux (voir la section 1.4.4) est utilisé. À chaque itération, une solution voisine est acceptée si sa performance est supérieure ou bien dégradée de seulement  $t$  pour cent. Le paramètre  $t$  permet d’ajuster la neutralité du paysage. Plus  $t$  est grand, plus les ensembles de neutralité sont grands, plus on augmente les barrières d’entropie. Les meilleures performances de l’algorithme sont obtenues pour  $t^*$  égal à 5 pour cent. Avant  $t^*$ , l’algorithme

---

<sup>8</sup>ces activations sont des nombres réels

n’arrive pas à atteindre de bonnes valeurs performances, après  $t^*$  la recherche devient aléatoire et se détériore. Dans ce problème, l’introduction de neutralité permet de “gommer” des barrières de performance. Toutefois, lorsque la barrière d’entropie créée devient trop importante, l’algorithme n’est plus capable de converger vers les bonnes solutions.

Nous allons exposer dans la suite les moyens d’introduire explicitement de la neutralité dans un paysage adaptatif.

### 1.4.6 Neutralité synthétique

Étant donné un espace de recherche  $S$ , une méthode naturelle pour créer de la neutralité est de définir un nouvel espace de recherche  $S'$  de taille supérieure à  $S$  et une application surjective  $\varphi : S' \rightarrow S$ . Ainsi le nouvel espace  $S'$  est dit redondant puisque plusieurs solutions de  $S'$  codent une même solution de  $S$  et donc ont la même valeur d’adaptation. Plus précisément, la surjection  $\varphi$  induit une partition sur  $S'$  par la relation d’équivalence  $\mathcal{R} : s'_1 \mathcal{R} s'_2$  ssi  $\varphi(s'_1) = \varphi(s'_2)$ . Toutes les solutions d’une même classe d’équivalence ont les mêmes performances. Dans la suite nous allons distinguer trois types de redondance selon la structure des classes d’équivalence sur l’espace  $S'$  : la redondance *brute*, la redondance *plate* et la redondance *encapsulée*.

**Redondance brute** Dans le cadre des algorithmes génétiques où les espaces de recherche sont les chaînes binaires de longueur  $\lambda$ , plusieurs types d’application surjective, appelée mapping ont été proposés [116] [33]. Avec le *mapping d’automate cellulaire*,  $S'$  est définie par l’adjonction de  $\lambda$  tables de règles à chaque chaîne de longueur  $\lambda$  afin de définir un automate cellulaire non-uniforme. Les éléments  $S$  sont déterminés par l’itération de l’automate cellulaire. Le *random boolean mapping* diffère du précédent par le voisinage des cellules de l’automate cellulaire. Au lieu d’être classiquement celui de cellules contiguës, le voisinage est aléatoire parmi l’ensemble des cellules. Les auteurs de [116] [33] argumentent en faveur de ces codages en montrant qu’ils augmentent la connectivité entre les valeurs de performance, ainsi que le nombre de génotypes  $S$  rencontrés au cours d’une marche aléatoire sur  $S'$  ou le taux d’innovation au cours d’une marche neutre<sup>9</sup>. Malheureusement, ces codages semblent trop aléatoires pour être performants sur tous types de problème. Knowles [75] reprend le random boolean mapping et montre que sur les problèmes NK, Max-SAT et H-IFF le codage direct est plus performant.

Un codage redondant aléatoire n’a pas d’influence *a priori* sur les mesures caractéristiques du paysage (taux innovation, nombre moyen de portes d’un réseau, etc.) et ne peut pas améliorer les performances d’un AE. Introduire de la neutralité de façon “aveugle” doit donc être évité.

**Redondance plate** Afin d’évaluer la qualité de la neutralité, Rothlauf *et al* [109] introduisent le concept de redondance synonymique. La redondance est synonymique lorsque pour toute classe d’équivalence  $\mathcal{C}$  par la relation  $\mathcal{R}$ , les éléments de  $\mathcal{C}$  sont proches. Cette redondance est régulière au sens où elle “duplique” localement les solutions et tend à augmenter la taille des réseaux; elle ne modifie pas la dynamique d’un AG bien que la taille de la population et le nombre de générations doivent être augmentés.

Le *mapping par vote de majorité* est un exemple de redondance synonymique. Ici,  $S'$  est l’ensemble des chaînes binaires  $\{0, 1\}^{(2p+1)\lambda}$ . Le nombre de bits positionnés à 1 entre les positions  $(2p+1)k$  et  $(2p+1)(k+1) - 1$  d’une chaîne de  $s' \in S'$  détermine la valeur du bit  $k$  de la chaîne  $s \in S$ . Si ce nombre est supérieur ou égal à  $p$ , alors il y a une majorité de 1, la valeur du bit de  $s$  vaut 1, sinon le bit vaut 0.

---

<sup>9</sup>Toutefois, il n’a pas été comparé avec celui d’une marche aléatoire sur  $S$  ce qui ne permet pas de connaître l’innovation réelle d’un réseau de neutralité

**Redondance encapsulée** Dans cette section, nous allons présenter trois exemples où la redondance repose à la fois sur une surjection  $\varphi$  et sur des opérateurs spécifiques agissant sur l'ensemble  $S'$ . Ces opérateurs exploitent la redondance introduite et tiennent compte des caractéristiques du problème. Ainsi, ils permettent d'améliorer les performances d'un algorithme évolutionnaire. Cette redondance est appelée *encapsulée* par analogie au concept d'encapsulation en informatique des structures de données et des méthodes de traitement associées.

Dans le cadre de l'optimisation de problèmes dynamiques [49], les AE sont confrontés à deux difficultés. Dans un problème dynamique de bonnes solutions à un instant donné ne sont, en général, plus adaptées à une nouvelle situation. l'algorithme doit alors préserver la diversité de la population afin de conserver la capacité de créer de nouvelles solutions. Par ailleurs, il est fréquent que des situations proches de celles déjà réalisées surviennent de nouveau, aussi l'algorithme doit être capable de mémoriser des solutions dans le but de les réutiliser. Dans ce contexte, Levenick [80] étudie l'avantage d'ajouter dans le génotype des introns, c'est-à-dire des régions non codantes. Ces régions doublent au minimum la longueur de la chaîne. Lorsque la performance d'un individu devient négligeable, un opérateur de swapping effectue une permutation entre bits non codants et bits codants. Ce mécanisme préserve la diversité de la population et la partie non codante constitue un réservoir de diversité qui peut jouer le rôle de mémoire.

Les stratégies d'évolution sont des AE utilisés pour optimiser les problèmes codés à l'aide de vecteur de nombres réels. Ces nombres sont mutés en ajoutant une réalisation d'une variable aléatoire normalement distribuée de moyenne nulle et dont l'écart-type est un paramètre de la recherche. Dans une version simple de ces algorithmes, l'écart-type est fixe et ne dépend que de la variable considérée. Toussaint et Igel [130] soulignent l'intérêt de la neutralité, dans une version plus efficace de l'algorithme<sup>10</sup>. En effet, chaque vecteur de nombres réels, est complété par le vecteur des écart-types du bruit blanc réalisant la mutation; on ajoute ainsi de la neutralité dans le paysage. Là encore, un opérateur de croisement permet de recombinaison efficacement les écart-types<sup>11</sup>. Les performances comparées des deux algorithmes sont largement en faveur de la version avec neutralité. Les auteurs mettent en avant le fait que la neutralité autorise l'auto-adaptation des paramètres de recherche sans perte de valeur de performance. Si l'adaptation des écart-types s'effectuait uniquement sous l'effet de la pression sélective, on ne pourrait pas optimiser le réglage des paramètres. Toutefois, il faut noter que si la recombinaison de ces paramètres n'était pas réalisée de manière pertinente, la neutralité à elle seule ne serait pas suffisante pour permettre l'auto-adaptation.

Dans le contexte des algorithmes génétiques manipulant des chaînes binaires de longueur fixe, Collard *et al* [20] ont proposé une approche parcimonieuse de la redondance. On suppose que les solutions sont initialement codées par des chaînes binaires de longueur  $\lambda$  fixée et l'on "plonge" l'espace génotypique  $S$  de base dans un espace  $S'$  de dimension  $\lambda + 1$  via l'ajout d'un nouveau bit en tête de la chaîne. Cet espace étendu devient alors l'espace de recherche effectif d'un algorithme génétique. Deux chaînes complémentaires  $0x$  et  $1\bar{x}$  dans  $S'$  correspondent au même génotype de base  $x$  et donc partagent la même performance  $f(x)$ . De plus les auteurs introduisent un nouvel opérateur neutre spécifique, dit de *mirroring*, qui transforme une chaîne de  $S'$  en la chaîne complémentaire. Cet opérateur est neutre car il a aucune influence sur la performance. Du point de vue de la neutralité, et indépendamment de la neutralité intrinsèque du problème, on a ainsi créé  $2^\lambda$  réseaux de neutralité de taille 2 dans lesquels on peut se déplacer par mirroring. On a là un exemple d'exploitation minimale du concept de neutralité. Notons toutefois que les auteurs proposent une généralisation de cette approche synthétique en considé-

---

<sup>10</sup>Ce n'est sans doute pas la meilleure version connue de l'algorithme mais le but ici est de comparer deux algorithmes similaires avec et sans neutralité

<sup>11</sup>pour plus de détail voir [130] ou [5]

rant des espaces  $S'$  de taille  $\lambda + k$ . L'intérêt de l'approche est mise en évidence sur des fonctions trompeuses présentant une multitude d'optima locaux. Un algorithme génétique classique est utilisé ( $\lambda = 200$ , TauxCroisement=0,7, TauxMutation=0,5 par chaîne, TaillePop=100). Sans neutralité, après  $10^3$  générations, la population est piégée sur un optimum local. Par contre l'introduction de neutralité (TauxMirroring=0,02) permet d'atteindre l'unique optimum global. L'examen de la dynamique montre une succession de périodes lors desquelles la performance évolue peu et la distance entre les chaînes augmente. Ces périodes sont entrecoupées de brusques augmentations de la performance. On retrouve là la dynamique des équilibres ponctués observée dans les paysages adaptatifs neutres.

Ces trois derniers exemples montrent que l'introduction d'une redondance structurée couplée avec des opérateurs capables d'exploiter la neutralité, peut être une méthode pertinente pour concevoir des algorithmes évolutionnaires performants.

## 1.5 Synthèse du chapitre

Le concept de paysage adaptatif issu de la biologie de l'évolution s'est imposé dans d'autres domaines scientifiques pour l'étude des systèmes dynamiques. En optimisation combinatoire, il permet de représenter l'ensemble des solutions potentielles et leur performance en tenant compte de l'opérateur local de recherche. À chaque représentation "géographique" d'un paysage donné correspond une dynamique particulière d'optimisation, ou mieux, une méthode d'optimisation adaptée.

L'exploration d'un paysage adaptatif peut s'effectuer en sélectionnant aléatoirement des solutions ou à l'aide de "marches" en utilisant la relation de voisinage. Deux géométries principales ont été étudiées.

La première relative aux optima locaux et à la régularité du paysage, appelée aussi rugosité, permet d'expliquer les dynamiques des recherches qui se déplacent toujours vers de meilleures solutions. Les optima locaux correspondent aux conditions d'arrêt de ces processus et la rugosité mesure la corrélation de performance entre solutions voisines. En particulier, la notion d'évolvabilité exprime la possibilité de proposer de meilleures solutions.

La seconde géométrie relative à l'existence de plateaux dans un paysage, appelé *réseaux de neutralité*, explique des dynamiques de dérive aléatoire et d'équilibres ponctués. Celle-ci découverte plus récemment dans les années 1960 par M. Kimura dans le domaine de la biologie moléculaire. Par exemple, dans la mise au point d'un contrôleur de robot ou dans l'optimisation par programmation génétique. Certains problèmes académiques comme le problème des Routes Royales, MAX-SAT ou le problème de majorité présentent également des géométries neutres. Des paysages dédiés à l'étude de la neutralité ont été conçus et seront étudiés en détail dans le chapitre 3. La meilleure connaissance du concept de neutralité a permis de concevoir des métaheuristiques adaptés (Nageur de Réseau et Extrema Sélection) à l'optimisation de problèmes conduisant à des paysages neutres. Par ailleurs, des auteurs ont tenté d'introduire de la neutralité dans la résolution de problème afin d'améliorer les performances des métaheuristiques. Seul l'introduction de neutralité structurée a permis une amélioration significative.

Les problèmes conduisant à des paysages neutres sont des espaces redondants où de nombreuses solutions ont une même performance. Ce qui conduit à définir les ensembles des solutions de même performance, les *ensembles de neutralité*. Dans le chapitre suivant, nous allons définir des outils d'études de ces ensembles à l'aide de la notion d'évolvabilité.

## Chapitre 2

# Ensemble de Neutralité : Nuage Adaptatif

Nous allons définir et utiliser dans ce chapitre le *Nuage Adaptatif* (NA) qui permet d'étudier la corrélation de performance entre solutions voisines relativement à un opérateur de recherche local. L'idée d'analyser la corrélation de performance n'est pas nouvelle comme nous avons pu le voir dans la section 1.3.2. Galton [43] a développé des outils statistiques, comme par exemple la *droite de régression*, afin d'étudier l'hérédité de caractères quantitatifs et de prédire leur évolution. Dans le contexte des paysages adaptatifs, certaines statistiques ont permis d'étudier l'évolvabilité, *i.e.* la capacité d'un opérateur à produire de meilleures solutions. L'avantage de la notion de nuage adaptatif est qu'elle permet d'unifier un grand nombre de mesures relatives à l'évolvabilité et permet l'analyse du passage d'un ensemble de neutralité à un autre via un opérateur local,

Le nuage adaptatif ne doit pas être considéré comme une alternative au paysage adaptatif mais plutôt comme un outil d'analyse de celui-ci. Nous avons introduit ce concept dans l'article [137] et nous allons donner dans une première section une définition et les mesures statistiques déduites du NA. Dans une deuxième section, nous allons présenter des résultats concernant l'opérateur local de recherche aléatoire sur une classe de paysages généralisant les paysages NK et MAX-SAT. Dans une troisième section, nous étudierons d'autres opérateurs locaux relatifs à la marche adaptative et à la métaheuristique du recuit simulé. Enfin, nous proposerons un indicateur permettant de "mesurer" la difficulté à optimiser pour la programmation génétique.

### 2.1 Définition et propriétés

La notion de *Nuage Adaptatif* (NA) a été présentée dans les travaux [137] et [22]. Celle-ci a aussi été développée de façon indépendante par Barnett ([8] page 30) et étudiée sur les paysages NKp. Elle consiste à représenter la corrélation entre solutions voisines relativement à un opérateur. Les espaces de recherche sont généralement grands et il n'est pas possible de représenter l'ensemble de tous les points  $(x, y)$  pour tout  $x \in \mathcal{S}$  et  $y \in \mathcal{V}(x)$ . Aussi dans un nuage adaptatif, nous partitionnons l'ensemble des solutions par classe de solutions de même performance :  $\mathcal{S}_\varphi = \{x \in \mathcal{S} \mid f(x) = \varphi\}$ . Cette partition relative à la neutralité du paysage correspond à la notion d'ensemble de neutralité lorsque la structure de voisinage du paysage n'est pas prise en compte. Nous verrons dans les sections suivantes que cette partition est suffisante pour modéliser la dynamique d'évolution à long terme d'une métaheuristique pour certains opérateurs locaux sur une famille de paysages définis additivement. La corrélation est alors décrite sous forme de distribution de probabilité conditionnelle d'obtenir une solution de performance donnée après application d'un opérateur local connaissant sa performance initiale.

**Définition:** Soient  $(\mathcal{S}, \mathcal{V}, f)$  un paysage adaptatif et  $op : \mathcal{S} \rightarrow \mathcal{S}$  un opérateur<sup>12</sup> local agissant sur  $\mathcal{S}$  tel que pour tout  $s \in \mathcal{S}$ ,  $op(s) \in \mathcal{V}(s)$ . Notons  $X : \mathcal{S} \rightarrow \mathbb{R}$  la v.a définie par  $X(s) = f(s)$  et  $Y : \mathcal{S} \rightarrow \mathbb{R}$  l'évolvabilité, *i.e.* la v.a. définie par  $Y(s) = f(op(s))$ . Le *nuage adaptatif* relatif à l'opérateur  $op$  est la densité de probabilité conditionnelle bivariée de  $Y$  sachant  $X$  noté  $p_{Y|X}^{op}$  ou plus simplement  $p^{op}$

La figure 2.1 montre un exemple de nuage adaptatif sur un paysage NK. Le NA décrit l'évolvabilité (voir section 1.3.2) des solutions de même performance. En effet, pour chaque valeur de performance, le NA est la distribution de probabilité des valeurs d'adaptation (voir figure 2.2). Cette distribution permet d'évaluer pour chaque niveau de performance, la probabilité d'obtenir de meilleures performances après application d'un opérateur ou la performance moyenne des solutions qui améliorent l'adaptation.

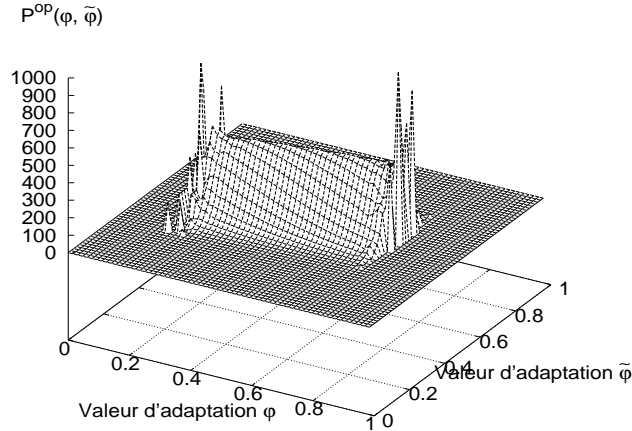


FIG. 2.1 – Nuage adaptatif d'un paysage NK de paramètres  $N = 25$  et  $K = 20$  relativement à l'opérateur local de recherche aléatoire.

Pour visualiser le nuage adaptatif en deux dimensions, nous dessinons l'ensemble des points  $(\varphi, \tilde{\varphi})$  tels que  $p^{op}(\varphi, \tilde{\varphi}) \neq 0$ . Cette représentation justifie le terme de nuage. Nous appellerons *contour* du nuage l'ensemble des points  $(\varphi, \tilde{\varphi}_{min}), (\varphi, \tilde{\varphi}_{max})$  où  $\tilde{\varphi}_{min}$  et  $\tilde{\varphi}_{max}$  sont respectivement les minimum et maximum de  $\tilde{\varphi}$  tels que  $p^{op}(\varphi, \tilde{\varphi}) \neq 0$ . La figure 2.3 montre un exemple d'un tel contour.

Afin d'obtenir le NA, lorsque l'espace de recherche est très petit, il est possible de calculer l'ensemble des points  $(f(x), f(op(x)))$  pour tout  $x \in \mathcal{S}$ . Cependant, la plupart des espaces de recherche ne permettent pas ce calcul, il est alors nécessaire d'échantillonner l'espace de recherche. L'échantillonnage uniforme sur l'espace de recherche est une méthode d'estimation du NA. Néanmoins les solutions visitées par cette méthode sont peu utilisées par une métaheuristique qui utilise les solutions de "haute" performance. Nous utiliserons donc dans la section 2.4 une estimation basée sur l'échantillonnage de Métropolis.

Plusieurs statistiques pertinentes peuvent être déduites du NA. Une statistique pour mesurer la tendance centrale est la fonction de régression  $\tilde{E}$  qui donne la moyenne de  $\tilde{\varphi}$  des valeurs d'adaptation atteignables à partir d'une solution de performance  $\varphi$  :  $\tilde{E}(\varphi) = E(Y | X = \varphi)$ .

<sup>12</sup>cet opérateur peut-être stochastique

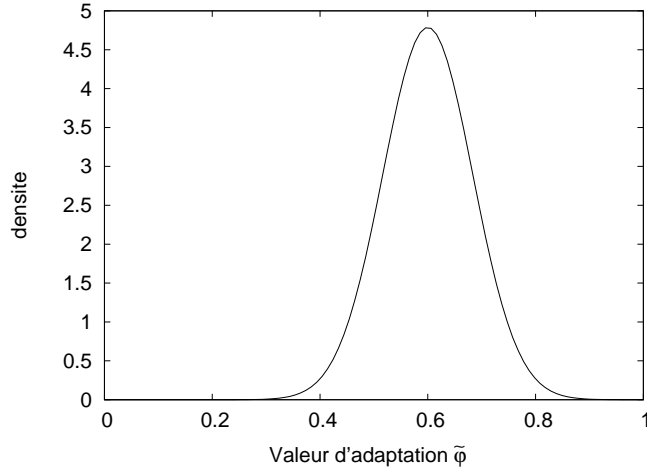


FIG. 2.2 – Exemple d'évolvabilité pour la valeur d'adaptation  $\varphi = 0.6$

Nous appellerons la *courbe moyenne* la courbe représentative de la fonction  $\tilde{E}$ . Pour estimer la dispersion de la distribution, nous utiliserons de la même manière la fonction donnant l'écart-type  $\sigma(Y \mid X = \varphi)$ .

De la fonction de régression, nous pouvons distinguer deux cas, selon la position de la courbe moyenne par rapport à la première bissectrice. Un cas typique correspond à  $\tilde{E}$  croissante avec une valeur unique de performance  $\beta$  solution de l'équation  $\beta = \tilde{E}(\beta)$ , correspondant à l'intersection de la courbe moyenne avec la première bissectrice d'équation  $\tilde{\varphi} = \varphi$ . Selon la valeur d'adaptation  $\varphi$ , l'opérateur local agit différemment sur la performance (voir la figure 2.3) :

1. si  $\varphi \leq \beta$  :  $\tilde{\varphi}$  est en moyenne plus haute que la valeur  $\varphi$ . Ainsi, en moyenne, l'opérateur local est avantageux.
2. si  $\beta < \varphi$  :  $\tilde{\varphi}$  est en moyenne plus basse que la valeur  $\varphi$ . Ainsi, en moyenne, l'opérateur local est désavantageux.

La fonction de régression informe aussi sur le comportement moyen de l'heuristique définie par l'itération de l'opérateur local. Pour une valeur de performance  $\varphi$ , en moyenne après application de l'opérateur, nous obtenons la performance  $\tilde{E}(\varphi)$ . Définissons la suite  $(\varphi_i)_{i \geq 0}$  par  $\varphi_0 \in f(\mathcal{S})$  et  $\forall i > 0, \varphi_{i+1} = \tilde{E}(\varphi_i)$  et notons  $op^i(s)$  la solution obtenue après application de  $i$  fois de l'opérateur local. La question est de savoir si la performance moyenne après l'application itérée de l'opérateur local sur une solution  $s$  peut être approchée par composition de la fonction  $\tilde{E}$  sur la valeur d'adaptation  $f(s)$ . Autrement dit, nous cherchons à savoir si la différence

$$E(f(op^i(s))) - \tilde{E}^i(s) \tag{2.1}$$

est négligeable. Si la différence est suffisamment petite, le NA permettrait de prédire le comportement à long terme de l'heuristique.

## 2.2 Modèle analytique relatif à une marche aléatoire

Il est bien sûr impossible d'obtenir une expression analytique du NA valable sur l'ensemble des paysages adaptatifs. Dans cette section, nous allons donc étudier le NA relativement à l'opérateur local de recherche aléatoire qui sélectionne une solution uniformément parmi l'ensemble de toutes les solutions voisines sur la famille de *paysages embarqués uniformes* généralisant les



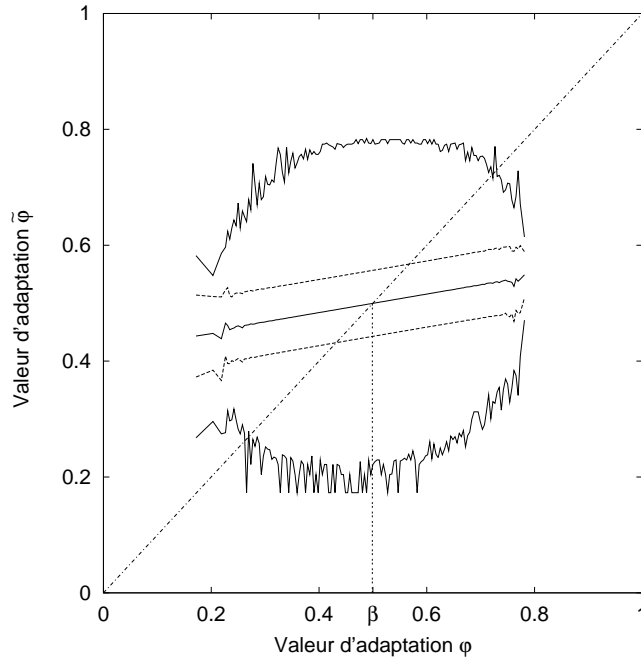


FIG. 2.3 – Contour et courbe moyenne du nuage adaptatif d’un paysage NK de paramètre  $N = 25$  et  $K = 20$  relativement à l’opérateur local de recherche aléatoire.

paysages NK et MAX-SAT. Après avoir défini cette famille de paysages, nous montrons que le NA est une somme de densités binormales et que la fonction de régression est linéaire. Cette étude généralise les travaux de l’article [137] réalisé sur les paysages NK à une plus large famille de paysages et donne l’expression analytique du NA sur celle-ci.

### 2.2.1 Famille des paysages embarqués

Heckendorn [55] a introduit la famille des *paysages embarqués* dans lesquels “beaucoup de petits sous-problèmes interagissent les uns avec les autres de manière à construire un problème plus complexe”. Cette famille de paysages est très proche de la famille des *paysages additifs aléatoires* définie parallèlement par Reidys et Stadler [107]. La définition des paysages embarqués explicite clairement les interactions entre sous-problèmes ; de plus, les résultats sur les paysages embarqués peuvent être adaptés aux paysages additifs aléatoires. C’est pour ces raisons que nous avons choisi les paysages embarqués comme base de notre étude.

Notons  $\mathcal{B}$  l’ensemble  $\{0, 1\}$ . L’espace de recherche  $\mathcal{S} = \mathcal{B}^N$  est l’ensemble des chaînes binaires de  $N$  bits et le voisinage  $\mathcal{V}(s)$  d’une solution  $s$  est l’ensemble des solutions à une distance de Hamming 1. Les paramètres d’un paysage embarqué sont  $P$  chaînes binaires  $m_j \in \mathcal{B}^N$  et  $P$  fonctions  $g_j : \mathcal{B}^{bc(m_j)} \rightarrow \mathbb{R}$ , où  $bc(m_j)$  est le nombre de 1 de la chaîne  $m_j$ . Afin d’alléger les notations, nous noterons  $\mathcal{B}^{m_j} = \mathcal{B}^{bc(m_j)}$ . La fonction d’adaptation  $g : \mathcal{B}^N \rightarrow \mathbb{R}$  est alors la somme des  $P$  fonctions “embarquées”  $g_j$  :

$$g(s) = \sum_{j=1}^P g_j(pack_j(s, m_j)) \quad (2.2)$$

où  $pack_j : \mathcal{B}^N \times \mathcal{B}^N \rightarrow \mathcal{B}^{m_j}$  est la fonction qui masque les bits de  $s$  avec les  $N$  bits du masque  $m_j$ .

Les paysages embarqués généralisent les paysages NK et MAX-SAT (voir section 1.4.3) qui sont tous deux des problèmes NP-complets<sup>13</sup>. Dans le cas des paysages NK, le nombre de masques et de fonctions embarquées est égale à la longueur des chaînes binaires  $P = N$ ; les masques  $m_j$  sont tels que  $bc(m_j) = K + 1$  ( $m_j = 0^{j-1}1^{K+1}0^{N-K-1}$  dans le cas adjacent et les 1 sont uniformément distribués dans le cas aléatoire); les fonctions embarquées  $g_j$  sont définies à partir des contributions  $g_j = \frac{1}{N}f_j$ . Dans le cas des paysages MAX-SAT, le nombre de fonctions embarquées correspond au nombre de clauses  $P = m$ ; les masques  $m_j$  sont déterminés par les variables de la clause  $j$ ,  $\forall i \in \{1, N\}$ ,  $m_j[i] = 1$  ssi la variable numéro  $i$  apparaît dans la  $j^{\text{ème}}$  clause; enfin les fonctions embarquées  $g_j$  sont définies à l'aide de l'évaluation des littéraux de la clause  $j$ .

## Paysages embarqués uniformes

Il n'est pas possible de donner une expression analytique générale pour l'ensemble des paysages embarqués. Bien que la définition des paysages embarqués semble distinguer les liens entre les variables des contributions, elle reste insuffisante quand au mode de construction des fonctions embarquées. Nous définissons une famille particulière de paysages embarqués, les *Paysages Embarqués Uniformes* (notés PEU), où toutes les fonctions embarquées  $g_j$  sont construites indépendamment et de manière identique à l'aide d'une variable aléatoire. Les paysages embarqués uniformes précisent l'idée que la dépendance entre les variables est exprimée seulement par les masques  $m_j$  et que les fonctions  $g_j$  sont statistiquement indépendantes. Ces paysages sont dits uniformes car toutes les valeurs des fonctions embarquées sont générées à partir d'une même v.a. et que la corrélation entre ces valeurs au sein d'une même fonction embarquée est constante.

Plus formellement, nous notons  $\mathbb{P}$  l'ensemble des entiers compris entre 1 et  $P$ . Soient  $\forall j \in \mathbb{P}$ ,  $P$  propriétés  $W_j$  servant à construire un ensemble de fonctions embarquées  $g_j : \forall j \in \mathbb{P}, \mathcal{G}_j = \{h : \mathcal{B}^{m_j} \rightarrow \mathbb{R} \mid h \text{ vérifiant la propriété } W_j\}$ . Soient les  $P$  espaces de probabilité  $\Omega_j = (\mathcal{G}_j, \mathcal{A}_j, \mu_j)$  où  $\mathcal{A}_j$  est une tribu sur  $\mathcal{G}_j$  et  $\mu_j : \mathcal{A}_j \rightarrow [0, 1]$  une mesure sur  $\mathcal{G}_j$ . Nous considérons la famille des variables aléatoires  $\forall j \in \mathbb{P}, \forall s \in \mathcal{B}^{m_j}, X_{j,s} : \Omega_j \rightarrow \mathbb{R}$  telles que  $X_{j,s}(g) = g_j(\text{pack}(s, m_j))$ . les v.a.  $X_{j,s}$  évaluent la solution  $s$  d'une fonction embarquée vérifiant la propriété  $W_j$ .

**Définition:** Un paysage embarqué uniforme  $(\mathcal{S}, \mathcal{V}, g)$ , paramétré par  $(W_j)_{j \in \mathbb{P}}, (m_j)_{j \in \mathbb{P}}$ , une v.a.  $X$  et un coefficient de corrélation  $\rho$ , noté  $PEU_{P,W,m,X,\rho}$ , est un paysage vérifiant :

1.  $(\mathcal{S}, \mathcal{V}, g)$  est un paysage embarqué,
2.  $\forall j \in \mathbb{P}, g_j \in \mathcal{G}_j$ ,
3.  $\forall j \in \mathbb{P}, \forall s \in \mathcal{B}^{m_j}, X_{j,s}$  est de loi  $X$ ,
4.  $\forall j \in \mathbb{P}, \forall s \in \mathcal{B}^{m_j}, \forall s' \in \mathcal{B}^{m_j}$ , la corrélation entre  $X_{j,s}$  et  $X_{j,s'}$  est constante égale à  $\rho$
5.  $\forall i \in \mathbb{P}, \forall j \in \mathbb{P}, i \neq j, \forall s \in \mathcal{B}^{m_i}, \forall s' \in \mathcal{B}^{m_j}$  et  $X_{i,s}$  et  $X_{j,s'}$  sont indépendantes.

Cette définition n'est pas trop restrictive puisque, par exemple, les paysages NK et les problèmes MAX-k-SAT sont des paysages embarqués uniformes. Pour les paysages NK, chaque valeur des fonctions de contribution est choisie uniformément et indépendamment dans  $[0, 1]$ .  $\forall i \in \mathbb{P}, \forall j \in \mathbb{P}, \forall s \in \mathcal{B}^{m_i}, \forall s' \in \mathcal{B}^{m_j}, X_{i,s}$  et  $X_{j,s'}$  sont indépendantes et suivent la même loi uniforme  $\mathcal{U}(0, 1/N)$  et  $\rho = 0$ .

Dans les paysages MAX-k-SAT, chaque clause contient exactement  $k$  littéraux dont les variables sont choisies uniformément et sans remise dans l'ensemble des  $N$  variables. La propriété  $W$  est  $\exists! s_f \in \mathcal{B}^k, g_j(s_f) = 0$  et  $\forall s \neq s_f, g_j(s) = 1$  : pour chaque clause, toutes les valeurs des fonctions  $g_j$  sont égales à 1 sauf une valeur qui est égale à 0 lorsque tous les littéraux sont à la valeur *faux*. Par conséquent, pour toute solution  $s \in \mathcal{B}^k$ , les  $P$  v.a.  $(X_{j,s})_{j \in \mathbb{P}}$  suivent la

<sup>13</sup>lorsque  $k \geq 3$  pour MAX-SAT,  $K \geq 2$  pour le paysage NK

même loi de Bernoulli de paramètre  $1 - (1/2)^k$  et  $\forall i \neq j, \forall (s, s') \in (\mathcal{B}^k)^2$ ,  $X_{i,s}$  et  $X_{j,s'}$  sont indépendantes.  $X_{j,s}$  et  $X_{j,s'}$  ne sont plus indépendantes, par exemple, lorsqu'une clause évaluée sur une solution est fausse, elle devient vraie dès que l'on modifie un bit de cette solution. Or,  $E(X_{j,s}X_{j,s'})$  est égale à la probabilité de  $(X_{j,s} = 1) \cap (X_{j,s'} = 1)$ . En notant  $P(i)$  la probabilité d'avoir  $i$  littéraux vrais dans une clause, nous avons donc :

$$\begin{aligned} E(X_{j,s}X_{j,s'}) &= \sum_{i=1}^k P(X_{j,s'} = 1|i)P(i) \\ &= \sum_{i=1, i \neq d}^k P(i) + P(d)\left(1 - \frac{1}{\binom{k}{d}}\right) \\ &= 1 - 2^{-k+1} \end{aligned}$$

On obtient le coefficient de corrélation qui est indépendant du nombre de variables modifiées  $\rho = \frac{-1}{2^{k-1}}$ .

### 2.2.2 Expression analytique du NA sur la famille des $PEU$

Dans cette section, nous reprenons les notations sur les paysages embarqués uniformes de la section précédente. Nous allons donner l'expression du NA pour la famille des paysages embarqués uniformes  $PEU_{W,m,X,\rho}$  relativement à l'opérateur de recherche aléatoire  $op^d$  qui sélectionne de manière équiprobable une solution à une distance de Hamming  $d$ . Cette section dans la continuité des travaux initiés dans [137, 22], généralise les précédents résultats sur les mesures d'évolvabilité de Smith *et al* des paysages NKp et NKq [119] et généralise à une famille plus large de paysages les travaux de Barnett [8] sur les paysages NKp.

Le nuage adaptatif de la famille des paysages embarqués uniformes  $PEU_{W,m,X,\rho}$  est caractérisé par les nombres  $c(n) = \frac{n}{P}(1 - \rho)$ . Lorsque  $P$  est suffisamment grand, le NA a pour densité  $p^{op^d}$  qui est une somme de densités binomiales :

$$p^{op^d}(\tilde{\varphi}|\varphi) = \gamma^d(n)\delta_\varphi(\tilde{\varphi}) + \sum_{n=1}^P \gamma^d(n)p_n(\tilde{\varphi}|\varphi) \quad (2.3)$$

où  $\delta_\varphi$  est la distribution de Dirac en  $\varphi$  et  $p_n$  est la densité d'une loi conditionnelle binormale :

$$\begin{aligned} p_n(\tilde{\varphi}|\varphi) &= \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{(\tilde{\varphi} - \mu_n(\varphi))^2}{2\sigma_n^2}\right) \\ \text{avec } \mu_n(\varphi) &= P.E(X) + (1 - c(n))(\varphi - P.E(X)), \\ \sigma_n^2 &= c(n)(2 - c(n))P\sigma_X^2, \end{aligned}$$

$\gamma^d(n)$  la probabilité que  $n$  fonctions embarquées soient modifiées par l'opérateur  $op^d$ .

L'équation de la courbe moyenne est alors :

$$\tilde{E}(\varphi) = P.E(X) + (1 - C^d)(\varphi - P.E(X)) \quad (2.4)$$

où  $C^d = \frac{E(\gamma^d)}{P}(1 - \rho)$  où  $E(\gamma^d)$  est l'espérance de  $\gamma^d$ .

Preuve :

Nous allons étudier la corrélation de performance entre deux solutions distantes au plus de  $d$  sur l'ensemble des fonctions embarquées uniformes possibles. Soient  $s$  une solution de  $\mathcal{S}$ , et  $s^d$  l'une des  $2^d$  solutions à, au plus, une distance  $d$  de Hamming. Soient  $F(g) = g(s)$  et  $F^d(g) = g(s^d)$

les v.a. donnant respectivement les performances de  $s$  et  $s^d$  d'une fonction d'adaptation  $g$  d'un paysage de la famille  $PEU_{W,m,X,\rho}$ . Les v.a.  $F$  et  $F^d$  sont les sommes de v.a.  $X_{j,s_j}$ , nous avons donc :

$$F = \sum_{j=1}^P X_{j,s_j} \quad (2.5)$$

$$F^d = \sum_{j=1}^P X_{j,s_j^d} \quad (2.6)$$

où  $s_j = \text{pack}(s, m_j) \in \mathcal{B}^{m_j}$  et  $s_j^d = \text{pack}(s^d, m_j) \in \mathcal{B}^{m_j}$ .

Le NA consiste à calculer la distribution conditionnelle de  $F^d$  sachant  $F$ . Pour cela, nous allons d'abord calculer cette distribution lorsque  $n$  fonctions embarquées sont modifiées, pour ensuite moyenner ces distributions sur l'ensemble des  $n$  valeurs possibles. En effet, soit  $N$  la v.a. qui indique le nombre de fonctions embarquées modifiées entre les solutions  $s$  et  $s^d$  et  $\gamma^d(n) = P(N = n)$  la probabilité de modifier  $n$  fonctions embarquées par l'opérateur  $op^d$ , nous pouvons écrire :

$$p_{F^d|F}(\tilde{\varphi}|\varphi) = \sum_{n=0}^P p_{F^d,N|F}(\tilde{\varphi}, n|\varphi) \quad (2.7)$$

$$= \sum_{n=0}^P \gamma^d(n) p_{F^d|F,N}(\tilde{\varphi}|\varphi, n) \quad (2.8)$$

Dans un premier temps, calculons  $p_{F^d|F,N}(\tilde{\varphi}|\varphi, n)$  pour  $n \geq 1$ .  $F$  et  $F^d$  se décompose suivant les v.a.  $X_{j,\cdot}$  modifiées et l'on peut supposer, sans perte de généralité, que ce sont les  $n$  premières qui sont différentes :

$$F = U + V \quad (2.9)$$

$$F^d = U^d + V \quad (2.10)$$

avec

$$U = \sum_{j=1}^n X_{j,s_j} \quad , \quad V = \sum_{j=1}^n X_{j,s_j}$$

$$U^d = \sum_{j=1}^n X_{j,s_j^d}$$

$V$  est la partie commune entre  $F$  et  $F^d$ , et  $U$  et  $U^d$  les parties altérées par l'opérateur. D'après la définition des PEU,  $F$  et  $F^d$  sont des sommes de variables indépendantes de même loi  $X$ ,  $F$  et  $F^d$  suivent donc des lois normales lorsque  $P$  est suffisamment grand. Nous utilisons le résultat suivant concernant les lois normales bivariées (voir par exemple [6]) :

Si  $(N_1, N_2)$  est un vecteur gaussien où  $N_1$  et  $N_2$  sont deux v.a. suivant respectivement les lois normales  $\mathcal{N}(\mu_1, \sigma_1)$  et  $\mathcal{N}(\mu_2, \sigma_2)$  et de corrélation  $r$ , alors la distribution conditionnelle  $p_{N_1|N_2}$  de  $N_1$  sachant  $N_2$  a pour densité :

$$p_{N_1|N_2}(y|x) = \frac{1}{\sqrt{2\pi(1-r^2)}\sigma_1} \exp\left(-\frac{1}{2(1-r^2)\sigma_1^2} \left(y - \mu_1 - r\frac{\sigma_1}{\sigma_2}(x - \mu_2)\right)^2\right)$$

et pour un  $x$  fixé,  $p_{N_1|N_2}(y|x)$  suit une loi normale de moyenne  $\mu_1 + r\frac{\sigma_1}{\sigma_2}(x - \mu_2)$  et de variance  $\sigma_1^2(1-r^2)$ .

Pour déterminer la densité, il suffit donc de calculer le coefficient de corrélation  $\rho_{F,F^d}$  entre  $F$  et  $F^d$ .

$$\text{cov}(F, F^d) = \text{cov}(U, U^d) + \text{cov}(U, V) + \text{cov}(U^d, V) + \text{var}(V) \quad (2.11)$$

or,  $U$  et  $V$  sont indépendantes, de même pour  $U^d$  et  $V$ , d'où

$$\text{cov}(F, F^d) = \text{cov}(U, U^d) + \text{var}(V) \quad (2.12)$$

Les v.a.  $X_{j,s}$  sont indépendantes et de même variance  $\sigma_X^2$ , nous avons donc :

$$\sigma_V^2 = (P - n)\sigma_X^2 \quad (2.13)$$

$$\sigma_F^2 = P\sigma_X^2 \quad (2.14)$$

Pour toutes solutions  $s_1$  et  $s_2$  et pour tout  $i$  et  $j \neq i$ , les v.a.  $X_{i,s_1}$  et  $X_{j,s_2}$  sont indépendantes, la covariance  $\text{cov}(U', U)$  entre  $U'$  et  $U$  s'exprime donc par :

$$\text{cov}(U^d, U) = \sum_{j=1}^n \text{cov}(X_{j,s_j}, X_{j,s_j^d}) \quad (2.15)$$

Or, la corrélation entre  $X_{j,s_j}$  et  $X_{j,s_j^d}$  est constante, nous avons donc :

$$\text{cov}(U^d, U) = n\rho\sigma_X^2 \quad (2.16)$$

On obtient alors le coefficient de corrélation

$$\rho_{F,F^d} = \frac{\text{cov}(U, U^d) + \sigma_V^2}{\sigma_F\sigma_{F^d}} \quad (2.17)$$

$$= \frac{n\rho\sigma_X^2 + (P - n)\sigma_X^2}{P\sigma_X^2} \quad (2.18)$$

$$= 1 - \frac{n}{P}(1 - \rho) \quad (2.19)$$

Calculons maintenant,  $\mu_n(\varphi)$  et  $\sigma_n^2$  la moyenne et la variance sachant la performance  $\varphi$  et  $n$ .

$$\mu_n(\varphi) = PE(X) + \rho_{F,F^d} \frac{\sigma_{F^d}}{\sigma_F} (\varphi - PE(X)) \quad (2.20)$$

$$= PE(X) + (1 - \frac{n}{P}(1 - \rho))(\varphi - PE(X)) \quad (2.21)$$

$$= PE(X) + (1 - c(n))(\varphi - PE(X)) \quad (2.22)$$

$$\sigma_n^2 = \sigma_{F^d}^2(1 - \rho^2) \quad (2.23)$$

$$= n(1 - \rho)(2 - \frac{n}{P}(1 - \rho))\sigma_X^2 \quad (2.24)$$

$$= c(n)(2 - c(n))P\sigma_X^2 \quad (2.25)$$

avec  $c(n) = \frac{n}{P}(1 - \rho)$ .

La distribution conditionnelle lorsque  $n$  fonctions embarquées sont modifiées avec  $n \geq 1$  est donc  $p_{F^d|F,N}(\tilde{\varphi}|\varphi, n) = \frac{1}{\sqrt{2\pi\sigma_n}} \exp(-\frac{(\tilde{\varphi} - \mu_n(\varphi))^2}{2\sigma_n^2})$ . Pour  $n = 0$ ,  $F$  et  $F^d$  sont identiques, la densité conditionnelle  $p_{F^d|F,N}(\tilde{\varphi}|\varphi, 0)$  est alors la distribution de Dirac  $\delta_\varphi(\tilde{\varphi})$ .

Maintenant l'équation 2.7 permet d'obtenir le NA en sommant les distributions  $p_{F^d|F,N}$  :

$$p^{op^d}(\tilde{\varphi}|\varphi) = \sum_{n=0}^P \gamma^d(n) p_{F^d|F,N}(\tilde{\varphi}|\varphi) \quad (2.26)$$

L'équation de la courbe moyenne peut être déduite de l'expression du NA :

$$\tilde{E}(\varphi) = E(F_d | F = \varphi) \quad (2.27)$$

$$= \sum_{n=0}^P E(\gamma^d(n)(P.E(x) + (1 - \frac{n}{P}(1 - \rho))(\varphi - P.E(X)))) \quad (2.28)$$

$$= P.E(X) + (1 - C^d)(\varphi - P.E(X)) \quad (2.29)$$

où  $C^d = \frac{E(\gamma^d)}{P}(1 - \rho)$  où  $E(\gamma^d)$  est l'espérance de  $\gamma^d$  ■

Toutes les densités binormales  $p_n$  sont centrées sur le point moyen  $G = (P.E(X), P.E(X))$ . Les moyennes  $\mu_n$  sont des fonctions affines de la variable  $\varphi$  centrées sur le point  $G$  dont le coefficient de proportionnalité est  $1 - c(n)$ . La pente est comprise entre  $-1$  et  $1$  et est positive si et seulement si la corrélation  $\rho$  est positive. La pente est maximale lorsque  $c(n)$  est nul, et est minimale lorsque  $c(n) = 1$ . Les variances  $\sigma_n$  sont indépendantes de  $\varphi$  et sont des fonctions trinômes par rapport à  $c(n)$ , la variance est maximale et vaut  $P\sigma_X^2$  lorsque  $c(n) = 1$  et est nulle lorsque  $c(n) = 0$  ou  $c(n) = 2$ .

Les nombres  $c(n)$  sont le produit du facteur  $\frac{n}{P}$ , qui représente l'épistasie *i.e.* la dépendance entre les variables, et du facteur  $1 - \rho$ , qui représente le degré de corrélation entre les valeurs d'une même fonction embarquée. L'épistasie et la non-corrélation influencent de la même manière les moyennes et les variances des distributions conditionnelles  $p_{F|F,N}$ . En effet, la variation d'un facteur  $k$  de  $\frac{n}{P}$  est équivalente à la variation d'un facteur  $k$  de  $1 - \rho$ . En particulier,  $c(n)$  est nul lorsqu'il n'y a pas d'épistasie ou lorsque la corrélation est  $1$ , au contraire,  $|c(n)| = 1$  seulement si la corrélation est nulle et l'épistasie est maximale.

La courbe moyenne est une droite également centrée sur le point moyen  $G$ . La pente de cette droite  $1 - C^d$  s'interprète de la même manière que le coefficient  $1 - c(n)$  de l'expression des moyennes  $\mu_n$ . De même,  $\frac{E(\gamma^d)}{P}$  et  $1 - \rho$  modifient identiquement la pente de la courbe moyenne. Comme le montre Barnett [8], la pente de la droite moyenne  $1 - C^d$  est le coefficient d'autocorrélation entre solutions voisines par  $op^d$ . Le point d'intersection  $\beta$  entre la courbe moyenne et la première bissectrice est unique et vaut  $\beta = P.E(X)$ . La valeur  $\beta$  est alors le point de convergence de l'itération de l'opérateur local de recherche aléatoire.

## Application aux paysages NK

Pour la famille des paysages NK, le nombre de fonctions embarquées est  $P = N$ ,  $\rho$  est nul et la loi  $X$  est la loi uniforme  $\mathcal{U}(0, 1/N)$  d'où  $E(X) = \frac{0.5}{N}$  et  $\sigma_X^2 = \frac{1}{12N^2}$ . Dans le cas de liens épistatiques aléatoires, chaque contribution est modifiée par un bit, nous avons pour  $n < d$ ,  $\gamma^d(n) = 0$ . Les nombres  $m_j$  sont choisis indépendamment, pour  $0 \leq n$ ,  $\gamma^d(n + d)$  suit donc une loi binormale de paramètres  $N - d$  et  $\alpha_d$ ,  $\alpha_d$  étant la probabilité qu'une fonction de contribution soit affectée par un changement de  $d$  bits, d'où pour  $0 \leq n$ ,  $\gamma^d(n + d) = \binom{N}{n} \alpha_d^n (1 - \alpha_d)^{N-n}$  et  $E(\gamma^d) = (N - d)\alpha_d + d$ . Les  $d$  bits sont choisis uniformément et indépendamment parmi les  $N$  bits, la probabilité qu'au moins un bit modifie une contribution est donc  $\alpha_d = 1 - \frac{\binom{N-1-d}{K}}{\binom{N-1}{K}}$ .

La figure 2.4 montre la variation de  $\alpha_d$  en fonction du nombre de bits  $d$ .  $\alpha_d$  est une fonction croissante de  $d$  et de limite 1. La croissance est d'autant plus rapide que  $K$  est grand.

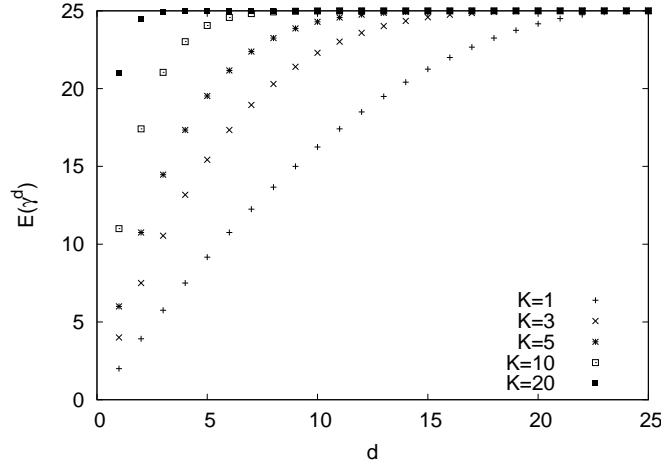


FIG. 2.4 – Nombre moyen  $E(\gamma^d)$  de contributions affectées en fonction du nombre  $d$  de bits changeant de valeur pour un paysage NK avec  $N = 25$  et différentes valeurs de  $K$ .

On obtient les paramètres de la densité binormale  $p_{F^d|F,N}(\tilde{\varphi}|\varphi, n)$  :

$$\mu_n(\varphi) = 0.5 + \left(1 - \frac{n}{N}\right)(\varphi - 0.5) \quad (2.30)$$

et

$$\sigma_n^2 = n\left(2 - \frac{n}{N}\right)\frac{1}{12N^2} \quad (2.31)$$

La figure 2.6 montre les densités conditionnelles théoriques pour  $N = 32$  et  $K = 8$  pour différentes valeurs de  $\varphi$ . Ces densités sont proches d'une densité normale.

L'équation 2.4 de la courbe moyenne devient :

$$\tilde{E}_d(\varphi) = 0.5 + \frac{\binom{N-1-d}{K}}{\binom{N-1}{K}}\left(1 - \frac{d}{N}\right)(\varphi - 0.5) \quad (2.32)$$

Cette équation confirme les précédents résultats lorsque  $d = 1$

$$\tilde{E}(\varphi) = 0.5 + \left(1 - \frac{K+1}{N}\right)(\varphi - 0.5) \quad (2.33)$$

L'équation de la courbe moyenne est confirmée expérimentalement. Le tableau 2.1 donne les paramètres de la courbe moyenne déterminés expérimentalement pour différentes valeur de  $N$ ,  $K$  (avec  $d = 1$ ). Pour chaque valeur des paramètres  $N$  et  $K$ , 300 instances de paysages NK sont générées afin d'évaluer les paramètres de la courbe moyenne. L'estimation de chaque NA s'effectue à l'aide de  $10^5$  solutions choisies aléatoirement de manière uniforme. Lorsque  $K$  est égale à 0, la droite est proche de la première bissectrice. A l'opposé, lorsque  $K$  est égale à  $N - 1$ , la droite est parallèle à l'axe des abscisses. La figure 2.7 montre deux exemples de contour du NA sur un paysage NK.

Afin de vérifier les distributions théoriques, nous avons comparé les écart-types expérimentaux et théoriques obtenus à partir de l'équation 2.3. La figure 2.8 représente les écart-types des distributions en fonction de la valeur d'adaptation  $\varphi$ . Les écart-types théoriques sont en adéquation avec ceux estimés expérimentalement, leur différence relative est inférieure à 3 %.

Enfin, nous avons réalisé le test du  $\chi^2$  entre les distributions théoriques et expérimentales qui confirme le modèle théorique.

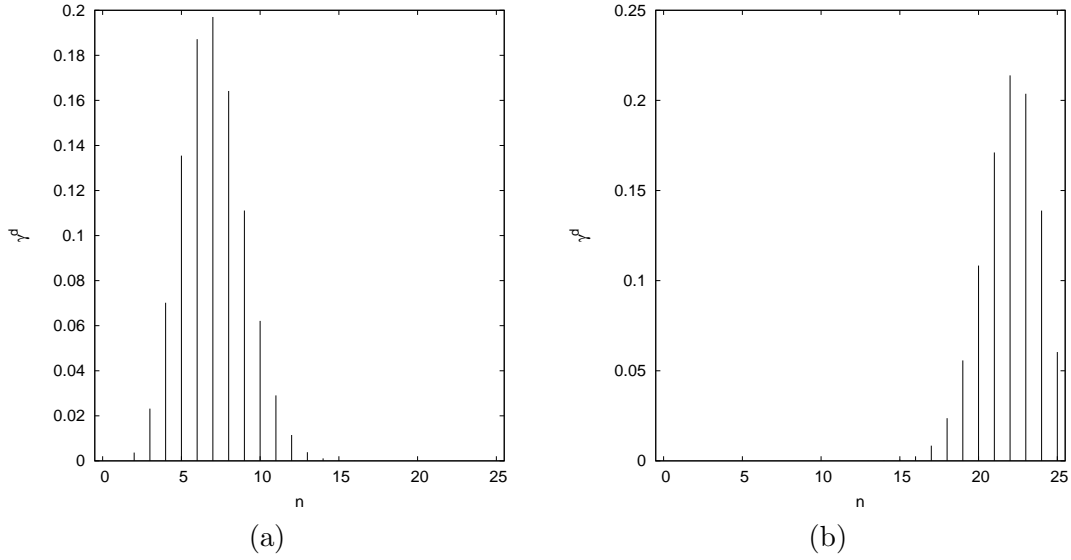


FIG. 2.5 – Distribution de probabilité  $\gamma^d(n)$  pour un paysage NK avec  $d = 1$ ,  $N = 25$ ,  $K = 5$  (a) et  $K = 20$  (b).

N	K	expérimentale		théorique	
		$a$	$b$	$a = 1 - \alpha_d$	$b = \beta$
20	2	0.826 <sub>0.011</sub>	0.501 <sub>0.023</sub>	0.850	0.5
25	5	0.757 <sub>0.005</sub>	0.500 <sub>0.007</sub>	0.760	0.5
25	20	0.160 <sub>0.004</sub>	0.499 <sub>0.001</sub>	0.160	0.5
64	10	0.828 <sub>0.001</sub>	0.499 <sub>0.001</sub>	0.828	0.5

TAB. 2.1 – Moyenne et écart-type sur 300 instances des paramètres de la droite moyenne du NA  $\tilde{E}(\varphi) = b + a(\varphi - b)$  pour les paysages NK avec  $d = 1$ .

### Application aux paysages MAX-k-SAT

Pour la famille des paysages MAX-k-SAT, le nombre de fonctions embarquées est  $P = m$ ,  $\rho$  est égal à  $\frac{-1}{2^k-1}$  (voir section 2.2.1) et la loi  $X$  est la loi de Bernoulli de paramètre  $1 - 2^{-k}$  d'où  $E(X) = 1 - 2^{-k}$  et  $\sigma_X^2 = (1 - 2^{-k})2^{-k}$ . Les variables dans chaque clause sont déterminées aléatoirement de manière uniforme parmi l'ensemble des variables, les nombres  $m_j$  sont choisis indépendamment, la probabilité  $\gamma^d(n)$  se calcule de la même manière que dans le cas des paysages NK :  $\alpha_d = 1 - \frac{\binom{N-d}{k}}{\binom{N}{k}}$  d'où  $\gamma^d(n) = \binom{m}{n} \alpha_d^n (1 - \alpha_d)^{m-n}$  et  $E(n_d) = m\alpha_d$ .

On obtient les paramètres de la densité binormale  $p_{F^d|F,N}(\tilde{\varphi}|\varphi, n)$  :

$$\mu_n(\varphi) = m(1 - 2^{-k}) + \left(1 - \frac{n}{m(1 - 2^{-k})}\right)(\varphi - m(1 - 2^{-k})) \quad (2.34)$$

et

$$\sigma_n^2 = n\left(2 - \frac{n}{m}2^{-k}\right)(1 - 2^{-k})2^{-2k} \quad (2.35)$$

Pour le paysage MAX-k-SAT, l'équation 2.4 de la courbe moyenne devient :

$$\tilde{E}_d(\varphi) = m(1 - 2^{-k}) + \left(1 - \frac{\alpha_d}{1 - 2^{-k}}\right)(\varphi - m(1 - 2^{-k})) \quad (2.36)$$



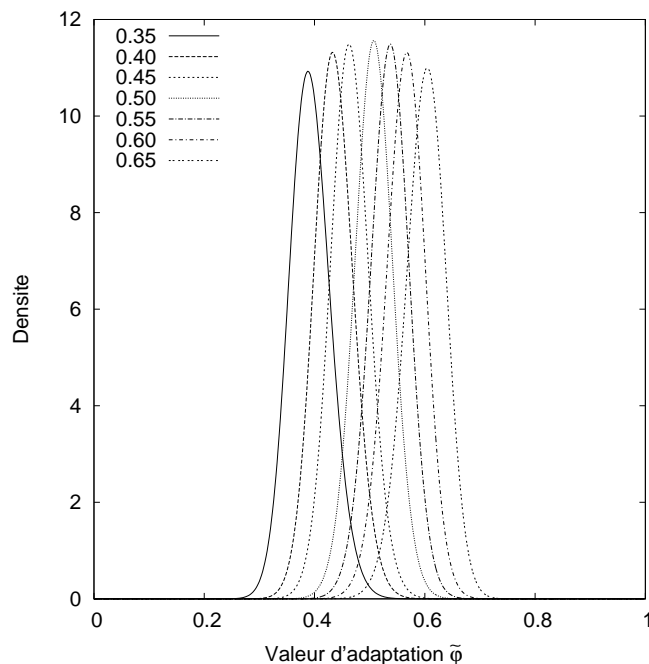


FIG. 2.6 – Densités conditionnelles théoriques pour différentes valeurs de  $\varphi$  pour un paysage NK avec  $N = 32$ ,  $K = 8$  et  $d = 1$ .

Lorsque  $d = 1$ , l'équation 2.36 devient :

$$\tilde{E}(\varphi) = m(1 - 2^{-k}) + \left(1 - \frac{k}{N(1 - 2^{-k})}\right)(\varphi - m(1 - 2^{-k})) \quad (2.37)$$

Remarquons que la pente est indépendante du nombre de clauses  $m$  et donc du seuil critique de transition de phase  $\alpha_c$ . La taille de l'ensemble des solutions au problème SAT est donc indépendante de la corrélation entre solutions voisines. L'équation de la courbe moyenne est confirmée expérimentalement. Pour chaque valeur des paramètres, 100 instances de paysages MAX-3-SAT de la bibliothèque SATLIB<sup>14</sup> sont utilisées, afin d'évaluer les paramètres de la courbe moyenne. L'estimation de chaque NA s'effectue à l'aide de  $10^5$  solutions choisies aléatoirement de manière uniforme. Le tableau 2.2 donne les paramètres de la courbe moyenne déterminée expérimentalement pour différentes valeur de  $N$ ,  $m$  et  $d$ . Pour chaque NA,  $\tilde{E}(\varphi)$  est linéairement corrélé avec un coefficient de corrélation supérieur à 0.99. La figure 2.9 montre différents exemples de contour du NA sur le paysage MAX-SAT.

---

<sup>14</sup>les instances ufN de [www.satlib.org](http://www.satlib.org)

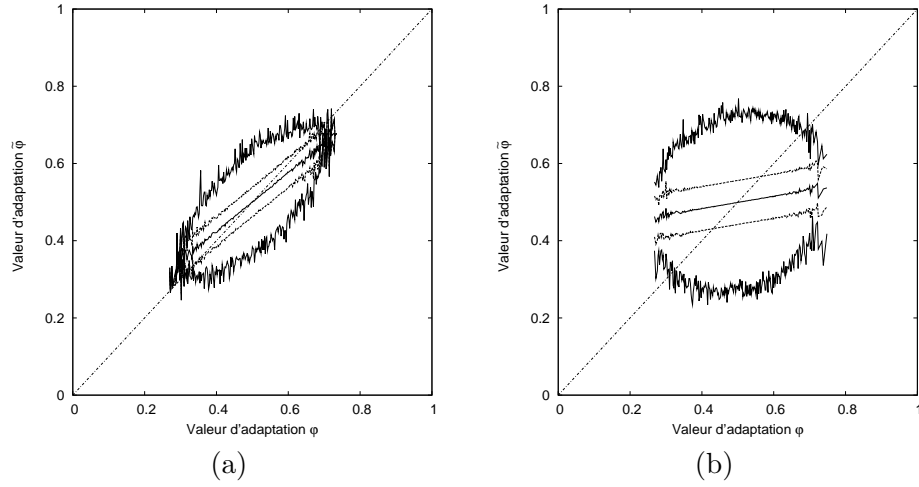


FIG. 2.7 – Contour du NA pour le paysage NK avec  $N = 25$ ,  $K = 5$  (a) et  $K = 20$  (b).

N	m	expérimental		théorique	
		a	b	a	b
20	91	0.8300 <sub>0.0073</sub>	79.5691 <sub>0.5976</sub>	0.829	79.6
50	218	0.9309 <sub>0.0023</sub>	190.5460 <sub>0.3281</sub>	0.931	190.75
100	430	0.9648 <sub>0.0008</sub>	376.0070 <sub>0.5094</sub>	0.9657	376.25
200	860	0.9830 <sub>0.0003</sub>	752.5460 <sub>0.5742</sub>	0.9829	752.5

TAB. 2.2 – Moyenne et écart-type sur 100 instances des paramètres de la droite moyenne du NA  $\tilde{E}(\varphi) = b + a(\varphi - b)$  pour le paysage Max-3-SAT avec  $d = 1$ .

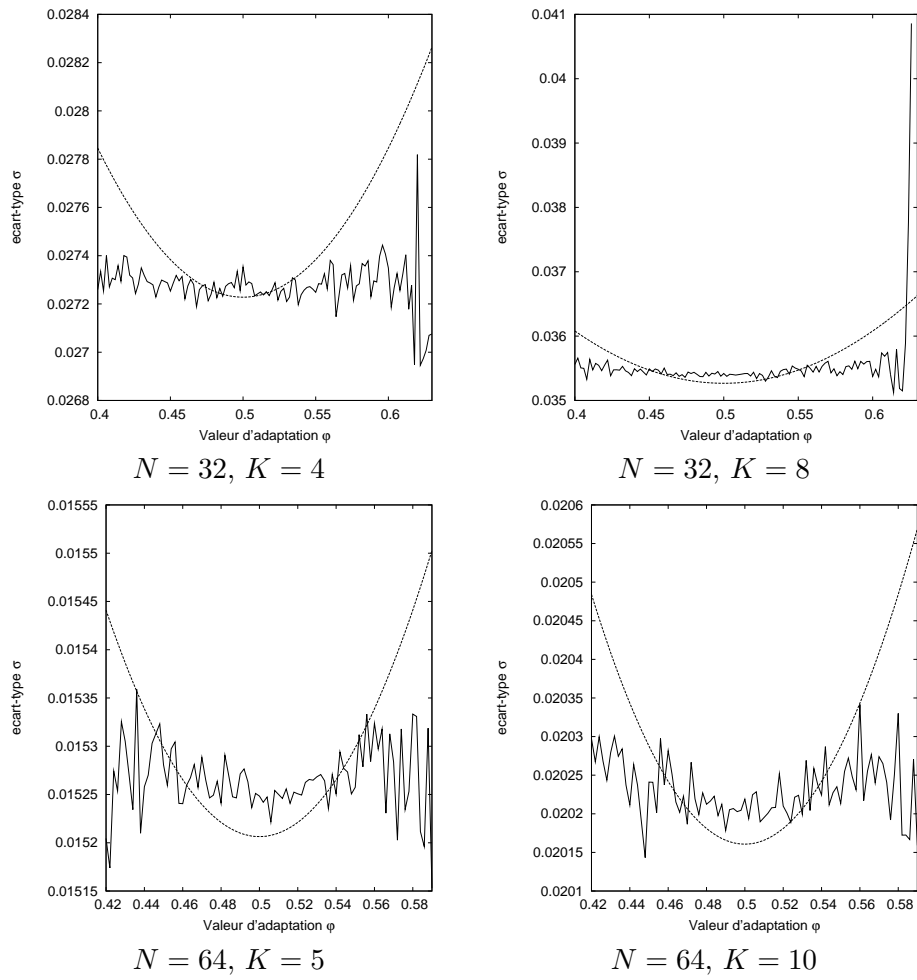


FIG. 2.8 – Écart-types expérimentaux et théoriques pour différents paysages NK.

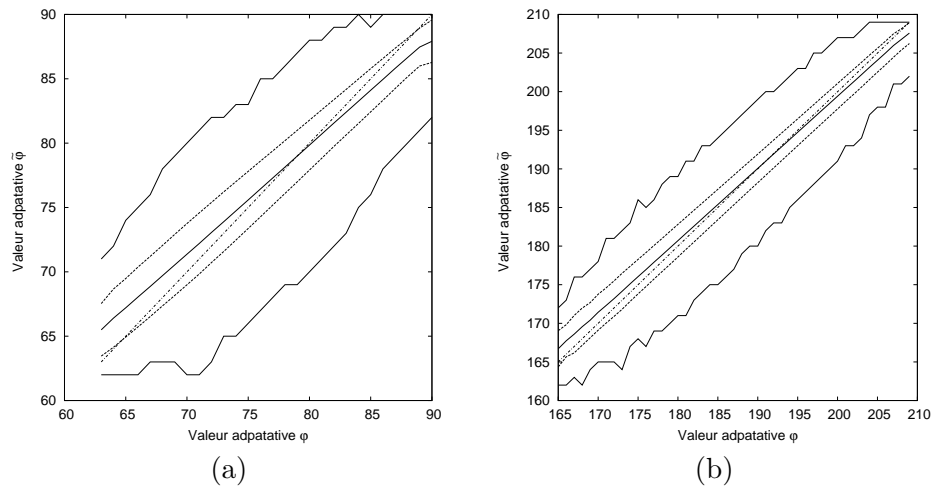


FIG. 2.9 – Contour du NA pour le paysage Max-3-SAT avec  $k = 3, n = 20, m = 91$  (a) et  $n = 50, m = 218$  (b).

N	K	$1 - \frac{K+1}{N}$	a	b	$\rho$
20	2	0.85	0.7890 <sub>0.0333</sub>	0.1696 <sub>0.0195</sub>	0.9986 <sub>0.0007</sub>
25	5	0.76	0.7257 <sub>0.0143</sub>	0.2125 <sub>0.0082</sub>	0.9977 <sub>0.0009</sub>
25	20	0.16	0.1557 <sub>0.0076</sub>	0.5337 <sub>0.0039</sub>	0.9122 <sub>0.0232</sub>
32	4	0.844	0.8094 <sub>0.0130</sub>	0.1533 <sub>0.0078</sub>	0.9988 <sub>0.0004</sub>
32	8	0.719	0.6922 <sub>0.0097</sub>	0.2285 <sub>0.0055</sub>	0.9974 <sub>0.0009</sub>
64	5	0.906	0.8851 <sub>0.0067</sub>	0.0943 <sub>0.0037</sub>	0.9995 <sub>0.0002</sub>
64	10	0.828	0.8079 <sub>0.0063</sub>	0.1443 <sub>0.0034</sub>	0.9989 <sub>0.0004</sub>

TAB. 2.3 – Résultats expérimentaux de la courbe moyenne  $\tilde{E}(\varphi) = a\varphi + b$  relative à l’opérateur HC sur les paysages NK pour différentes valeurs de  $N$  et  $K$ .

## 2.3 Généralisation à d’autres opérateurs

Le nuage adaptatif permet d’étudier la corrélation de performance entre une solution et l’image de cette solution par un opérateur de recherche local. Dans cette section, nous allons étudier le NA relativement à deux opérateurs : l’opérateur local qui sélectionne la solution de meilleure performance dans le voisinage et l’opérateur local utilisé dans la métaheuristique du recuit simulé.

### 2.3.1 Hill-Climbing

Beaucoup d’opérateurs de recherche locale sont des intermédiaires entre l’opérateur de recherche aléatoire et l’opérateur qui sélectionne la solution voisine de meilleure performance, l’opérateur de Hill-Climbing (HC). Les opérateurs locaux performants tentent de réaliser un compromis entre l’exploration du voisinage par l’opérateur RA et l’exploitation du voisinage par l’opérateur HC. Nous allons maintenant étudier ce dernier opérateur du point de vue du nuage adaptatif afin de compléter l’étude du voisinage dans les paysages embarqués uniformes.

Un optimum local est une solution dont toutes les solutions voisines sont de performance inférieure (voir définition section 1.3.1). Par conséquent, la meilleure des solutions voisines a une performance plus faible que la solution optimum local elle-même. Dans la représentation du NA, les optima locaux sont situés en dessous de la première bissectrice (voir figure 2.10). Le NA permet donc d’évaluer et de localiser la performance des optima locaux d’un paysage.

Contrairement à l’opérateur de RA, il n’est pas possible pour le HC d’établir l’expression du NA ou l’équation de la courbe moyenne dans le cas des paysages embarqués uniformes. Nous avons donc réalisé l’estimation du NA pour les paysages NK et MAX-k-SAT de la même manière que dans la section 2.2.2. Pour chaque valeur de paramètres,  $10^5$  solutions sont choisies aléatoirement uniformément dans l’espace de recherche. Pour chaque solution, la meilleure solution voisine de l’ensemble des voisins est sélectionnée.

### Résultats sur les Paysages NK

La figure 2.10 montre des exemples de contour de NA. Les optima locaux sont les solutions de plus grandes performances. La courbe moyenne est encore apparemment proche d’une droite. Pour chacune des 300 instances de paysages NK, nous avons calculé le coefficient de corrélation linéaire ainsi que les paramètres de la droite de régression. Les moyennes et les écart-types de ces valeurs sont reportés dans le tableau 2.3 pour différentes valeurs de  $N$  et  $K$ .

Les coefficients de corrélation sont supérieurs à 0.99 (sauf pour  $N = 25$  et  $K = 20$ ). La courbe moyenne est une droite, la pente de cette droite est plus petite que la pente  $1 - \frac{K+1}{N}$

obtenue avec l'opérateur de RA (voir tableau 2.1).

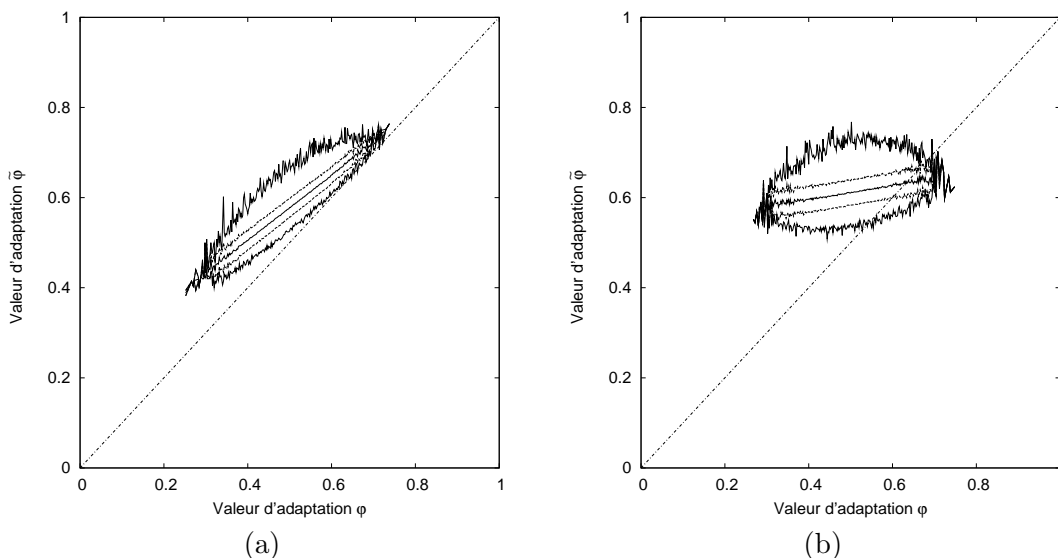


FIG. 2.10 – Contour du NA avec l'opérateur HC pour un paysage NK avec  $N = 25$ ,  $K = 5$  (a) et  $K = 20$  (b).

### Dynamique d'évolution sur le NA

La valeur  $\beta$ , intersection de la courbe moyenne et de la première bissectrice (voir définition 2.1), semble être une *barrière de performance*, *i.e.* un majorant du point de convergence de l'itération de l'opérateur de recherche HC à partir d'une solution initiale uniformément choisie dans l'espace de recherche.

Afin de valider cette conjecture, nous avons effectué  $10^3$  exécutions consistant en  $N$  itérations de l'opérateur HC sur chaque instance de paysage NK. Si au cours d'une itération, la solution courante est un optima local, sans possibilité d'amélioration par l'opérateur HC, alors les solutions suivantes restent inchangées par l'opérateur. Pour chacune des valeurs des paramètres  $N$  et  $K$ , nous avons calculé la *trajectoire moyenne* qui est la moyenne à chaque itération de la performance. La performance finale est notée  $\beta^*$  et peut-être comparée à la valeur  $\beta$  attendue.

La figure 2.11 montre les trajectoires moyennes et la courbe calculée à partir de la courbe moyenne pour différentes valeurs des paramètres. La valeur prédite par la courbe moyenne du NA est plus grande (sauf pour  $N = 25$  et  $K = 20$ ) que la valeur  $\beta^*$  obtenue par la trajectoire moyenne. Les trajectoires commencent en moyenne à la valeur 0.5 et suivent la trajectoire attendue pendant les premières itérations. La courbe moyenne du NA permet donc une bonne approximation du comportement moyen d'un opérateur HC pendant les premières itérations. La conjecture 2.1 est donc vérifiée lorsque le nombre d'itérations est petit ( $i \leq 5$ ). Par contre, il ne permet d'estimer qu'une borne supérieure à la performance finale obtenue par l'itération de l'opérateur HC. Cette différence peut peut-être être expliquée par l'évolution de la courbe moyenne entre les itérations  $i$  et  $i + 1$  lorsque  $i$  augmente.

### Résultats sur le paysage MAX-SAT

La figure 2.12 montre la moyenne des contours de NA. Les optima locaux sont les solutions de plus grande performance. La courbe moyenne est encore apparemment proche d'une droite.

TAB. 2.4 – Valeur moyenne et écart-type du point d’intersection  $\beta$  du NA comparée avec la valeur moyenne du point de convergence  $\beta^*$  de l’itération de l’opérateur HC pour les paysages NK.

N	K	$\beta$	$\beta^*$
20	2	0.810347 <sub>0.0518808</sub>	0.7116 <sub>0.0408044</sub>
25	5	0.775204 <sub>0.0148091</sub>	0.712494 <sub>0.0339714</sub>
25	20	0.632122 <sub>0.00145688</sub>	0.645753 <sub>0.0278156</sub>
32	4	0.805528 <sub>0.0191376</sub>	0.716038 <sub>0.0302279</sub>
32	8	0.742707 <sub>0.00637298</sub>	0.701227 <sub>0.0288993</sub>
64	5	0.822391 <sub>0.0167046</sub>	0.717881 <sub>0.0217064</sub>
64	10	0.751317 <sub>0.00741334</sub>	0.699374 <sub>0.0208754</sub>

TAB. 2.5 – Résultats expérimentaux de la courbe moyenne  $\tilde{E}(\varphi) = a\varphi + b$  relative à l’opérateur HC sur les paysages MAX-SAT pour différentes valeurs de  $N$  et  $m$ .

N	m	$a_{RA}$	a	b	$\rho$
20	91	0.829	0.75408 <sub>0.0279096</sub>	22.7226 <sub>2.36714</sub>	0.998557 <sub>0.00134519</sub>
50	218	0.931	0.887027 <sub>0.0119675</sub>	25.5441 <sub>2.37327</sub>	0.999659 <sub>0.000166875</sub>
100	430	0.9657	0.939744 <sub>0.00533962</sub>	27.214 <sub>2.08028</sub>	0.999847 <sub>9.45114e-05</sub>
200	860	0.9829	0.968815 <sub>0.00252211</sub>	28.5838 <sub>1.93409</sub>	0.999934 <sub>3.0739e-05</sub>

Pour chacune des 100 instances de paysages MAX-SAT, nous avons calculé le coefficient de corrélation linéaire ainsi que les paramètres de la droite de régression de la courbe moyenne. Les moyennes et les écart-types de ces valeurs sont reportés dans le tableau 2.5 pour différentes valeurs de  $N$  et  $m$ . Les coefficients de corrélation sont supérieures à 0.99, la courbe moyenne est donc une droite dont la pente est plus faible que celle obtenue dans le cas de l’opérateur de recherche aléatoire (noté  $a_{RA}$  dans le tableau).

**Dynamique d’évolution sur le NA** Nous avons réalisé la même série d’expériences que pour les paysages NK afin de vérifier la conjecture 2.1.

La figure 2.13 montre les trajectoires moyennes et la courbe calculée à partir de la courbe moyenne pour différentes valeurs des paramètres. La table 2.6 donne les valeurs de  $\beta$  et  $\beta^*$ . La valeur prédite par la courbe moyenne du NA est plus grande que la valeur  $\beta^*$  obtenue par la trajectoire moyenne. Les trajectoires commencent autour de la valeur  $\frac{7m}{8}$  et suivent la trajectoire attendue pendant les premières itérations. Les conclusions sont les mêmes que pour les paysages NK, la courbe moyenne du NA permet une bonne approximation du comportement moyen d’un opérateur HC pendant les premières itérations. La conjecture 2.1 est donc vérifiée lorsque le nombre d’itérations est petit ( $i \leq 4$ ). Par contre, il ne permet d’estimer qu’une borne supérieure à la performance finale obtenue par l’itération de l’opérateur HC.

TAB. 2.6 – Valeur moyenne et écart-type du point d’intersection  $\beta$  du NA comparée avec la valeur moyenne du point de convergence  $\beta^*$  de l’itération de l’opérateur HC pour les paysages MAX-3-SAT.

N	m	$\beta$	$\beta^*$
20	91	89.7 <sub>10.6</sub>	88.44
50	218	227.3 <sub>2.9</sub>	212.5
100	430	454.4 <sub>4.7</sub>	419.2
200	860	917.7 <sub>9.8</sub>	838.6

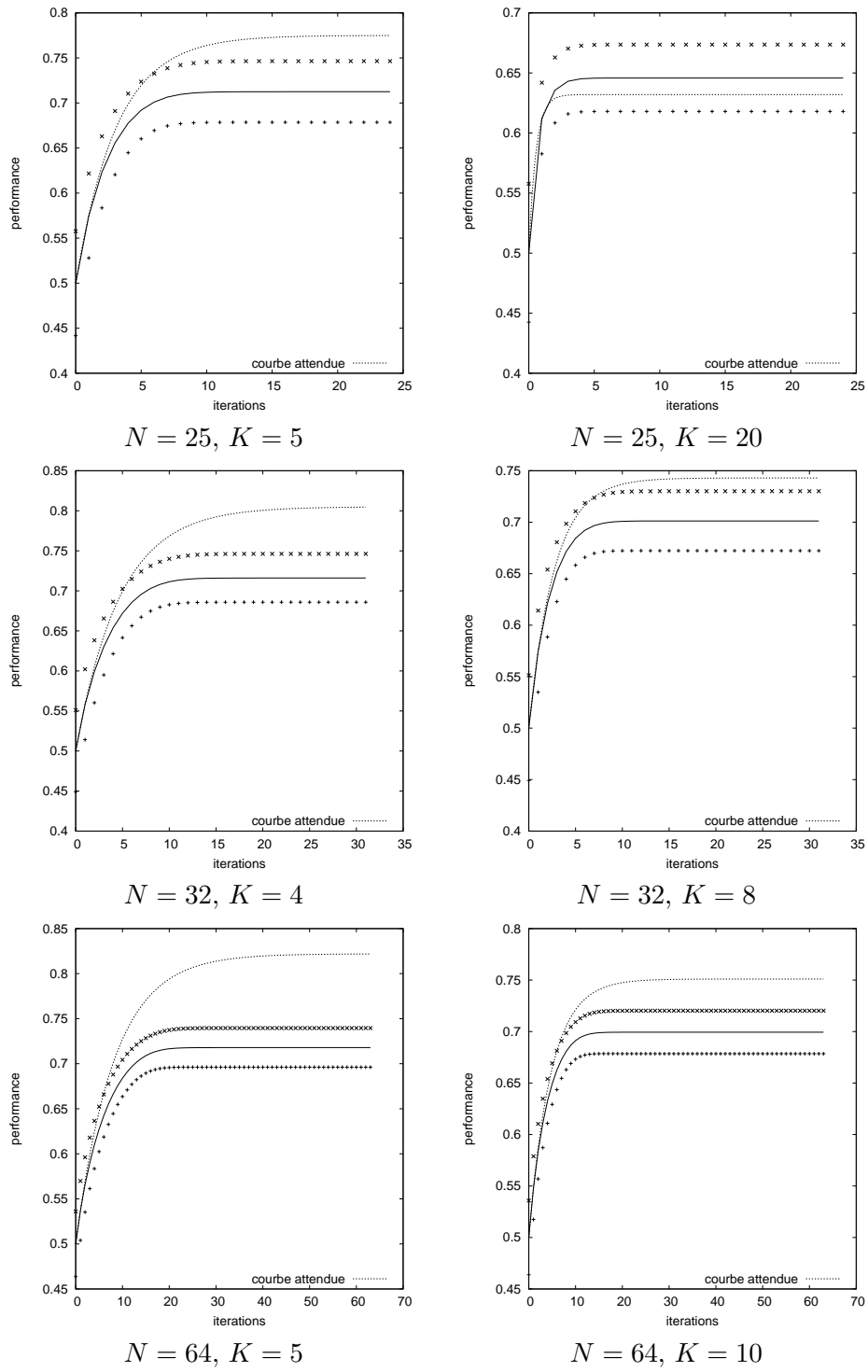


FIG. 2.11 – Trajectoire moyenne (avec l'écart-type) de l'itération de l'opérateur HC pour différents paysages NK et courbe calculée à partir de la courbe moyenne du NA.



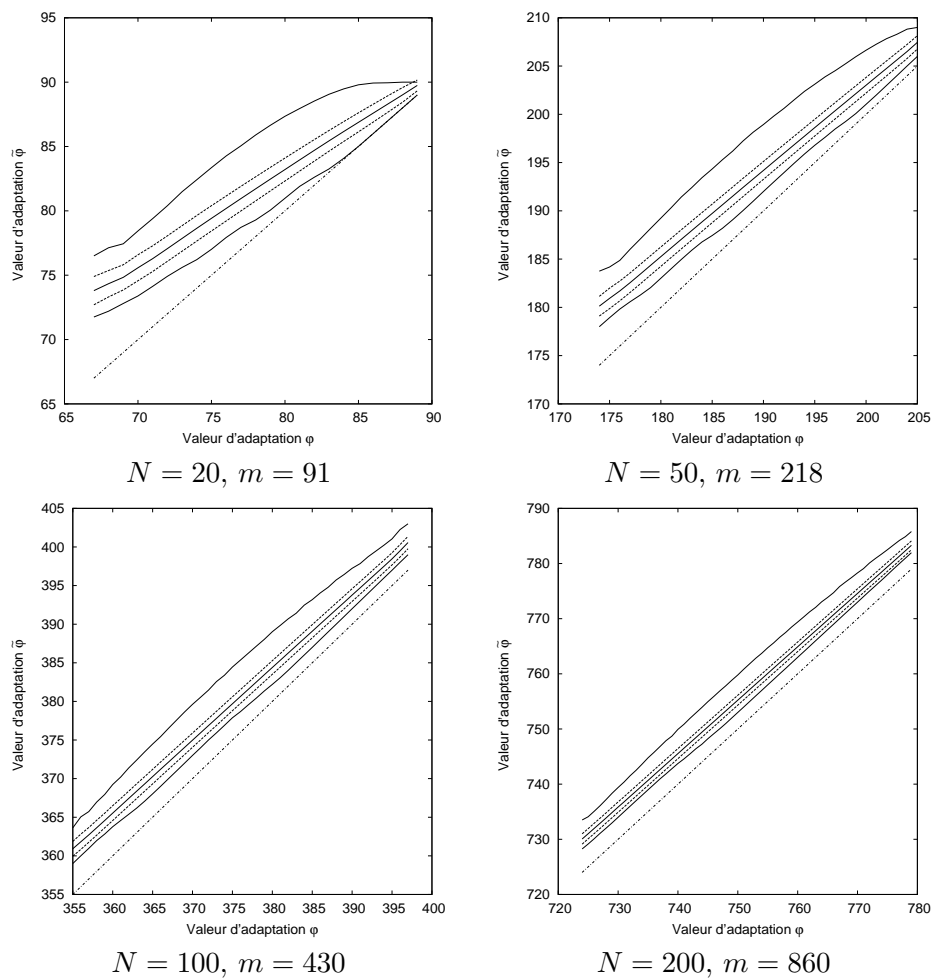


FIG. 2.12 – Contour du NA avec l'opérateur HC pour les paysages MAX-SAT.

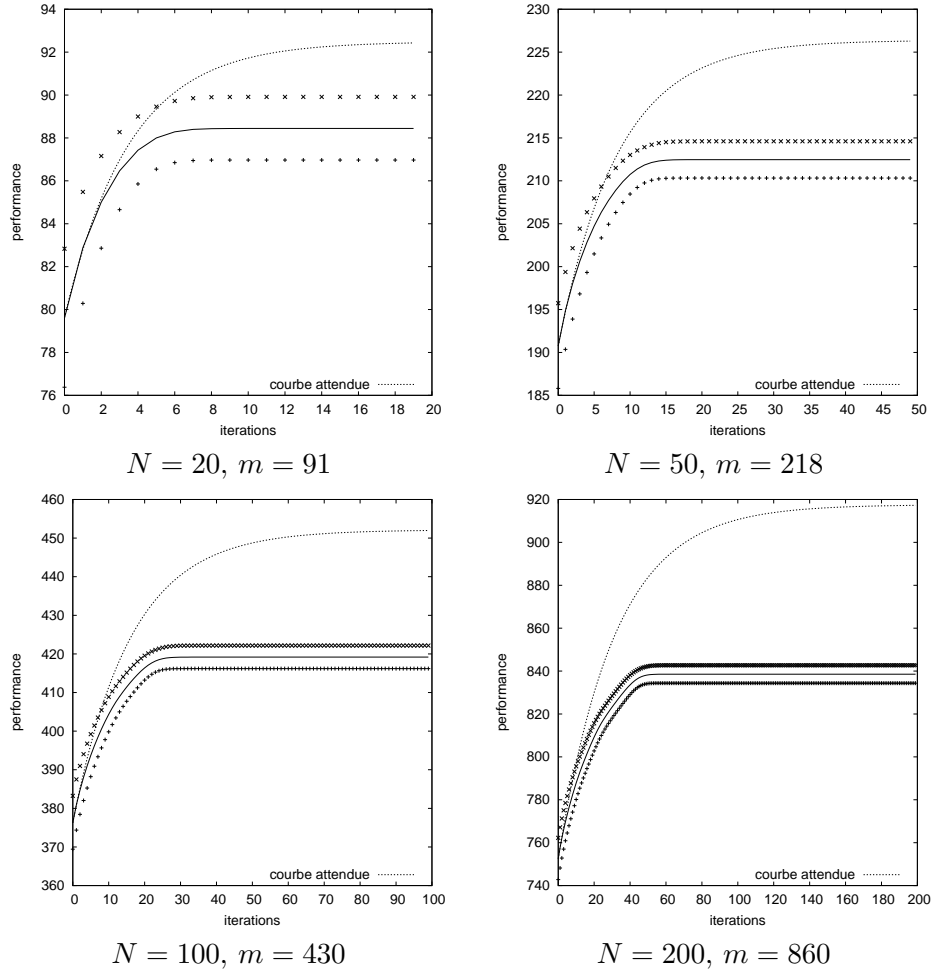


FIG. 2.13 – Trajectoire moyenne (avec l'écart-type) de l'itération de l'opérateur HC pour différents paysages MAX-3-SAT et courbe calculée à partir de la courbe moyenne du NA.

### 2.3.2 Recuit Simulé

Dans cette section, nous étudions le nuage adaptatif relatif à l'opérateur local du recuit simulé. Le recuit simulé est une métaheuristique qui permet d'éviter les optima locaux. l'opérateur local, noté  $op^{RS}$ , du recuit simulé est défini à partir de l'opérateur local  $op^{RS}$  de recherche aléatoire et d'un paramètre d'acceptation  $T$  de dégradation de performance, assimilé à une température. Plus précisément,

$$op^{RS}(s) = \begin{cases} s' & \text{si } \Delta \geq 0 \text{ ou } u < \exp(\frac{\Delta}{T}) \\ \text{avec } s' = op^d(s), \Delta = f(s') - f(s) \text{ et } u \text{ nombre aléatoire de } [0, 1], & \\ s & \text{sinon.} \end{cases}$$

A partir d'une valeur initiale du paramètre  $T$ , le recuit simulé consiste à itérer l'opérateur local  $op^{RS}$  tout en modifiant le paramètre  $T$  suivant une loi de décroissance. Pour une description plus complète de la méthode du recuit simulé, on pourra consulter par exemple l'ouvrage de P. Siarry *et al* [117].

L'expression analytique du NA relative à l'opérateur  $op_T^{RS}$  et à la température  $T$  peut s'obtenir à partir de celle obtenue relativement à l'opérateur de recherche aléatoire  $op^d$  :

$$p_T^{RS}(\tilde{\varphi}|\varphi) = \begin{cases} p^{op^d}(\tilde{\varphi}|\varphi) & \text{si } \tilde{\varphi} > \varphi, \\ \int_{-\infty}^{\varphi} p^{op^d}(f|\varphi) (1 - \exp(\frac{f-\varphi}{T}))df & \text{si } \tilde{\varphi} = \varphi, \\ p^{op^d}(\tilde{\varphi}|\varphi) \exp(\frac{\tilde{\varphi}-\varphi}{T}) & \text{si } \tilde{\varphi} < \varphi. \end{cases} \quad (2.38)$$

Pour les valeurs de performances strictement au-dessus de la première bissectrice, les distributions  $p^{RS}$  et  $p^{RA}$  sont identiques; pour les valeurs de performances strictement au-dessous de la première bissectrice, la distribution  $p^{RS}$  est proportionnelle à la distribution  $p^{RA}$  d'un facteur  $\exp(\frac{\tilde{\varphi}-\varphi}{T})$  strictement plus petit que 1; enfin, sur la bissectrice, la distribution  $p^{RS}$  est supérieure à la distribution  $p^{RA}$ .

De même que dans le cas de l'opérateur HC et suivant le même protocole expérimental, nous avons réalisé l'estimation du NA pour les paysages NK et MAX-k-SAT.

#### Résultats sur les Paysages NK

La figure 2.14 montre des exemples de contour de NA relativement à l'opérateur RS. La courbe moyenne n'est plus une droite. Pour les faibles performances, les courbes moyennes  $\tilde{E}(\varphi)$  relatives à l'opérateur RS sont proches des courbes moyennes relatives à l'opérateur de recherche aléatoire RA (voir la sous-section 2.2.2). Pour les hautes performances, les courbes moyennes se confondent avec la première bissectrice et ceci d'autant plus vite que la température est faible.

**Dynamique d'évolution sur le NA** La figure 2.15 montre les trajectoires moyennes et la courbe calculée à partir de la courbe moyenne pour différentes valeurs des paramètres. Lors des premières itérations, les deux courbes sont confondues. La conjecture 2.1 est donc vérifiée lorsque le nombre d'itérations est petit. Pour les itérations suivantes, l'accroissement de la trajectoire moyenne est plus petite que celle de la courbe calculée à partir de la courbe moyenne du NA. Pour  $K = 4$ , Le point de convergence de la trajectoire moyenne est au-dessus du point de convergence obtenue à partir de la courbe moyenne. Pour  $K = 8$ , pour la température  $T = 0.1$ , les points de convergence des deux courbes sont égales; pour la température  $T = 0.05$ , le point de convergence de la trajectoire moyenne est au-dessus de celui de la courbe moyenne; pour la

TAB. 2.7 – Résultats expérimentaux (moyenne et écart-type sur 100 instances de paysage) de la droite de régression  $\tilde{E}(\varphi) = a\varphi + b$  relative à l’opérateur RS des paysages MAX-SAT pour différentes valeurs de  $N$  et  $m$  et de température  $T$ .

N	m	T	$\rho$	a	b
20	91	0.75	0.9985 <sub>0.0019</sub>	0.8937 <sub>0.0184</sub>	9.1852 <sub>1.5176</sub>
		1.3	0.9984 <sub>0.0018</sub>	0.8864 <sub>0.0170</sub>	9.6744 <sub>1.3914</sub>
		2.6	0.9986 <sub>0.0013</sub>	0.8719 <sub>0.0146</sub>	10.6683 <sub>1.1822</sub>
50	218	0.75	0.9995 <sub>0.0004</sub>	0.9569 <sub>0.0077</sub>	8.8819 <sub>1.5037</sub>
		1.3	0.9994 <sub>0.0006</sub>	0.9547 <sub>0.0081</sub>	9.2069 <sub>1.5756</sub>
		2.6	0.9993 <sub>0.0005</sub>	0.9482 <sub>0.0098</sub>	10.2894 <sub>1.8942</sub>
100	430	0.75	0.9998 <sub>0.0002</sub>	0.9781 <sub>0.0047</sub>	8.8545 <sub>1.8107</sub>
		1.3	0.9998 <sub>0.0002</sub>	0.9766 <sub>0.0048</sub>	9.3508 <sub>1.8440</sub>
		2.6	0.9996 <sub>0.0003</sub>	0.9736 <sub>0.0058</sub>	10.3207 <sub>2.2204</sub>
200	860	0.75	0.9999 <sub>0.0001</sub>	0.9896 <sub>0.0028</sub>	8.4608 <sub>2.1479</sub>
		1.3	0.9998 <sub>0.0001</sub>	0.9885 <sub>0.0029</sub>	9.2256 <sub>2.2346</sub>
		2.6	0.9998 <sub>0.0001</sub>	0.9874 <sub>0.0036</sub>	9.8747 <sub>2.7060</sub>

température  $T = 0.01$ , le point de convergence de la trajectoire moyenne est au-dessus de celui de la courbe moyenne.

Il n’est pas possible de prédire le point de convergence de la trajectoire moyenne à partir de la courbe moyenne du NA pour l’opérateur lié au RS. Il est possible que la courbe moyenne soit trop proche de la première bissectrice pour permettre cette prédiction.

### Résultats sur les Paysages MAX-SAT

La figure 2.16 montre des exemples de contour de NA. Les courbes moyennes semble être des droites. Le tableau 2.7 donne le résultat de la régression linéaire de la courbe moyenne. Les coefficients de corrélation sont supérieurs à 0.99, les courbes moyennes peuvent donc être considérées comme des droites. Les pentes diminuent et les ordonnées à l’origine augmentent avec l’augmentation de la température. Pour les basses températures et les hautes valeurs de performances, les courbes moyennes sont confondues avec la première bissectrice.

**Dynamique d’évolution sur le NA** La figure 2.17 montre, pour différentes valeurs des paramètres, les trajectoires moyennes et la courbe calculée à partir des droites de régression des courbes moyennes du tableau 2.7. Pour les premières itérations, la courbe estimée est très proche de la trajectoire moyenne. La conjecture 2.1 est donc vérifiée lorsque le nombre d’itérations est petit. Pour  $N = 50$ , le point de convergence de la courbe estimée est au dessus pour  $T = 0.75$ , égale pour  $T = 1.3$  et au-dessus pour  $T = 2.6$  du point de convergence de l’itération de l’opérateur local RS. Pour  $N = 100$ , le point de convergence de la courbe estimée est au-dessus du point de convergence de l’itération de l’opérateur local RS. Le point de convergence de la courbe estimée n’est donc ni un majorant ni un minorant du point de convergence de l’itération de l’opérateur local RS.

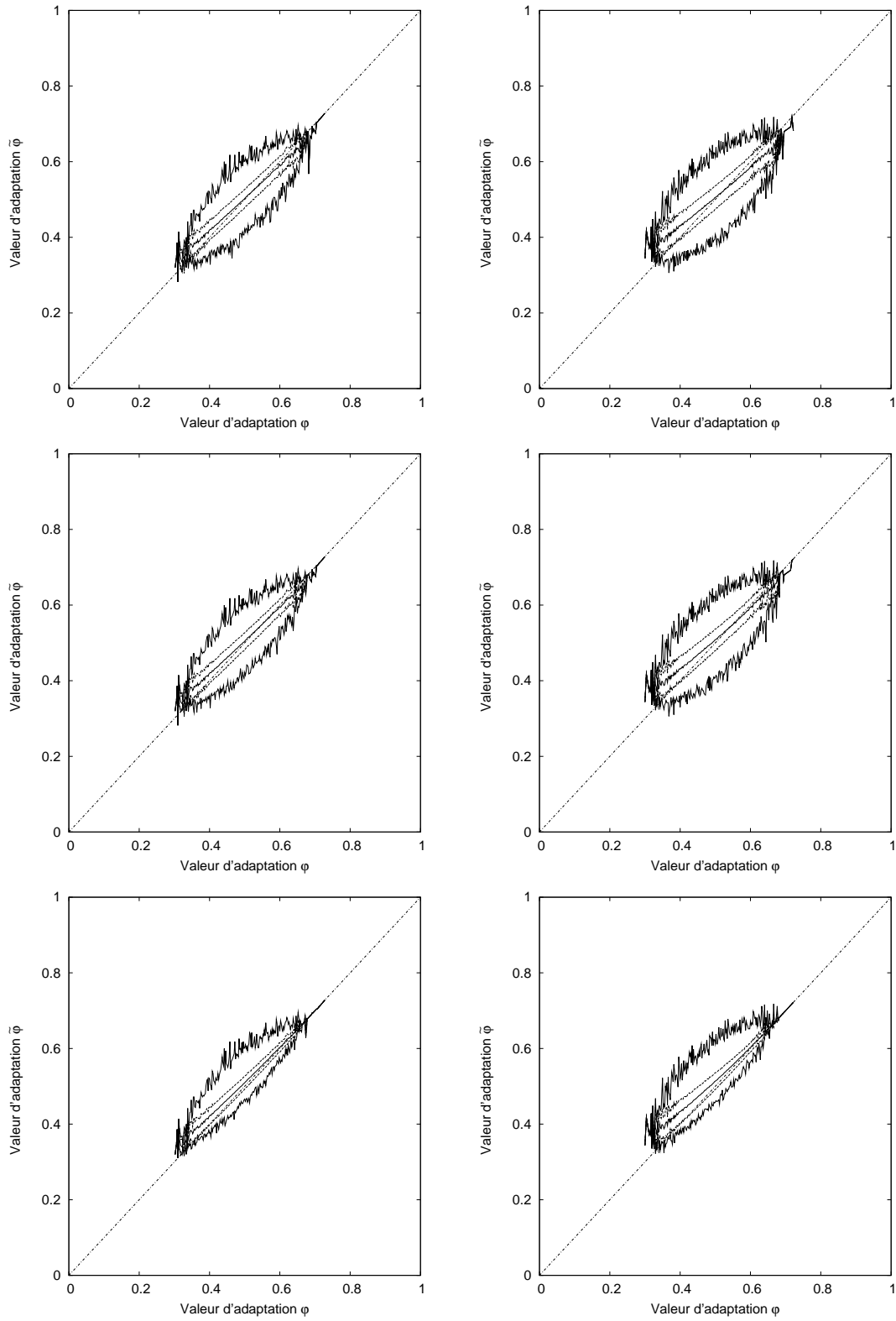


FIG. 2.14 – Contour du NA relativement à l'opérateur RS aux trois températures  $T = 0.1$ ,  $T = 0.05$  et  $T = 0.01$  (de haut en bas) pour le paysage NK avec  $N = 32$ ,  $K = 4$  (colonne de gauche) et  $K = 8$  (colonne de droite).

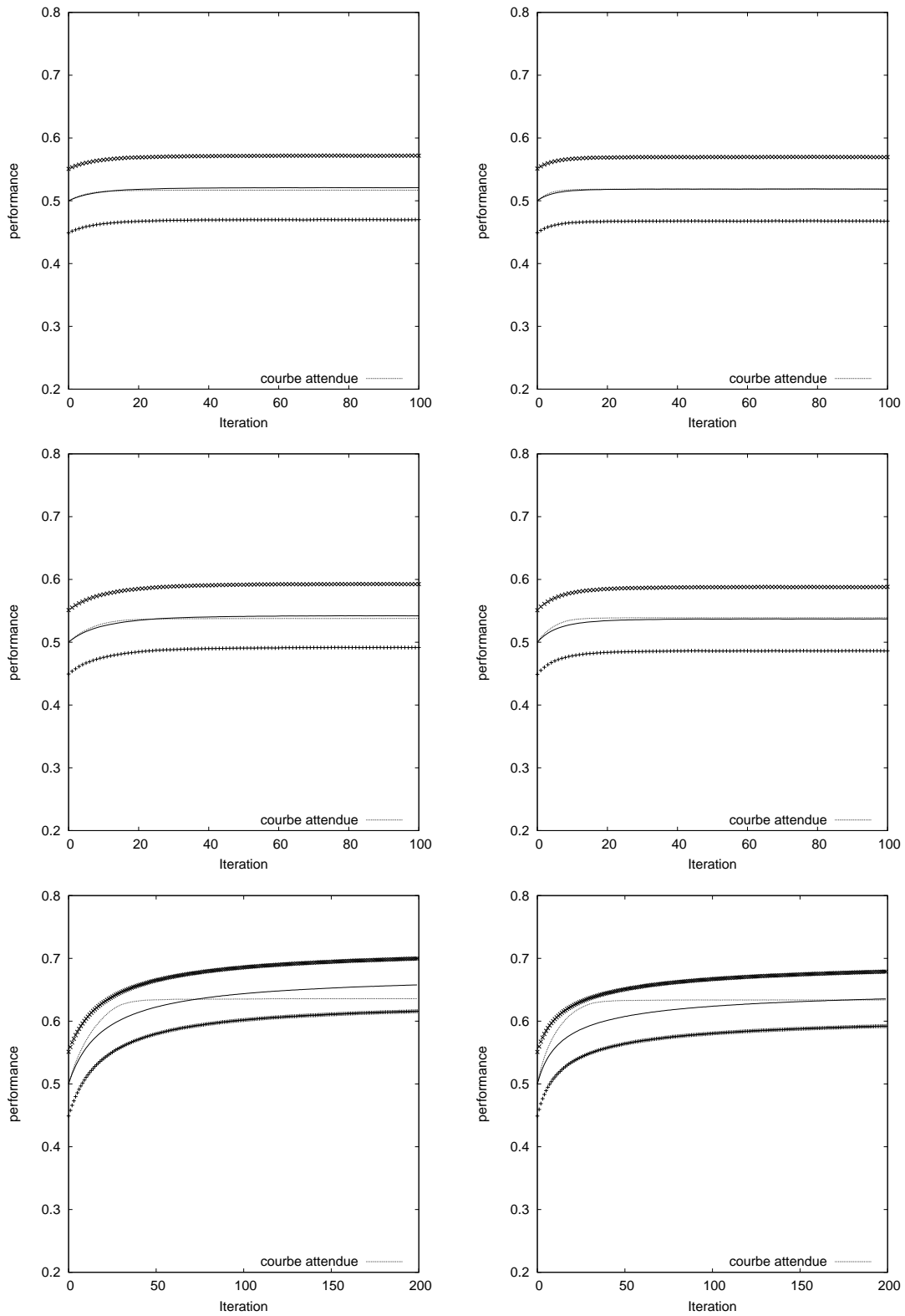


FIG. 2.15 – Trajectoire moyenne (avec l'écart-type) de l'itération de l'opérateur RS et courbe attendue grâce à la courbe moyenne du NA aux trois températures  $T = 0.1$ ,  $T = 0.05$  et  $T = 0.01$  (de haut en bas) pour le paysage NK avec  $N = 32$ ,  $K = 4$  (colonne de gauche) et  $K = 8$  (colonne de droite).

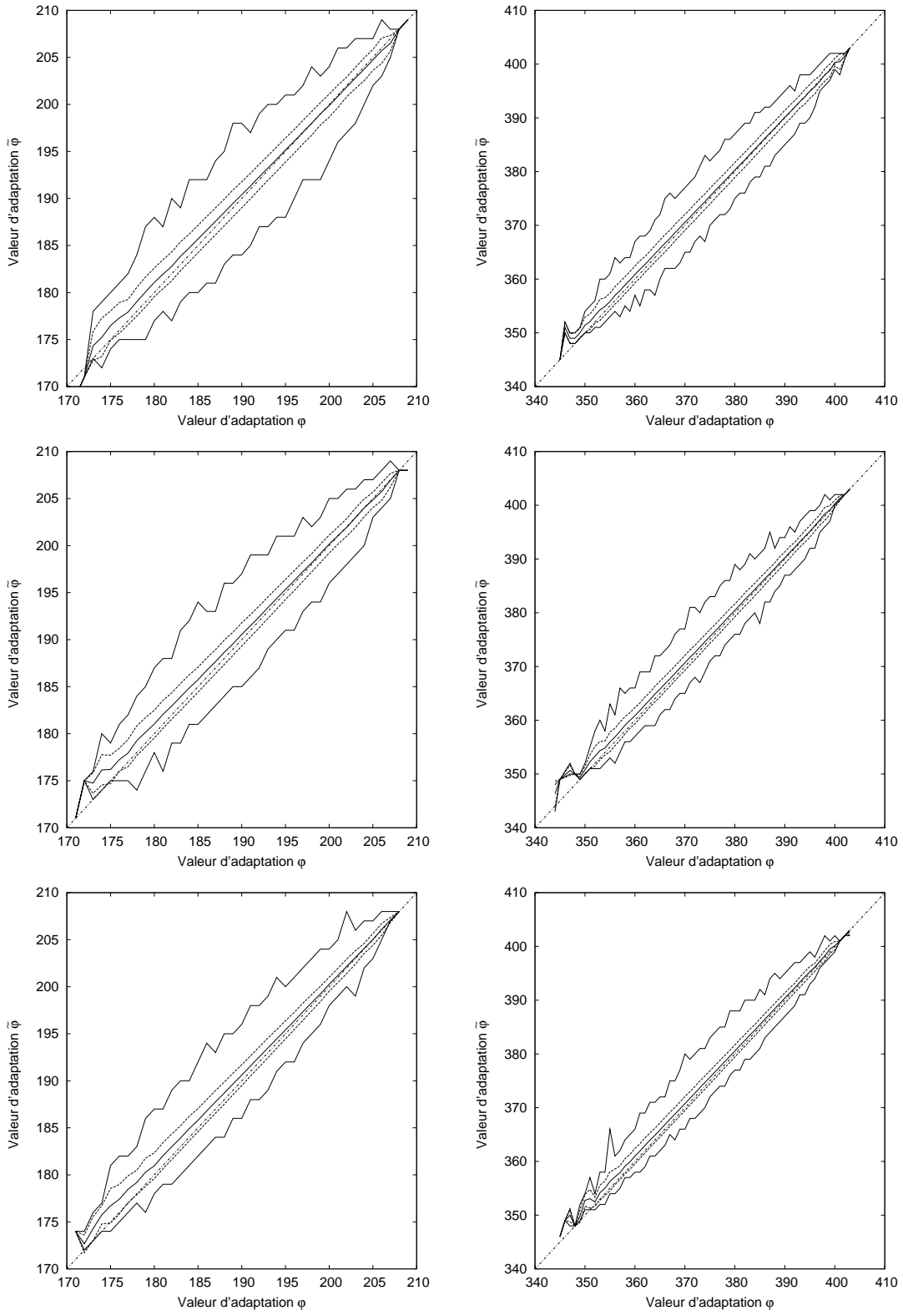


FIG. 2.16 – Contour du NA relativement à l’opérateur RS aux trois températures  $T = 2.6$ ,  $T = 1.3$  et  $T = 0.75$  (de haut en bas) pour le paysage Max-3-SAT avec  $N = 50$ ,  $m = 218$  (colonne de gauche) et  $N = 100$ ,  $m = 430$  (colonne de droite).

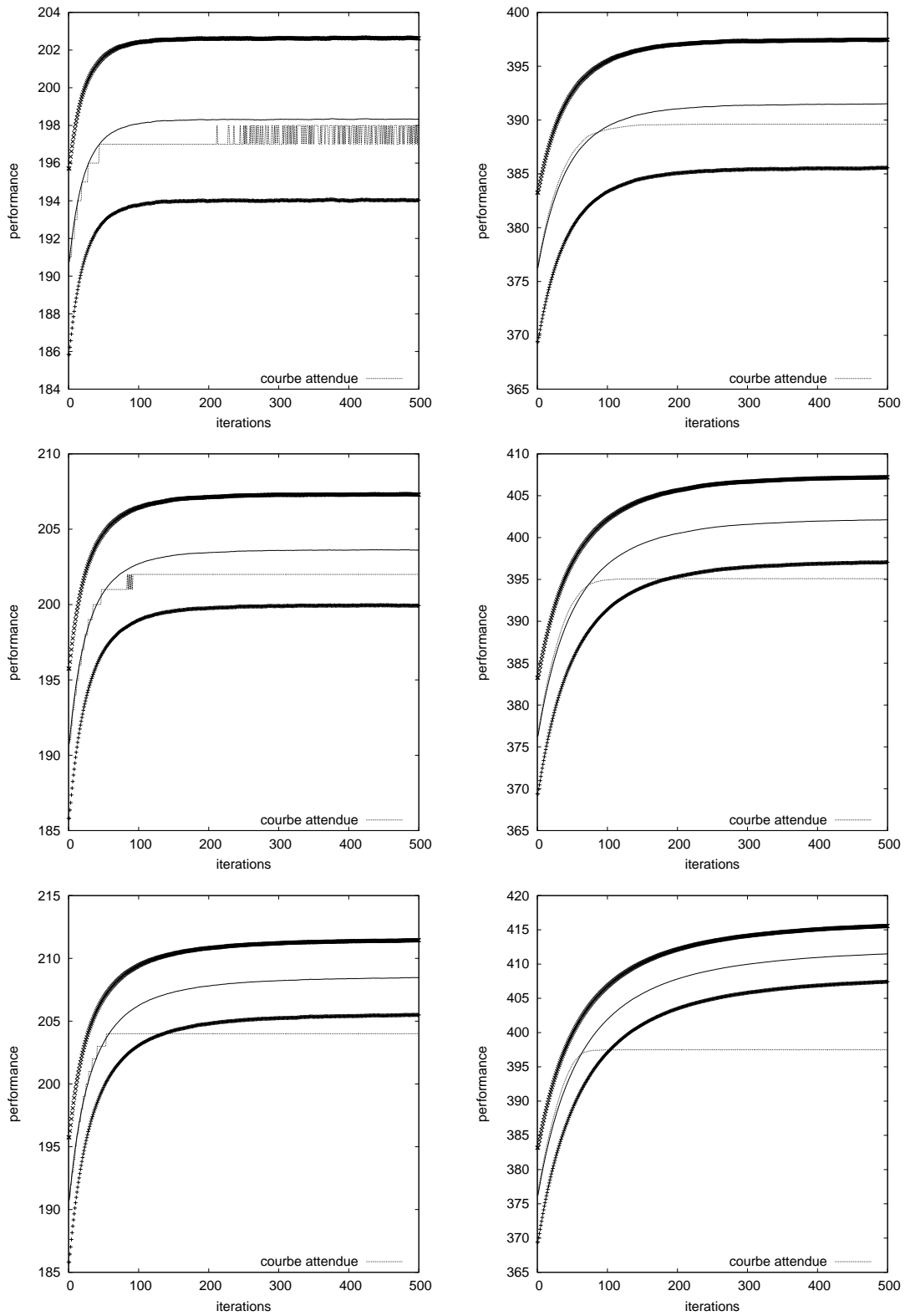


FIG. 2.17 – Trajectoire moyenne (avec l'écart-type) de l'itération de l'opérateur RS et courbe attendue grâce à la courbe moyenne du NA aux trois températures  $T = 2.6$ ,  $T = 1.3$  et  $T = 0.75$  (de haut en bas) pour le paysage Max-3-SAT avec  $N = 50$ ,  $m = 218$  (colonne de gauche) et  $N = 100$ ,  $m = 430$  (colonne de droite).



## 2.4 Coefficient de pente négative

Le nuage adaptatif permet d’estimer la valeur moyenne de la performance d’une solution après avoir appliqué un opérateur de recherche local et de connaître le comportement moyen de l’itération de cet opérateur à court terme. Toutefois, le nuage adaptatif présente plusieurs défauts que nous exposons dans la suite, et auxquels nous apportons plusieurs solutions.

### 2.4.1 Avantages / Inconvénients du NA

Le premier inconvénient résulte de la méthode d’échantillonnage de l’espace de recherche. Dans les expériences précédentes, nous avons utilisé l’échantillonnage uniforme de l’espace. De même, l’expression analytique du NA sur les paysages embarqués uniformes est donnée pour une solution uniformément choisie dans l’espace de recherche. Or, les heuristiques visitent plus souvent les solutions de haute performance que les solutions de performance moyenne. Cette méthode d’échantillonnage est d’autant moins adaptée lorsque l’espace de recherche est grand. Le NA pourrait être plus pertinent en échantillonnant l’espace avec des solutions de plus grande performance.

Nous avons vu une majorité d’exemples de paysages adaptatifs où la courbe moyenne du NA est une droite. Dans ce cas, il est aisé d’analyser le NA à l’aide du nombre  $\beta$  et d’interpréter la relation entre les performances de solutions voisines. Les statistiques que nous avons utilisées ne sont plus aussi adaptées lorsque la courbe moyenne n’est plus une droite sur l’ensemble du paysage.

Le dernier problème concerne le lien entre le NA et la difficulté à optimiser une instance d’un problème par une recherche locale. Le NA nous a permis de prévoir l’évolution de la performance en appliquant un opérateur local donné, par contre, il n’informe pas directement sur la difficulté d’optimisation à l’aide de cet opérateur.

Afin de répondre à ces insuffisances, nous proposons une nouvelle méthode d’estimation du NA et une statistique déduite du NA mesurant la difficulté d’optimisation par une recherche locale particulièrement adaptée aux grands espaces de recherche hétérogène, comme ceux rencontrés en programmation génétique.

### 2.4.2 Définition

Dans cette section, nous présentons une mesure de difficulté originale appelée *Coefficient de Pente Négative* (CPN) introduite dans l’article [133]. Cette mesure repose sur le nuage adaptatif dont l’échantillonnage et l’opérateur local sont particuliers.

Tout d’abord, La définition du CPN est donnée à partir d’un nuage adaptatif échantillonné suivant la méthode de Métropolis-Hasting [82] afin de mieux prendre en compte les solutions de grande performance. La méthode d’échantillonnage pour obtenir un échantillon de taille  $n$  est présentée dans l’algorithme 2.

En PG, le voisinage des solutions est de grande taille, il est donc nécessaire de choisir un opérateur de recherche locale adapté à ce voisinage. Intuitivement, le voisinage d’une solution est peu adapté à la recherche locale lorsque les “bonnes” solutions sont trop rares dans le voisinage. L’opérateur local utilisé pour définir le CPN tente donc d’échantillonner les solutions de grandes performances du voisinage. Pour cela, l’opérateur réalise un tournoi de taille  $t$  avec les solutions voisines. Plus précisément,  $op(s)$  est la solution de plus grande performance parmi les  $t$  solutions obtenues par l’opérateur de recherche aléatoire  $op^d : s_1 = op^d(s), \dots, s_t = op^d(s)$ .

Pour définir le CPN, considérons un paysage  $(\mathcal{S}, \mathcal{V}, f)$  et le nuage adaptatif, noté  $NA'$ , décrit comme précédemment. Soit une partition  $I$  de  $f(\mathcal{S})$  en  $m$  segments de même longueur  $I = \{I_1, I_2, \dots, I_m\}$ . Nous pouvons donc définir les  $m$  points moyens  $M_i$  d’abscisse  $x_i = E_{s \in I_i}(f(s))$

---

**Algorithme 2** Échantillonnage de Métropolis-Hasting

---

Choisir solution initiale  $s \in \mathcal{S}$

$k \leftarrow 1$

**tant que**  $k < n$  **faire**

**répéter**

    Choisir  $s' \in \mathcal{V}(s)$  aléatoirement

    Choisir  $u$  un nombre aléatoire suivant une loi uniforme  $\mathcal{U}(0, 1)$

**jusqu'à**  $u \leq \min(1, \frac{f(s)}{f(s')})$

$s \leftarrow s'$

$k \leftarrow k + 1$

**fin tant que**

---

et d'ordonnée  $y_i = E_{s \in I_i}(f(op(s)))$ , et les  $m - 1$  pentes  $P_i$  des segments joignant les points moyens  $M_i$  à  $M_{i+1}$  :

$$P_i = \frac{x_{i+1} - x_i}{y_{i+1} - y_i}$$

Finalement, la statistique *coefficient de pente négative* est définie par la somme des pentes négatives  $P_i$  :

$$CPN = \sum_{i=1}^{m-1} \min(P_i, 0)$$

Le CPN n'est pas une mesure pertinente dans les cas des paysages embarqués. En effet, nous avons vu que la courbe moyenne est une droite dans le cas des paysages embarqués, ce qui permet de caractériser la corrélation entre solutions de manière satisfaisante. Les espaces de recherche en programmation génétique (PG) sont de grande taille et le voisinage de chaque solution est important. Par ailleurs, peu de mesure de difficulté existe dans ce domaine ; citons la corrélation performance-distance à un optimum (ou Fitness Distance Correlation (FDC) en anglais) [19, 135, 134]. Le CPN est donc une statistique candidate à la mesure de difficulté en PG. Nous avons choisi de calculer expérimentalement le CPN sur des problèmes tests issus de la programmation génétique et de comparer les résultats obtenus avec la difficulté connue d'optimisation par PG de ces problèmes.

### 2.4.3 Résultats expérimentaux

Nous avons choisi trois problèmes académiques appartenant à des classes importantes de problèmes pour la PG : le problème binomial-3 (problème de régression symbolique), le problème de parité<sup>15</sup> (problème booléen) et enfin un problème de programmation de robot : la fourmi artificielle du Santa Fe. Ces trois problèmes sont des problèmes de minimisation. Pour chaque paysage, l'estimation du nuage adaptatif s'effectue à partir d'un échantillon de  $n = 4.10^4$  solutions. L'opérateur local est le tournoi de taille 10 où l'opérateur de recherche aléatoire est la mutation de sous-arbre standard. Le nombre de segments est  $m = 10$ .

#### Le problème binomial-3

Ce problème de régression symbolique a été proposé par Daida *et al* [26]. Il consiste à approcher la fonction polynôme  $f(x) = (1 + x)^3$ . La performance d'un programme est la somme sur 50 points d'évaluation des valeurs absolues de l'erreur entre la valeur de la fonction et

---

<sup>15</sup>nous traduisons par "problème de parité" le nom anglais "even parity problem"

valeur retournée par le programme. L'ensemble des opérateurs utilisés dans les programmes est  $\mathcal{F} = \{+, -, *, /\}$  où  $//$  est la division protégée qui retourne 1 si le dénominateur est nul et l'ensemble des terminaux  $\mathcal{T} = \{x, \mathcal{R}\}$ , où  $x$  est la variable symbolique et  $\mathcal{R}$  est l'ensemble des constantes aléatoires éphémères (CAE). Les CAE sont les réalisations de la loi uniforme  $\mathcal{U}(-a_{\mathcal{R}}, a_{\mathcal{R}})$ , elles sont générées une seule fois pour l'initialisation de la population et ne changent pas de valeur durant l'exécution. La difficulté d'optimisation par PG est ajustée par la valeur de la constante  $\mathcal{R}$  selon Daida *et al* [26]. Le problème est plus difficile lorsque  $\mathcal{R}$  est grand.

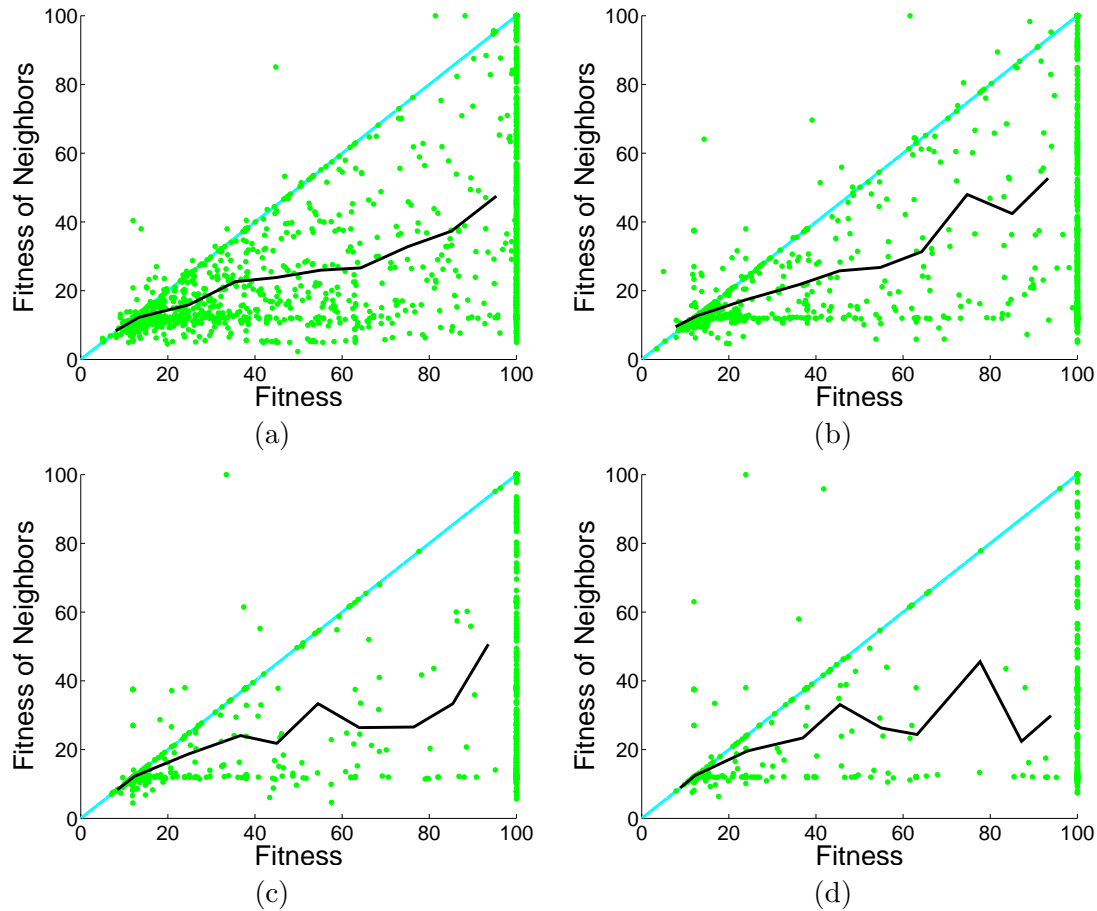


FIG. 2.18 – Nuage adaptatif et segments moyens pour le problème binomial-3 pour différentes valeurs de  $a_{\mathcal{R}}$ . (a) :  $a_{\mathcal{R}} = 1$ , (b) :  $a_{\mathcal{R}} = 10$ , (c) :  $a_{\mathcal{R}} = 10^2$  et (d) :  $a_{\mathcal{R}} = 10^3$ .

La figure 2.18 montre les nuages adaptatifs et les 10 segments obtenus pour différentes valeurs de  $a_{\mathcal{R}}$ . Nous n'avons pas représenté les points de performance au delà de 100 pour plus de clarté. Le tableau 2.8 donne les valeurs du CPN correspondantes. Les résultats montrent que la valeur du CPN devient plus petite à mesure que le problème devient plus difficile à optimiser.

### Le problème de parité paire

Le problème de parité paire, introduit par Koza [76], consiste à trouver la fonction booléenne de  $k$  variables qui renvoie *vrai* si son nombre de variables affectées à *vrai* est paire et renvoie *faux* dans le cas contraire. La performance d'un programme est le nombre d'erreurs de réponse parmi les  $2^k$  affectations de variables possibles. Un programme a au plus une performance de  $2^k$  et un programme parfait a une performance nulle. L'ensemble des fonctions utilisées par un

$a_{\mathcal{R}}$	CPN
1	0.0
10	-0.53
$10^2$	-1.01
$10^3$	-3.39

TAB. 2.8 – CPN pour le problème binomial-3 pour différentes valeurs de  $a_{\mathcal{R}}$ .

programme est  $\{NAND, OR\}$  et l'ensemble des terminaux est composé des  $k$  variables de la fonction booléenne à trouver. La difficulté est ajustée par le nombre de variables  $k$  de la fonction.

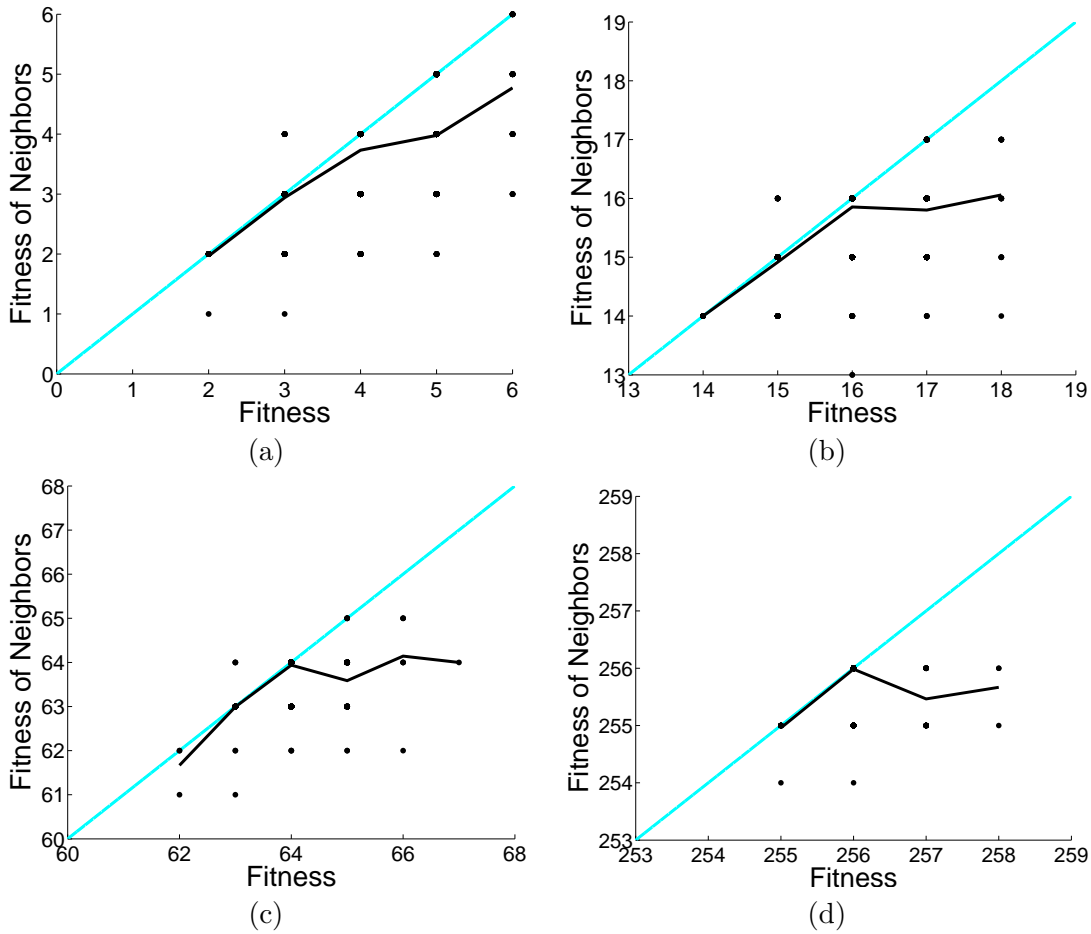


FIG. 2.19 – CPN et nuage adaptatif pour le problème de parité paire pour différent nombre de variables : (a)  $k = 3$ , (b)  $k = 5$ , (c)  $k = 7$ , (d)  $k = 9$ .

La figure 2.19 montre les nuages adaptatifs et les 10 segments obtenus pour différent nombre de variables  $k$ . Le tableau 2.9 donne les valeurs du CPN correspondant. De nouveau, les résultats montrent que la valeur du CPN devient plus petite à mesure que le problème devient plus difficile à optimiser.

Nombre de variables $k$	CPN
3	0.0
5	-0.11
7	-0.49
9	-0.55

TAB. 2.9 – CPN pour le problème binomial-3 pour différentes valeurs de  $a_{\mathcal{R}}$ .

### La fourmi artificielle

Dans ce problème, une fourmi artificielle est placée sur grille toroïdale de dimension  $32 \times 32$ . Certaines cellules de la grille contiennent des grains de nourritures. Le but est de trouver un programme de navigation pour la fourmi qui maximise le nombre de grains de nourriture pris. Nous utilisons le même ensemble d'instructions que dans J. Koza [76] sur la même grille de nourriture (Santa Fe trail). La fonction de performance d'un programme est le nombre restant de grains de nourriture parmi les 89 présents au départ. Langdon et Poli dans [79] ont étudié en détail ce problème. Ils ont montré que le nombre de bonnes solutions augmente exponentiellement avec la taille des programmes *i.e.* la profondeur de l'arbre syntaxique représentant un programme.

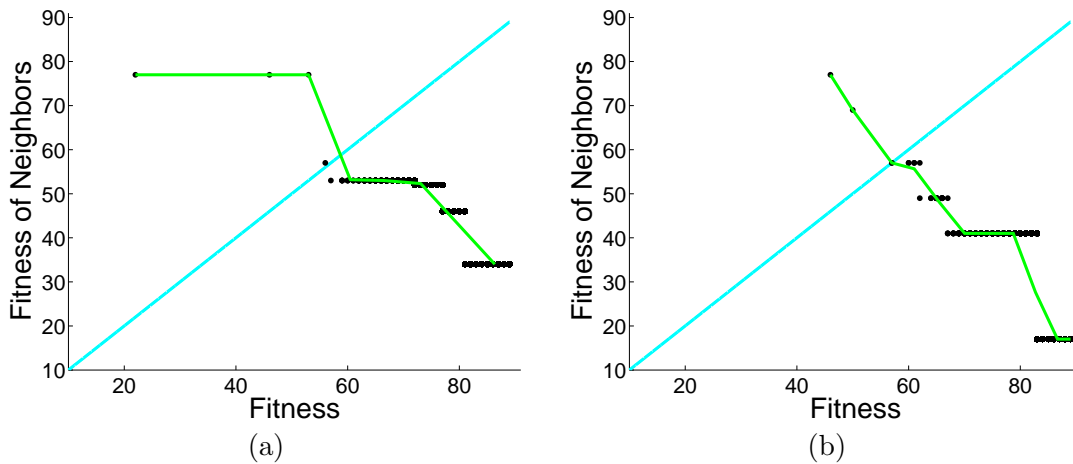


FIG. 2.20 – CPN et nuage adaptatif pour le problème de la fourmi artificielle pour deux profondeurs d'arbre (a) : profondeur 10, (b) : profondeur 6.

La figure 2.20 montre les nuages adaptatifs et les 10 segments obtenus pour deux profondeurs maximales de programme 10 et 6. Le CPN pour la profondeur 10 est  $-6.06$  et  $-11.42$  pour la profondeur 6. Ce problème contient beaucoup d'optima locaux et est difficile à optimiser, ce qui est corrélé avec les valeurs négatives du CPN. De plus, le CPN est plus petit lorsque le problème est plus difficile.

Dans les problèmes étudiés, nous avons montré expérimentalement que la statistique du coefficient de pente négative est corrélée avec la difficulté d'optimisation par programmation génétique. Sa valeur est négative lorsque l'instance du problème est difficile à optimiser. Au sein d'une même classe de problème, le CPN est autant plus négatif que l'instance est difficile à optimiser par PG. Seulement, le CPN ne permet pas de comparer la difficulté entre classes de problèmes puisque cette statistique n'est pas normalisée. Une autre faiblesse de la mesure est le choix du nombre de segments qui peut influencer la valeur du CPN.

TAB. 2.10 – Valeur du CPN pour les trois problèmes : multiplexeurs, Spirales enroulées, et arbres royaux (AR).

Problème	CPN	CPN <sub>bi</sub>
6 - multiplexeur	-0.16	0
11 - multiplexeur	-0.24	-0.21
Spirales enroulées	0	-0.41
AR racine B		0
AR racine C		0
AR racine D		0
AR racine E		-0.17
AR racine F		-0.21
AR racine G		-0.32

#### 2.4.4 Amélioration : méthode de la bisection

Afin de confirmer les résultats précédent et d’en mesurer les limites, nous avons réalisé la mesure du CPN sur trois nouveaux problèmes. Ces problèmes issus de la programmation génétique sont de natures différentes :

Le problème du *k-multiplexeur* [76] consiste à trouver une fonction booléenne de  $k$  bits en entrée et d’un bit en sortie. Les  $x$  premiers bits de l’entrée représentent un nombre entier entre 0 et  $2^x$  en écriture binaire qui désigne un des bits suivants entre la position  $x + 1$  et  $k$ , le bit à la position  $x + 1$  étant désigné par le nombre 0 et le bit à la position  $k$  désigné par le nombre  $2^x$ . La fonction booléenne recherchée donne en sortie la valeur du bit désigné en fonction des  $k$  bits de l’entrée. Les terminaux d’un programme sont les  $k$  bits de l’entrée, et les non-terminaux sont les opérateurs binaires booléens *AND*, *OR*, *NOT*, *IF*. La performance d’un programme est le nombre d’erreurs entre la valeur retournée par le programme et la valeur correcte attendue sur les  $2^k$  entrées possibles. Le résultat est divisé par  $2^k$  afin de normaliser la performance entre 0 et 1. Le problème k-multiplexeur est un problème de maximisation où la difficulté de résolution par PG augmente lorsque  $k$  augmente.

Le problème des *spirales enroulées* est un problème de classification [76]. Deux spirales enroulées l’une dans l’autre sont définies par 194 points dans un plan, il s’agit de classer les points appartenant à l’une ou l’autre des spirales. Les terminaux des programmes sont  $\{X, Y, \mathcal{R}\}$  où  $\mathcal{R}$  est un constante éphémère aléatoire comprise entre  $-1$  et  $1$ , et les non-terminaux sont  $\{+, -, *, //, IFLTE, SINCOS\}$  où  $//$  est la division protégée. La performance d’un programme est le nombre d’erreurs de classification sur les 194 possibles normalisé entre 0 et 1. Ce problème est un problème de maximisation où la PG ne trouve pas de solution exacte.

Le problème d’*arbres royaux* [103] consiste à trouver un arbre “parfait”. Les nœuds des arbres sont des fonctions notés A, B, C, etc d’arités respectives 1, 2, 3 etc. Un arbre parfait est un arbre où tous les liens sont “parfaits”. Un lien est parfait si un nœud d’arité  $n$  est joint avec un nœud d’arité  $n - 1$ . La performance d’un arbre est proportionnelle au nombre de liens parfaits. L’optimum global du problème est l’arbre parfait dont la racine est d’arité maximale. Punch [103] montre que la difficulté d’optimisation augmente avec l’arité maximale.

Le tableau 2.10 donne la valeur du CPN sur les trois problèmes. Les expériences sont menées de la même manière que dans la sous-section 2.4.3.

Les CPN pour les problèmes arbres royaux sont en accord avec la difficulté d’optimisation. On pourra se reporter aux travaux de Vanneschi [132] pour plus précision. Le CPN du 6-multiplexeur est négatif ce qui indique que le problème est difficile à optimiser ; pourtant, le taux

de succès de résolution par PG est au dessus de 50%. Pour le problème de spirale enroulées, il n’y a pas de résolution exacte connue par PG. Le CPN est de valeur nulle et donne une mauvaise indication.

Nous proposons donc une autre méthode de partitionnement des valeurs de performance, l’algorithme de bisection, afin de mieux prendre en compte le nombre de points dans chaque segment. Au premier pas de l’algorithme, l’ensemble des points du nuage adaptatif est divisé en deux segments de même taille contenant l’un les points d’abscisses les plus petites et l’autre les points d’abscisses les plus grandes. La même opération est appliquées récursivement aux segments obtenus. L’algorithme s’arrête soit lorsque le nombre de points dans un segment est plus petit qu’un seuil fixé, soit lorsque la taille du segment est plus petite qu’un autre seuil. Expérimentalement, nous avons choisi 50 points pour le premier seuil, et 5% de différence entre les abscisses des points de performance minimale et maximale pour le second seuil. Ensuite, le CPN est calculé de la même façon. Nous avons noté  $CPN_{bi}$  dans le tableau 2.10 le CPN ainsi obtenu par bisection. Le  $CPN_{bi}$  est en accord avec la difficulté des problèmes. Le  $CPN_{bi}$  reste un indicateur correct pour les problèmes d’arbres royaux. De plus, il est nul pour le problème 6-multiplexeur et négatif pour le problème des spirales enroulées.

Le  $CPN_{bi}$  améliore la mesure des pentes négatives puisque ce nouveau coefficient est en accord avec la difficulté d’optimisation par PG sur un plus grand nombre de problèmes représentatifs.

### 2.4.5 Synthèse du chapitre

Dans ce chapitre, nous avons défini le nuage adaptatif (NA) qui exprime la corrélation de performance entre solutions voisines relativement à un opérateur local. Il permet l’analyse du passage d’un ensemble de neutralité à un autre via un opérateur local, en particulier à l’aide de la courbe moyenne. Ce type d’analyse permet de décider de la pertinence d’un opérateur et du voisinage sur lequel il est basé puisqu’elle permet d’en déduire un certain nombre de mesures d’évolvabilité : la probabilité d’un ensemble de neutralité de meilleur performance, la performance moyenne des solutions atteignables par un opérateur, etc.

Nous avons donné l’expression analytique du NA pour une large famille de paysages, les paysages embarqués uniformes, relativement à l’opérateur de recherche aléatoire. Cette famille, que nous avons définie, généralise la famille des paysages NK et MAX-SAT. La fonction d’adaptation d’un paysage uniforme embarqué est une somme de sous-fonctions “indépendantes et identiques”. Dans ce cas, la corrélation est une droite dont la pente dépend du nombre de liens “épistatiques” entre les sous-fonctions et de la corrélation entre les valeurs prises par une même sous-fonction. Le NA est une somme de distributions normales qui dépendent des mêmes paramètres que la courbe moyenne. Les résultats théoriques ont été confirmés expérimentalement sur les paysages NK et MAX-SAT.

L’outil NA s’applique à tout opérateur local. Nous avons donc analysé le NA dans le cas de l’opérateur HC, d’exploitation maximale du voisinage, et de l’opérateur local SA utilisé par un recuit-simulé. L’étude analytique n’est que très partiellement possible pour l’opérateur SA. Une étude expérimentale menée sur les paysages NK et MAX-SAT montre que, pour l’opérateur HC, la courbe moyenne est une droite moyenne de pente plus faible que celle de l’opérateur de recherche aléatoire et pour l’opérateur SA, la courbe moyenne n’est plus une droite.

Nous avons montré que le NA peut servir de modèle de prédiction de l’évolution moyenne de la performance lors de l’itération d’un opérateur local. L’étude expérimentale sur les paysages NK et MAX-SAT montre que pour les opérateurs HC et SA, la modélisation est valide pour les premières itérations de l’opérateur.

Enfin, un lien entre la difficulté d’optimisation par un opérateur local et les caractéristiques du NA a été mis en évidence. Le Coefficient de Pente Négative (CPN) est une statistique ob-

tenue en sommant les pentes négatives de la courbe moyenne d'un NA spécifique où l'opérateur local est une sélection par tournoi et dont l'échantillonnage s'effectue à l'aide de l'algorithme de Métropolis-Hasting. Cette mesure de difficulté, particulièrement adaptée aux paysages issus de la programmation génétique, a été testée et validée sur un certain nombre de problèmes représentatifs. Une amélioration du calcul CPN par la méthode de bisection a permis d'élargir son domaine d'application. Une faiblesse du CPN est sa validation uniquement expérimentale. Une autre faiblesse est l'absence d'échelle universelle de ce coefficient sur l'ensemble des problèmes. Des avancées au niveau expérimentale et théorique restent à mener, toutefois, ce coefficient met en évidence que les pentes négatives de la courbe moyenne du NA sont corrélées avec la difficulté d'optimisation.

Le nuage adaptatif est un outil qui se décline sur un grand nombre de paysages et d'opérateurs afin d'étudier les rapports entre les ensembles de neutralité et un opérateur particulier : pertinence de celui-ci, prédiction de son évolution et difficulté d'optimisation. Cet outil met en avant la pertinence d'étudier un paysage adaptatif du point de vue des ensembles de neutralité. Dans le chapitre suivant, nous allons étudier un autre aspect de la neutralité qui ajoute une structure supplémentaire aux ensembles de neutralité : les réseaux de neutralité.





# Chapitre 3

## Réseaux de Neutralité

Dans ce chapitre, nous allons présenter les mesures existantes qui permettent de caractériser les réseaux de neutralité (RN) d'un paysage adaptatif. Nous allons définir une nouvelle mesure, l'*autocorrélation de l'évolvabilité*, qui mesure la corrélation de l'évolvabilité au cours d'une marche sur un RN. Cette mesure permet de compléter la description des paysages neutres à ces proximités des RN.

Nous analyserons à l'aide des mesures trois familles de paysages pour lesquelles la neutralité est ajustable. Ces trois variantes des paysages NK, les paysages  $NK_q$ ,  $NK_M$  et  $NK_p$ , sont représentatives de la manière d'obtenir de la neutralité dans un paysage additif.

### 3.1 Mesures des paysages adaptatifs neutres

Nous allons maintenant définir précisément les concepts liés à la neutralité dans le contexte des paysages adaptatifs : test de neutralité, voisinage neutre, marche neutre et réseau de neutralité.

**Définition:** Un *test de neutralité* est un prédicat  $isNeutral : S \times S \rightarrow \{vrai, faux\}$  qui à chaque couple de solutions associe une des valeurs de vérité *vrai* ou *faux*.

La valeur *vrai* du prédicat pour un couple de solutions  $(s_1, s_2) \in S^2$  signifie que la différence entre  $f(s_1)$  et  $f(s_2)$  est négligeable. Très fréquemment le prédicat  $isNeutral(s_1, s_2)$  est *vrai* ssi  $f(s_1) = f(s_2)$ . Dans ce cas,  $isNeutral$  induit une relation d'équivalence sur  $S$ . Le test de neutralité permet aussi de considérer d'autres situations fréquentes de "quasi-neutralité" où l'égalité de performance entre solutions n'est pas vérifiée strictement. Par exemple en évolution artificielle, nous pouvons définir  $isNeutral(s_1, s_2) = vrai$  ssi  $|f(s_1) - f(s_2)| \leq 1/S_{pop}$  avec  $S_{pop}$  la taille de la population. Lorsque  $f$  est stochastique, par exemple dans des problèmes d'apprentissage,  $isNeutral(s_1, s_2)$  est *vrai* ssi  $|f(s_1) - f(s_2)|$  est inférieure à l'erreur d'évaluation.

En biologie, la théorie neutraliste repose sur l'existence de mutations neutres au sens où elles n'ont pas ou peu d'influence sur la pression sélective. Dans le cadre des paysages adaptatifs, cela nous amène à définir la notion de *voisinage neutre*.

**Définition:** Pour tout  $s \in S$ , le *voisinage neutre* de  $s$  est l'ensemble  $N_{neut}(s) = \{s' \in \mathcal{V}(s) \mid isNeutral(s, s')\}$  et le *degré de neutralité* de  $s$ , noté  $nDeg(s)$  est le nombre de voisins neutres de  $s$ ,  $nDeg(s) = \#(N_{neut}(s) - \{s\})$ .

On dit qu'un paysage adaptatif est neutre, ou que son degré de neutralité est fort, lorsqu'un grand nombre de solutions ont un haut degré de neutralité. Cette définition est imprécise et il n'existe pas encore d'échelle qui permettrait de classer les problèmes selon leur degré de neutralité. Les plateaux, appelés aussi *réseaux de neutralité*, définis initialement par Schuster [112] en évolution moléculaire, s'ajoutent désormais à la description géométrique des paysages adaptatifs (voir figure 1.3). Nous modifions la définition de Schuster afin de l'adapter au cas de

quasi-neutralité. Un réseau de neutralité est un graphe connexe de solutions où il n’y a pas de différence de performance significative entre les solutions et où la relation de voisinage neutre permet de définir les arcs. Pour une définition formelle, nous utilisons la notion de *marche neutre* :

**Définition:** Une *marche neutre*  $W_{neut}$  de  $s$  à  $s'$  est une marche  $W = (s_0, s_1, \dots, s_m)$  de  $s$  à  $s'$  telle que pour tout  $(i, j) \in [0, m]^2$ ,  $isNeutral(s_i, s_j)$  est vrai.

**Définition:** Un *réseau de neutralité*, noté  $RN$ , est un graphe  $(G, N)$  où deux sommets de  $G$  sont connectés par un arc de  $N$  s’ils sont voisins neutres, et l’ensemble  $G$  des sommets est l’ensemble des solutions appartenant à  $\mathcal{S}$  tels que pour tout  $s$  et  $s'$  de  $G$ , il existe une marche neutre  $W_{neut}$  appartenant à  $G$  de  $s$  à  $s'$ .

### 3.1.1 Mesures existantes

Décrire la neutralité d’un paysage consiste à décrire dans un premier temps, les ensembles de neutralité comme nous l’avons fait dans le chapitre 2, et par la suite à décrire les réseaux de neutralité. Deux points de vue sont alors possibles, selon que l’on s’intéresse en propre à un réseau, ou bien aux relations entre réseaux.

L’étude en propre des RN repose sur la description des graphes des RN. Les mesures classiques des graphes utilisées pour décrire les RN sont alors :

- La *taille* : nombre de sommets du réseau,
- Le *diamètre* : distance (nombre minimal d’arcs d’un chemin reliant deux sommets) maximale entre solutions appartenant au réseau,
- La *distribution des degrés de neutralité* : distribution des degrés des sommets du graphe.

Pour étendre ces mesures à l’ensemble du paysage, on étudie pour la taille, la distribution des tailles des RN du paysage (voir la sous-section 3.2.2) ; pour la distribution des degrés de neutralité, la distribution des degrés de neutralité pour l’ensemble des solutions du paysage (voir la sous-section 3.2.1). Nous définissons le *degré de neutralité moyen du paysage* par la moyenne des degrés de neutralité des solutions du paysage.

Pour qualifier le graphe d’un  $RN$ , on étudie la corrélation des degrés qui permet une comparaison avec un graphe aléatoire. Ainsi, Bastolla [10] dans le domaine de l’évolution moléculaire, a défini l’autocorrélation des degrés de neutralité au cours d’une marche neutre : A partir des degrés collectés au cours d’une telle marche, on peut calculer l’autocorrélation de la série obtenue. Nous pouvons aussi utiliser cette mesure dans le cadre des paysages adaptatifs. L’autocorrélation des degrés mesure la structure de corrélation d’un  $RN$  (*i.e.* la répartition des degrés de neutralité sur le réseau). Une corrélation significative indique que la variation des degrés est faible ; il se crée alors sur le réseau des zones plus homogènes relativement au degré de neutralité. Aussi dans ce cas, le graphe n’est pas un graphe aléatoire. Toutes ces caractéristiques jouent un rôle important dans la dynamique des algorithmes évolutionnaires lorsque les solutions stagnent sur un réseau de neutralité (cf. partie 4.1).

L’étude inter-réseaux permet d’expliquer l’avantage potentiel de la neutralité dans un paysage adaptatif. On appelle *porte* une solution d’un réseau de neutralité dont un voisin au moins possède une valeur de performance strictement supérieure à celle du réseau. Huynen [60] a défini le *taux d’innovation* d’un  $RN$ , comme le nombre de nouvelles valeurs de performance (précédemment non rencontrées) atteignables dans le voisinage des solutions au cours d’une marche aléatoire neutre. Un réseau est innovant lorsque ce taux est supérieur au taux d’innovation calculé lors d’une marche aléatoire à travers l’ensemble du paysage. Lorsque le taux d’innovation est élevé, la *percolation* des  $RN$  est alors importante. Les réseaux de neutralité ont alors une grande capacité d’exploration de l’espace de recherche.

Au vu de ces définitions, la figure 3.1 présente une alternative à la représentation classique d’un paysage adaptatif neutre.

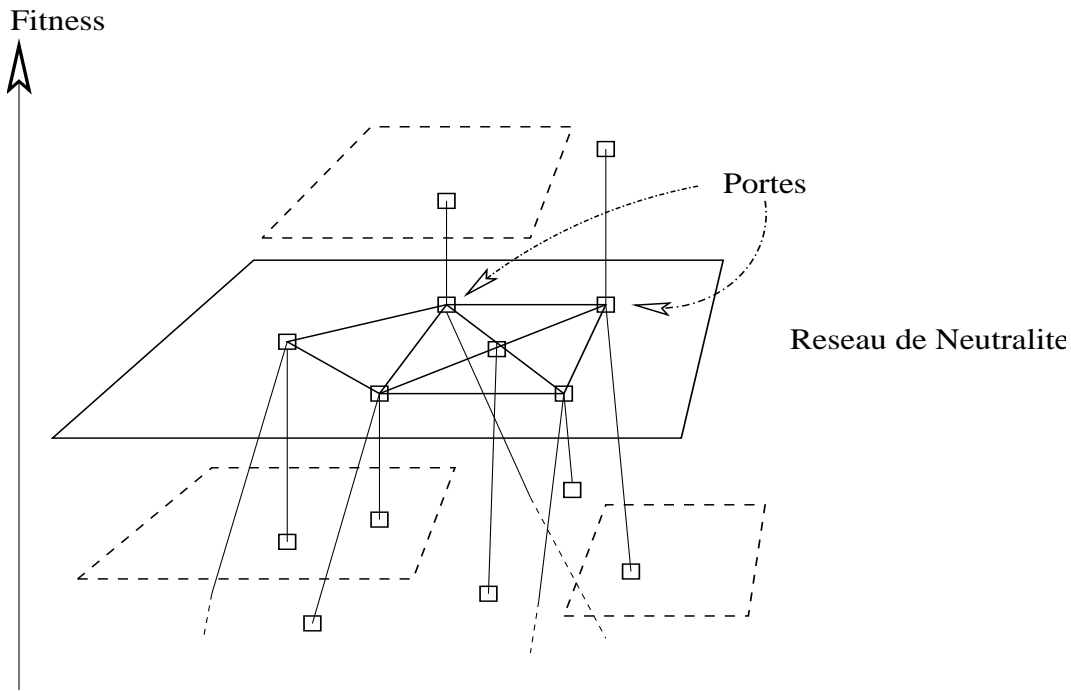


FIG. 3.1 – Représentation sous forme de graphe d’un paysage adaptatif neutre.

### 3.1.2 Nouvelle mesure : autocorrélation de l’évolvabilité

Le taux d’innovation permet de mesurer le nombre d’ensembles de neutralité accessibles dans le voisinage des solutions d’un réseau de neutralité. Il met en évidence la capacité d’exploration maximale d’un RN. Seulement, il ne mesure ni les performances de ces ensembles, ni la répartition autour du réseaux de ces performances.

Les performances dans le voisinage d’un RN peuvent être estimées à l’aide du nuage adaptatif. En effet, le nuage adaptatif donne les valeurs de performances accessibles par un opérateur local depuis un ensemble de neutralité. Or les ensembles de neutralité contiennent tous les RN de même performance, donc en supposant qu’il n’existe pas de différence statistique entre les RN de même performance, le nuage adaptatif permet d’obtenir les performances accessibles depuis un RN.

Afin de compléter la description inter-réseaux par la répartition des valeurs de performance accessibles depuis le voisinage d’un RN, nous définissons une famille de nouvelles mesures. Cette famille est basée sur les notions d’évolvabilité, *i.e.* la capacité d’un opérateur à produire de meilleures solutions (cf section 1.3.2), et de marche neutre sur les RN.

La notion d’évolvabilité est utilisée pour définir le nuage adaptatif. En effet, le nuage adaptatif représente l’évolvabilité des ensembles de neutralité relative à un opérateur. Ici, nous utilisons la possibilité supplémentaire offerte par les RN d’effectuer une marche neutre pour définir l’*autocorrélation de l’évolvabilité* au cours d’une marche neutre.

Nous utiliserons les notations suivantes :

- $(\mathcal{S}, \mathcal{V}, f)$  est un paysage adaptatif,
- $op : \mathcal{S} \rightarrow \mathcal{S}$  un opérateur<sup>16</sup> local agissant sur  $\mathcal{S}$  tel que pour tout  $s \in \mathcal{S}$ ,  $op(s) \in \mathcal{V}(s)$ ,
- $Y$  l’évolvabilité de cet opérateur, *i.e.* la v.a.  $Y : \mathcal{S} \rightarrow \mathbb{R}$  telle que  $Y(s) = f(op(s))$ ,
- $evol : \mathcal{S} \rightarrow \mathbb{R}$  une mesure d’évolvabilité relative à  $Y$  (cf. section 2.1).

<sup>16</sup>cet opérateur peut-être stochastique

**Définition:** L'*autocorrélation de l'évolvabilité* sur le réseau de neutralité  $N$  relative à l'opérateur  $op$  est l'autocorrélation d'une série  $(evol(s_0), evol(s_1), \dots)$  où  $(s_0, s_1, \dots)$  est une marche neutre sur  $N$ .

Plusieurs choix d'opérateurs locaux sont possibles : l'opérateur HC qui sélectionne une solution voisine de meilleure performance ou bien l'opérateur de sélection par tournoi utilisé dans la section 2.4.2 pour le calcul du CPN. Dans les deux cas, la mesure d'évolvabilité  $evol$  est définie par  $evol(s) = f(HC(s))$ . Si l'opérateur local est l'opérateur RA de recherche aléatoire, plusieurs choix de mesure d'évolvabilité sont possibles. Par exemple  $evol(s)$  peut être la probabilité d'obtenir une performance supérieure à celle du RN, ou la moyenne des performances supérieures à celle du RN, etc.

L'évolvabilité mesure la distribution de performance des solutions voisines, l'autocorrélation de l'évolvabilité permet donc de décrire le paysage à proximité des RN. Si la corrélation est importante, le voisinage proche des RN est "régulier", au contraire, absence de corrélation montre une grande diversité de l'évolvabilité et donc une discontinuité autour des RN.

Lorsque la recherche s'effectue sur un réseau de neutralité, la performance des solutions ne fournit pas une information suffisante pour guider la recherche. Nous avons vu que l'autocorrélation au cours d'une marche aléatoire (cf section 1.3.2), qui mesure la rugosité d'un paysage, indique si la performance entre solutions voisines est suffisamment corrélée pour pouvoir utiliser une recherche locale. De même, l'autocorrélation de l'évolvabilité indique si l'évolvabilité peut être une quantité que l'on peut optimiser. L'autocorrélation de l'évolvabilité ne fournit pas nécessairement d'information concernant les portes des RN si ce n'est dans le cas où la mesure d'évolvabilité est relative aux portes des RN. Nous utiliserons dans le chapitre 4, l'information de l'évolvabilité pour définir une nouvelle métaheuristique, la *recherche périscopique*, adaptée aux paysages neutres.

### 3.2 Réseaux de neutralité sur les variantes des paysages NK

Dans cette section, nous réalisons l'étude expérimentale de la neutralité de trois familles de paysages adaptatifs pour lesquels la neutralité est ajustable : les paysages  $NK_q$ , les paysages Technologiques, notés ici  $NK_M$ , et les paysages  $NK_p$  (voir la sous-section 1.4.3). Ces trois familles sont basées sur la définition du paysage NK qui est un paysage où seule l'épistasie est ajustable et où il n'existe pas de RN. Les trois variantes du paysages NK sont représentatives de la manière d'obtenir de la neutralité dans un paysage adaptatif additif. En effet, pour ces trois familles, la performance d'une solution est une somme de nombre réels. Cette somme est particularisée selon la famille de paysages, afin d'augmenter la probabilité qu'elle reste constante lorsque l'on modifie un certain nombre de ces termes.

- Dans les paysages  $NK_q$ , les termes de la somme sont des nombres entiers compris entre 0 et  $q - 1$ . Ainsi, lorsqu'on modifie certains termes, il est possible de conserver la même somme. Intuitivement, le degré de neutralité moyen augmente lorsque le paramètre de neutralité  $q$  diminue et on peut s'attendre à ce que les RN soient "structurées".
- Dans les paysages  $NK_M$ , on ne modifie pas la façon d'obtenir la somme, celle-ci est "arrondie" à l'une des fractions  $0, \frac{1}{M}, \dots, \frac{M-1}{M}$  immédiatement inférieure ( $\frac{k}{M} \leq S < \frac{k+1}{M}$ ). Intuitivement, le degré de neutralité moyen augmente grâce à la redondance lorsque  $M$  diminue. On peut s'attendre à ce que les RN soient peu "structurées".
- Dans les paysages  $NK_p$ , un terme de la somme est nul avec une probabilité  $p$ . Intuitivement, le degré de neutralité moyen augmente avec  $p$  et on peut s'attendre que les RN soient "structurées" comme pour les paysage  $NK_q$ , le paramètre  $p$  contrôlant plus fortement cette structure que le paramètre  $q$ .

Cette étude permet donc de comparer différentes façons d'introduire de la neutralité dans un paysage additif. De plus, elle permettra de mieux appréhender la neutralité d'un paysage adaptatif donné en comparant ses propres mesures de neutralité à celles présentées dans cette étude.

De façon à pouvoir mener une étude exhaustive, nous avons choisi une taille raisonnable ( $2^{16}$ ) pour l'espace de recherche ( $N = 16$ ). Pour les trois familles de paysages, le paramètre épistatique  $K$  décrit l'ensemble  $\{1, 2, 3, 5, 8\}$ . Les trois paramètres  $q$ ,  $M$  et  $p$  ajustant la neutralité décrivent respectivement les ensembles :  $\{2, 3, 4, 10\}$ ,  $\{16, 32, 48, 160\}$  et  $\{0.5, 0.8, 0.9, 0.95, 0.99\}$ . Pour chaque valeur des paramètres, 50 instances indépendantes de paysages sont générées. Afin de mettre en évidence l'influence du paramètre  $N$ , nous avons également réalisé l'étude pour une plus grande taille de l'espace de recherche ( $N = 64$ ). Pour cette valeur, le paramètre  $K$  décrit l'ensemble  $\{2, 4, 8, 12, 16\}$ . Les trois paramètres  $q$ ,  $M$  et  $p$  ajustant la neutralité décrivent les mêmes ensembles que précédemment. Pour chaque valeurs des paramètres où  $N = 64$ , 10 instances indépendantes de paysages sont générées. Tous les graphiques, pour toutes les valeurs des paramètres, n'ont pu être présenté dans ce mémoire, l'ensemble des résultats ainsi que le code c++ basé sur la librairie EO<sup>17</sup> sont disponibles sur la page web :

<http://www.i3s.unice.fr/~verel>.

### 3.2.1 Distribution du degré de neutralité

Dans cette sous-section, nous étudions la distribution du degré de neutralité pour les différentes valeurs des paramètres. Pour la famille des paysages  $NK_p$ , Barnett [8] (p. 113) donne l'expression analytique de ces distributions :

Pour la famille des paysages  $NK_q$  et  $NK_M$ , l'expression analytique des distributions n'est pas connue. Nous avons réalisé l'étude expérimentale de ces distributions. La figure 3.3 donne la moyenne des 50 distributions du degré de neutralité pour quelques valeurs de paramètres. Les distributions sont unimodales et proches d'une distribution binômiale pour les paysages  $NK_q$  et  $NK_M$ . Afin de vérifier si les distributions sont binômiales, nous avons réalisé le test du chi2 pour les paysages  $NK_q$  et  $NK_M$ . Le tableau 3.1 résume les valeurs obtenues. La première colonne indique la valeur moyenne des p-valeurs du test sur les 50 instances des paysages. La seconde colonne indique le nombre d'instances vérifiant le test au seuil de 5%.

Le nombre d'instances dont la distribution est proche d'une binômiale est faible pour  $K = 1$  et pour  $K = 8$ . Lorsque le paramètre  $K$  a une valeur intermédiaire (entre 2 et 5), ce nombre devient non nul, la distribution de degré de neutralité est une binômiale pour certaines instances. Ce nombre est d'autant plus grand que le paramètre ajustant la neutralité est petit. Un résumé des distributions s'obtient en calculant l'espérance et l'écart-type de celles-ci, *i.e.* le degré de neutralité moyen (et l'écart-type) du paysage. Le tableau 3.2 donne les valeurs des moyennes de l'espérance et l'écart-type des distributions sur les 50 instances. La figure 3.3 montre la moyenne des degrés de neutralité moyen des paysages en fonction du paramètre ajustant la neutralité pour les différentes valeurs de  $K$ . Pour l'ensemble des distributions, la moyenne décroît lorsque le paramètre  $K$  croît. Lorsque la multimodalité augmente, le degré de neutralité moyen du paysage diminue. Pour un paramètre  $K$  fixé, le degré de neutralité moyen du paysage est monotone en fonction du paramètre de neutralité. Il est décroissant pour les paysages  $NK_q$  et  $NK_M$  et croissant pour les paysages  $NK_p$ . Le degré de neutralité moyen des paysages et l'inverse du paramètre de neutralité ( $q$ ,  $M$  ou  $p$ ) du paysage sont corrélés. Si on note  $D$  le degré moyen du paysage et  $x$  l'un des trois paramètres de neutralité  $q$ ,  $M$  ou  $p$ , alors la loi de corrélation est  $D = \frac{a}{x} + b$ . Le tableau 3.3 donne les paramètres de la corrélation. La corrélation est forte puisque pour les paysages  $NK_q$  et  $NK_M$  le coefficient de corrélation est au-dessus de 0.999 et au dessus

---

<sup>17</sup><http://www.lri.fr/~marc/EO> ou <http://eodev.sourceforge.net>

TAB. 3.1 – Moyenne p-valeur du test du chi2 et nombre de tests vérifiés au seuil de 5% pour les distributions du degré de neutralité des familles de paysages  $NK_q$  et  $NK_M$ .

$K$	Paysages $NK_q$			Paysages $NK_M$		
	$q$	p-valeur	test <sup>+</sup>	$M$	p-valeur	test <sup>+</sup>
1	2	7384	0	16	2363	0
	3	6435	0	32	1392	0
	4	7149	0	48	1792	0
	10	6282	1	160	2254	1
2	2	1305	0	16	1182	0
	3	869	0	32	371	3
	4	848	1	48	424	5
	10	921	4	160	297	8
3	2	149	6	16	1109	0
	3	83	11	32	74	17
	4	79	14	48	48	22
	10	74	18	160	46	30
5	2	95	4	16	1589	0
	3	40	17	32	108	1
	4	74	2	48	47	17
	10	21	38	160	9	46
8	2	511	0	16	3861	0
	3	172	0	32	430	0
	4	195	0	48	162	0
	10	34	19	160	16	44

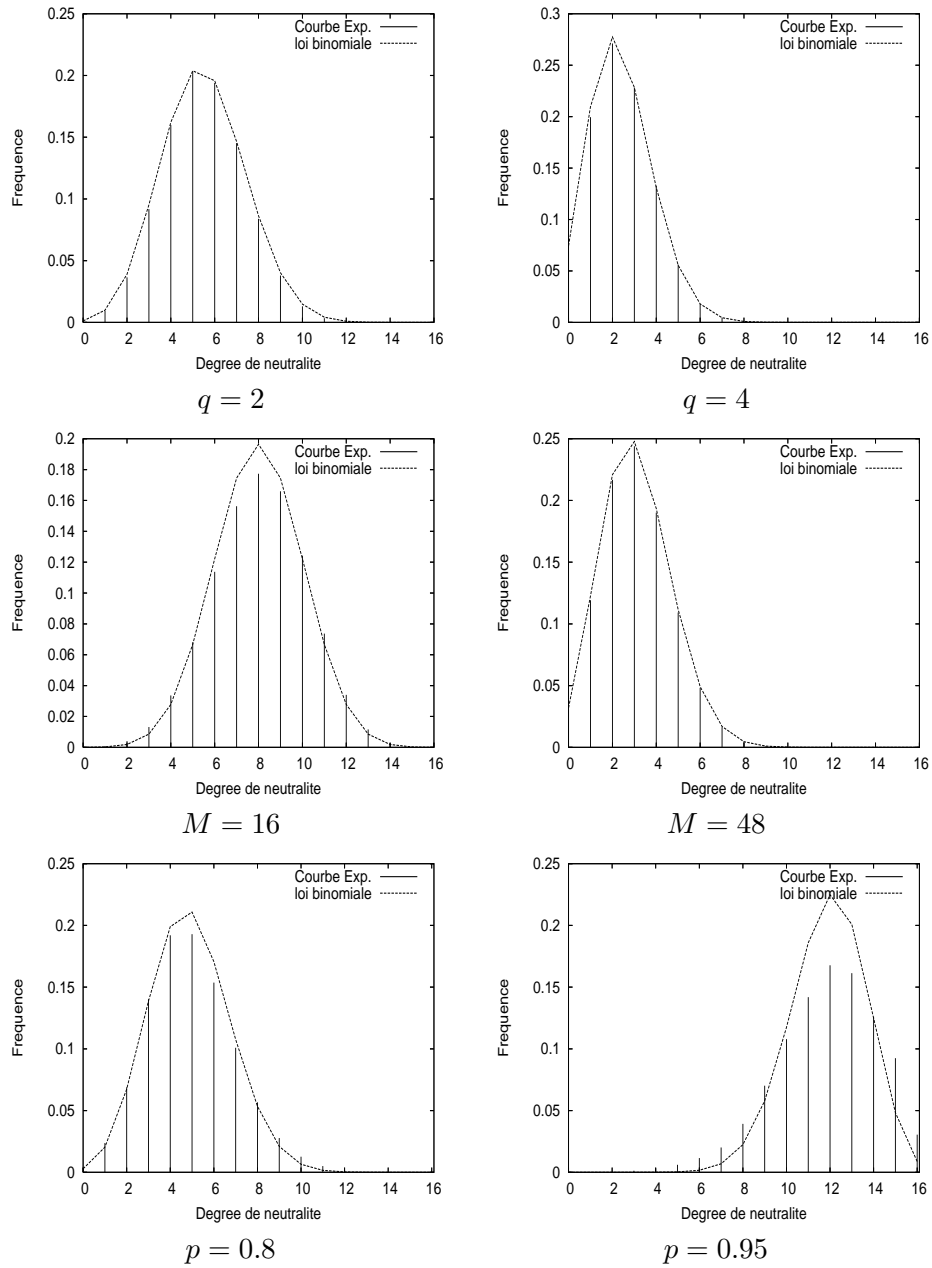


FIG. 3.2 – Exemple de moyenne de distribution du degré de neutralité pour différentes valeurs des paramètres (bâton) et distribution binômiale de même moyenne (ligne).

de 0.94 pour les paysages  $NK_p$ . L'équation de la droite de corrélation dépend du paramètre  $K$ . Le coefficient directeur de la droite de régression croît lorsque  $K$  augmente pour les paysages  $NK_q$  et  $NK_M$ , et décroît pour les paysages  $NK_p$ .

Nous avons étudié la distribution des degrés de neutralité des solutions de l'ensemble du paysage. Dans la suite de cette sous-section, nous allons étudier la répartition sur le graphe du RN de ces degrés de neutralité en utilisant l'autocorrélation des degrés de neutralité au cours d'une marche neutre introduite par Bastolla [10] (voir définition section 3.1). Cette étude permet de décider si les graphes des RN sont aléatoires ou non.

Pour chaque instance de paysage,  $10^2$  solutions initiales sont choisies uniformément dans le



TAB. 3.2 – Espérance ( $E$ ) et écart-type ( $\sigma$ ) des distributions du degré de neutralité pour les familles de paysages  $NK_q$ ,  $NK_M$  et  $NK_p$  pour  $N = 16$ .

$K$	Paysages $NK_q$			Paysages $NK_M$			Paysages $NK_p$		
	$q$	$E$	$\sigma$	$M$	$E$	$\sigma$	$p$	$E$	$\sigma$
1	2	6.2	1.53	16	9.0	2.08	0.50	1.8	0.86
	3	4.1	1.41	32	5.1	1.72	0.80	6.7	1.32
	4	2.7	1.19	48	3.5	1.50	0.90	10.5	1.34
	10	1.1	0.77	160	1.1	0.88	0.95	12.7	1.10
							0.99	15.4	0.35
2	2	5.5	1.79	16	8.0	2.11	0.50	0.8	0.72
	3	3.4	1.57	32	4.5	1.77	0.80	4.9	1.74
	4	2.4	1.37	48	3.1	1.54	0.90	8.6	2.00
	10	1.0	0.90	160	0.9	0.91	0.95	11.9	1.83
							0.99	15.0	0.91
3	2	4.7	1.81	16	7.1	2.13	0.50	0.4	0.52
	3	2.9	1.52	32	3.9	1.74	0.80	3.3	1.75
	4	2.0	1.33	48	2.7	1.49	0.90	7.1	2.43
	10	0.8	0.86	160	0.8	0.87	0.95	10.6	2.54
							0.99	14.8	1.39
5	2	3.7	1.73	16	6.0	2.11	0.50	0.1	0.21
	3	2.3	1.42	32	3.2	1.64	0.80	1.4	1.35
	4	1.6	1.23	48	2.2	1.38	0.90	4.8	2.63
	10	0.6	0.78	160	0.7	0.79	0.95	8.8	3.26
							0.99	14.2	2.22
8	2	3.0	1.65	16	5.0	2.10	0.50	0.0	0.03
	3	1.9	1.32	32	2.6	1.55	0.80	0.4	0.75
	4	1.3	1.12	48	1.7	1.29	0.90	2.6	2.36
	10	0.5	0.70	160	0.5	0.72	0.95	6.4	3.70
							0.99	13.3	3.22

TAB. 3.3 – Résultats pour les paysages  $NK_q$ ,  $NK_M$  et  $NK_p$  de la régression de la forme  $D = a/x + b$  où  $D$  est le degré de neutralité moyen du paysage et  $x$  l'un des paramètres de neutralité  $q$ ,  $M$  ou  $p$ .

$K$	Paysages $NK_q$			Paysages $NK_M$			Paysages $NK_p$		
	$a$	$b$	$\rho$	$a$	$b$	$\rho$	$a$	$b$	$\rho$
1	0.096	-0.031	0.999	0.006	0.021	1.000	-1.01	1.018	-0.978
2	0.108	-0.028	1.000	0.007	0.014	1.000	-2.37	2.297	-0.966
3	0.130	-0.039	1.000	0.008	0.013	1.000	-5.81	5.499	-0.957
5	0.167	-0.061	1.000	0.010	0.010	1.000	-37.5	34.8	-0.943
8	0.209	-0.079	1.000	0.012	0.008	1.000	-1630	1500	-0.933

paysage parmi les solutions appartenant aux RN de taille supérieure à 50 solutions. A partir de chaque solution initiale, 10 marches indépendantes sont effectuées. Nous réalisons ainsi  $10^3$  marches neutres et  $10^3$  marches aléatoires de longueur 150. Dans les instances où aucun RN n'est de taille supérieure à 50, aucune marche n'est effectuée. Pour chaque marche, nous calculons l'autocorrélation de la série des degrés de neutralité obtenue. Puis pour chaque instance, nous calculons la moyenne de chaque coefficient d'autocorrélation, enfin pour chaque valeur des paramètres, nous calculons de nouveau la moyenne des coefficients moyens obtenus pour chaque instance. Les figures 3.4, 3.5 et 3.6 présentent les résultats obtenus.

Pour les paysages  $NK_q$ , le coefficient d'autocorrélation  $\rho(1)$  d'ordre 1 décroît lorsque le paramètre de neutralité  $q$  augmente (sauf pour  $K = 1$  où  $\rho(1)$  est maximal pour  $q = 4$ ). Pour une valeur du paramètre  $q$  fixée,  $\rho(1)$  est décroissant avec  $K$ . Pour  $K = 1$  et  $K = 2$ ,  $\rho(1)$  est au dessus de la valeur 0.4 et pour  $K = 3$ ,  $\rho(1)$  est compris entre 0.45 et 0.25. La corrélation n'est pas nulle et le graphe des RN n'est pas un graphe aléatoire. Pour  $K = 5$  et  $K = 8$ ,  $\rho(1)$  est au-dessous de la valeur 0.2, la corrélation est faible, le graphe est plus proche d'un graphe aléatoire que précédemment. Les fonctions d'autocorrélation  $\rho$  sont de deux types. Lorsque le degré de neutralité moyen du paysage est plus grand, les fonctions d'autocorrélation sont des fonctions décroissantes (par exemple toutes les fonctions d'autocorrélation relative à  $K = 1$  et la fonction de paramètres  $K = 2$ ,  $q = 2$ ). Lorsque le degré de neutralité moyen est plus faible, les fonctions d'autocorrélation alternent entre une plus grande et une plus faible valeur pour la fonction  $\rho$ .

Pour les paysages  $NK_M$ , le coefficient d'autocorrélation  $\rho(1)$  d'ordre 1 décroît lorsque le paramètre  $M$  augmente. Pour une valeur du paramètre  $M$  fixée,  $\rho(1)$  est décroissant avec  $K$ . La valeur maximale de corrélation est de 0.27, et les valeurs pour  $K$  égales à 5 et 8 sont inférieures à 0.1. Les valeurs des corrélations sont plus faibles que pour les paysages  $NK_q$  bien que le degré de neutralité moyen soit plus élevé pour les paysages  $NK_M$ . Le graphe des RN des paysages  $NK_M$  est plus proche d'un graphe aléatoire. Les fonctions d'autocorrélation  $\rho$  sont toutes du second type décrit précédemment qui alternent deux valeurs de corrélation.

Pour les paysages  $NK_p$ , le coefficient d'autocorrélation  $\rho(1)$  d'ordre 1 croît lorsque le paramètre  $p$  augmente. Pour une valeur du paramètre  $p$  fixée,  $\rho(1)$  est décroissant avec  $K$ . Pour  $K$  inférieur à 8,  $\rho(1)$  est au dessus de la valeur 0.35 et pour  $K = 3$ ,  $\rho(1)$  seule la valeur pour  $p = 0.8$  est au-dessous de 0.3. La valeur minimale est 0.1 et la valeur maximale est 0.94. La corrélation n'est pas nulle, et est plus grande que dans le cas des autres paysages. Le graphe des RN n'est pas un graphe aléatoire. Les fonctions d'autocorrélation  $\rho$  sont principalement du type monotone décroissante (sauf pour  $p = 0.8$  avec  $K = 5$  et  $K = 8$ ).

## Influence du paramètre $N$

Afin de mettre en évidence l'influence de la taille de l'espace de recherche, nous avons étudié la distribution des degrés de neutralité lorsque  $N$  est égal à 64, et nous exposons les principaux résultats. Les distributions de degré de neutralité sont toujours unimodales, proches de distribution binômiale pour les familles de paysages  $NK_q$  et  $NK_M$ . Le tableau 3.4 donne la moyenne et l'écart-type des distributions pour différentes valeurs lorsque  $N$  est égale à 64. Pour les trois familles, comme pour  $N = 16$ , le degré de neutralité moyen décroît avec  $K$ . Pour un même rapport  $\frac{K}{N}$  et une même valeur du paramètre de neutralité, le degré de neutralité moyen est plus grand pour  $N = 64$  que pour  $N = 16$ . Pour les paysages  $NK_M$ , l'espace de recherche augmente et le nombre de valeurs de performance possibles reste constant, la probabilité que deux solutions voisines aient la même performance augmente donc. De même pour les paysages  $NK_q$  et  $NK_p$ , le nombre de valeurs de performance augmente moins que la taille de l'espace de recherche.

Nous avons étudié l'autocorrélation des degrés de neutralité pour  $N = 64$ . Pour chaque

TAB. 3.4 – Espérance ( $E$ ) et écart-type ( $\sigma$ ) des distributions du degré de neutralité pour les familles de paysages  $NK_q$ ,  $NK_M$  et  $NK_p$  pour  $N = 64$ .

$K$	Paysages $NK_q$			Paysages $NK_M$			Paysages $NK_p$		
	$q$	$E$	$\sigma$	$M$	$E$	$\sigma$	$p$	$E$	$\sigma$
2	2	21.343	3.577	16	53.555	9.624	0.5	3.543	1.550
	3	13.516	3.099	32	46.393	7.996	0.8	19.937	3.487
	4	9.791	2.754	48	38.411	5.719	0.9	35.260	4.043
	10	3.946	1.827	160	14.745	3.262	0.95	47.771	3.688
							0.99	60.161	1.930
4	2	16.663	3.491	16	52.165	9.756	0.5	0.726	0.792
	3	10.356	2.929	32	41.496	6.826	0.8	9.449	3.168
	4	7.580	2.572	48	32.611	4.992	0.9	24.329	5.129
	10	2.969	1.680	160	11.535	3.081	0.95	38.911	5.691
							0.99	57.812	3.692
8	2	12.282	3.170	16	48.466	9.416	0.5	0.022	0.138
	3	7.564	2.587	32	34.839	5.825	0.8	2.104	1.615
	4	5.537	2.273	48	25.777	4.525	0.9	11.284	4.482
	10	2.158	1.453	160	8.476	2.768	0.95	26.329	7.075
							0.99	53.391	6.329
12	2	10.148	2.985	16	45.305	9.061	0.5	0.002	0.038
	3	6.212	2.398	32	30.407	5.498	0.8	0.459	0.722
	4	4.545	2.075	48	21.920	4.356	0.9	5.047	3.157
	10	1.770	1.318	160	6.991	2.539	0.95	17.679	6.948
							0.99	49.337	8.373
16	2	8.830	2.858	16	42.704	8.766	0.5	0.001	0.018
	3	5.422	2.266	32	27.423	5.404	0.8	0.102	0.326
	4	3.962	1.953	48	19.407	4.284	0.9	2.318	2.101
	10	1.546	1.233	160	6.135	2.419	0.95	11.825	6.273
							0.99	45.550	10.092

instance de paysage,  $10^2$  solutions initiales sont choisies uniformément dans le paysage sans condition d'appartenance à un RN de taille suffisante. A partir de chaque solution initiale, 10 marches indépendantes sont effectuées. Nous réalisons ainsi  $10^3$  marches neutres et  $10^3$  marches aléatoires de longueur 150. Les fonctions d'autocorrélation sont calculées de la même manière que précédemment. La figure 3.24 donne le coefficient d'autocorrélation d'ordre 1 des degrés de neutralité pour les trois familles de paysages. Les valeurs des corrélations sont plus fortes pour  $N = 64$  que pour  $N = 16$ . Les variations du coefficient diffèrent pour les paysages  $NK_M$  et  $NK_p$ . Pour les valeurs du paramètre  $K$  égal à 2 et 4, les corrélations atteignent un minimum pour les valeurs du paramètre de neutralité  $M = 32$  et  $p = 0.9$  respectivement. Les corrélations restent les plus importantes pour les paysages  $NK_p$  et les plus faibles pour les paysages  $NK_M$ .

## Synthèse

Pour l'ensemble des trois paysages, les distributions de degré de neutralité sont unimodales. Pour les paysages  $NK_p$  et  $NK_M$ , nous avons montré expérimentalement que, pour certaines valeurs des paramètres, elles peuvent être considérées comme des distributions binômiales. Le degré de neutralité moyen du paysage est décroissant lorsque le paramètre épistatique  $K$  augmente, et il est décroissant lorsque le paramètre de neutralité  $q$  ou  $M$  augmentent dans les paysages  $NK_q$  et  $NK_M$ , ou le paramètre  $p$  diminue dans les paysages  $NK_p$ . Le paramètre de neutralité est prépondérant sur le degré d'épistasie  $K$  en ce qui concerne la valeur du degré de neutralité moyen. Les degrés moyens de neutralité les plus forts sont rencontrés avec les paysages  $NK_p$  et il est plus faible pour les paysages  $NK_q$  que pour les paysages  $NK_M$ . Pour des paramètres d'épistasie et de neutralité donnés, l'augmentation de la taille augmente le degré de neutralité moyen, l'augmentation la plus forte se rencontre dans les paysages  $NK_M$ .

Les corrélations de degré de neutralité entre solutions voisines dans un RN de neutralité ne sont pas nulles pour un certain nombre de paysages décrits ci-dessus. Cette corrélation est la plus forte pour les paysages  $NK_p$ , et la plus faible pour les paysages  $NK_M$ . La corrélation n'est pas directement la conséquence d'un haut de degré de neutralité moyen puisqu'elle peut différer à degré moyen comparable. Les graphes des RN ne sont pas des graphes aléatoires pour les familles de paysages étudiées, la variation des degrés est faible et il se crée donc un réseau des zones plus homogènes relativement au degré de neutralité. Empiriquement, au vu des valeurs obtenues pour les coefficients d'autocorrélation  $\rho(1)$  d'ordre 1, on peut établir une échelle pour qualifier la corrélation des degrés de neutralité (figure 3.8).

Pour les variantes des paysages additifs étudiés,

- discrétiser les termes de la somme limite la quantité de neutralité du paysage et donne une structure aux réseaux de neutralité,
- discrétiser les valeurs prises par la somme donne une structure plus faible aux réseaux de neutralité et ne limite pas la quantité de neutralité du paysage,
- rendre nulle certains termes de la somme permet de contrôler fortement le degré moyen du paysage. Le cas limite correspond à un paysage "plat". Cela donne aussi une structure aux réseaux de neutralité.

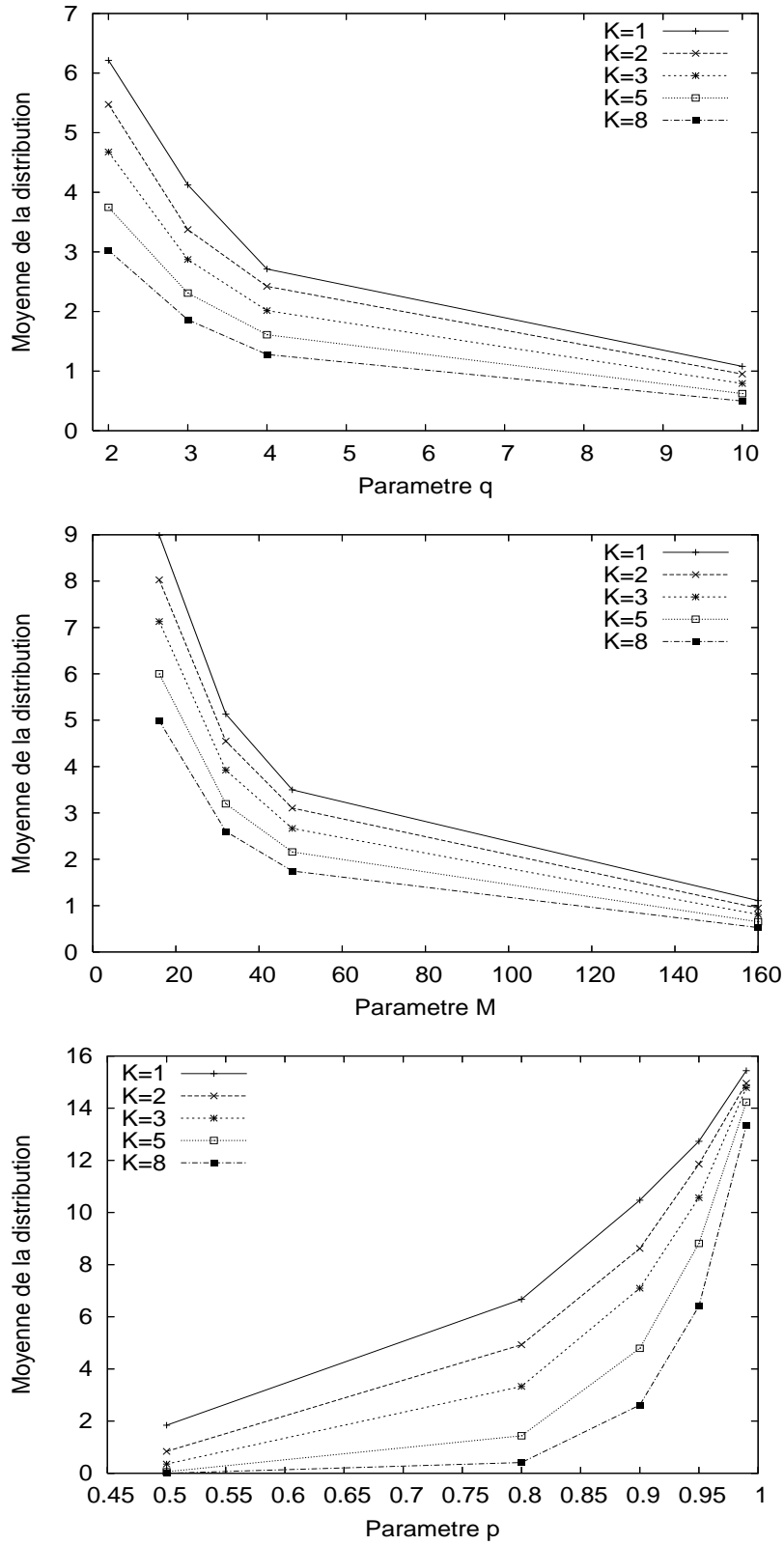


FIG. 3.3 – Degré de neutralité moyen en fonction des paramètres des paysages. On trouve de haut en bas la famille des paysages  $NK_q$ ,  $NK_M$  et  $NK_p$ .

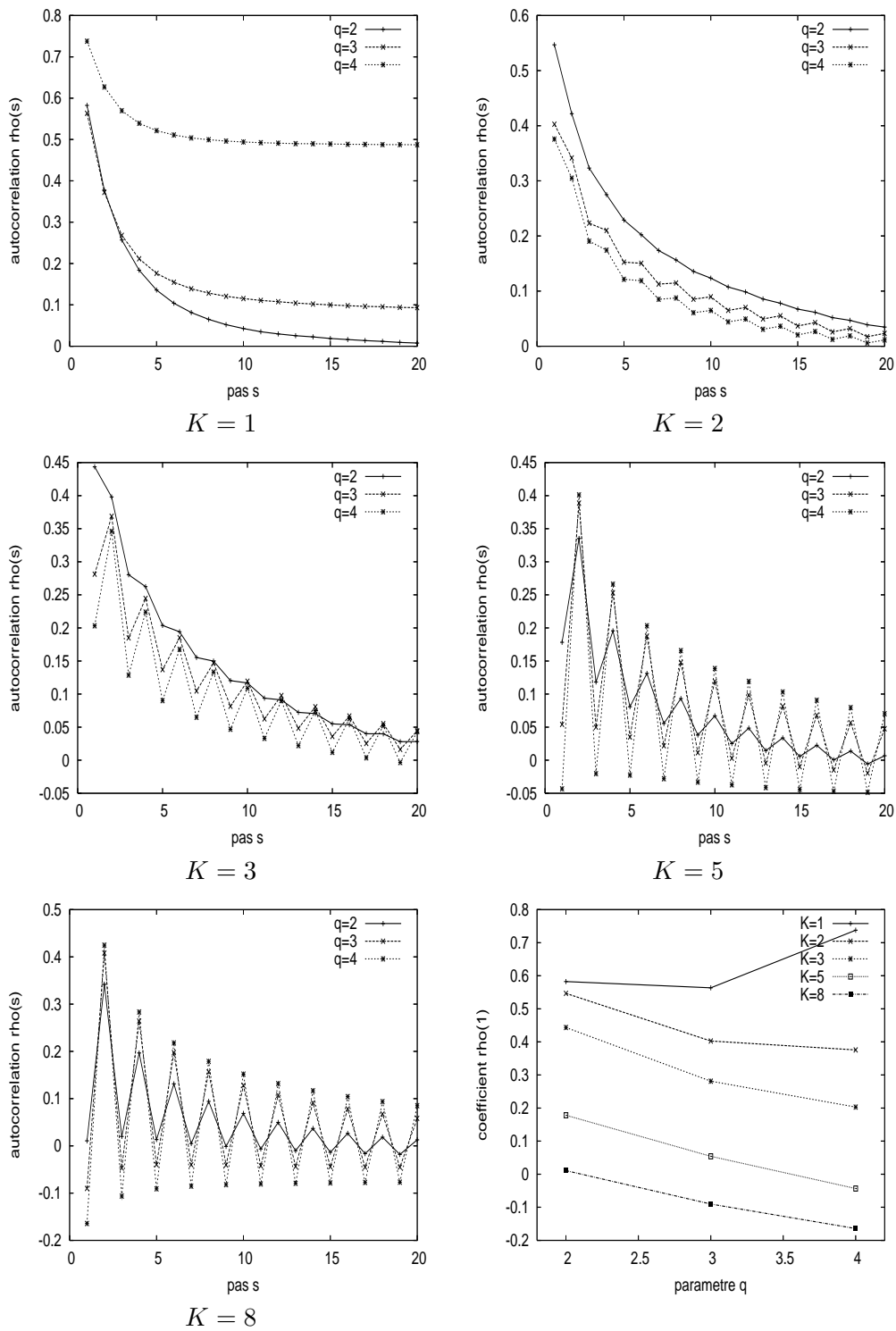


FIG. 3.4 – Fonctions d'autocorrélation des degrés de neutralité pour différentes valeurs du paramètre  $K$  et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages  $NK_q$ .

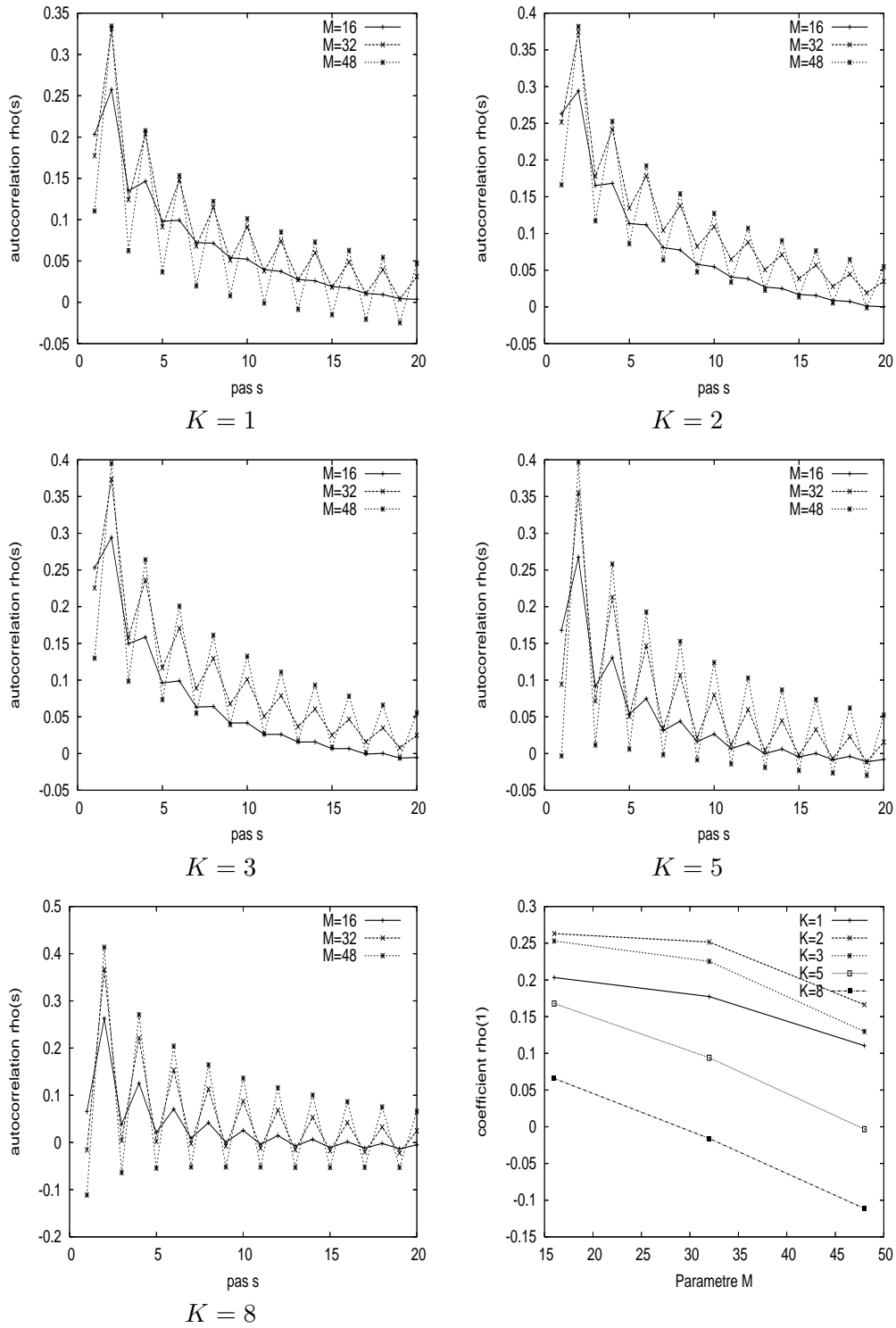


FIG. 3.5 – Autocorrélation des degrés de neutralité pour différentes valeurs du paramètre  $K$  et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages  $NK_M$ .

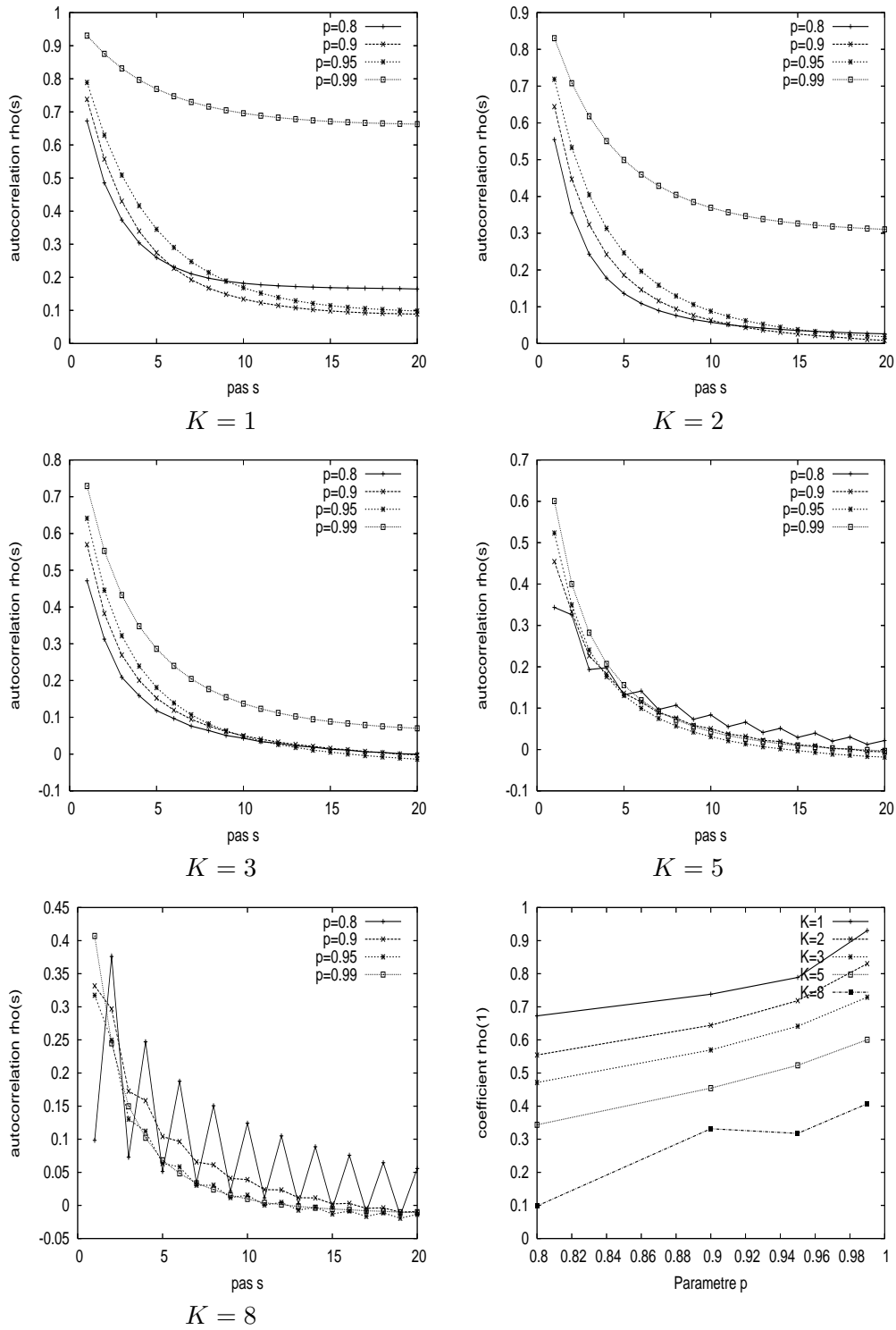


FIG. 3.6 – Autocorrélation des degrés de neutralité pour différentes valeurs du paramètre  $K$  et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages  $NK_p$ .



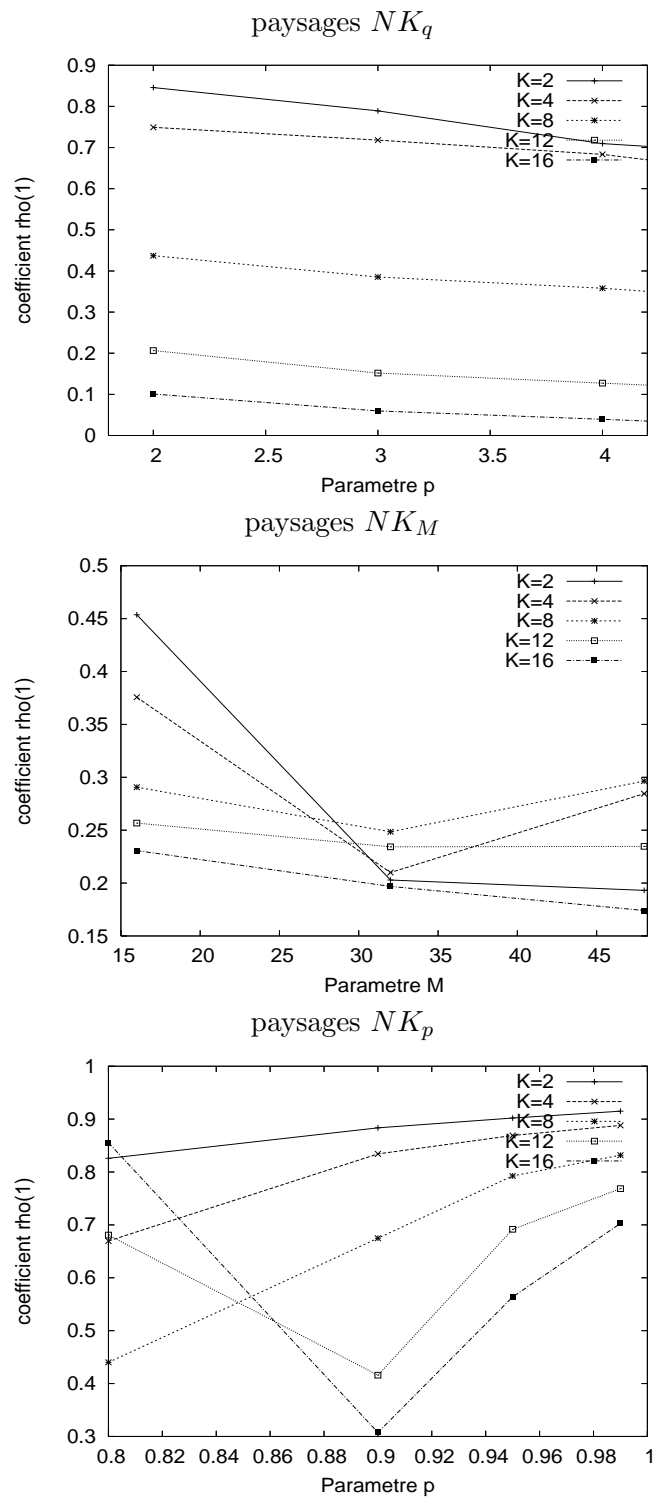


FIG. 3.7 – Coefficient d'autocorrélation d'ordre 1 des degrés de neutralité pour les paysages  $NK_q$ ,  $NK_M$  et  $NK_p$  avec  $N = 64$ .

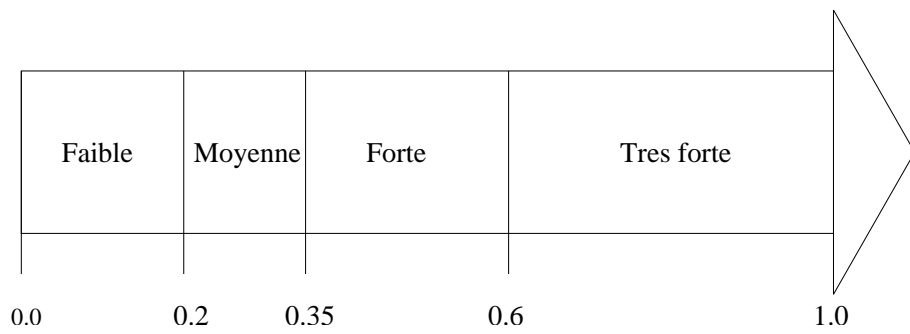


FIG. 3.8 – Échelle empirique de corrélation des degrés de neutralité en fonction du coefficient d'autocorrélation d'ordre 1.

### 3.2.2 Taille des réseaux de neutralité

Dans cette sous-section, nous étudions la taille des réseaux de neutralité pour les différentes familles de paysages. Pour les paysages  $NK_p$ , Barnett [8] (p. 133) expose une étude préliminaire où il présente les résultats expérimentaux du nombre de réseaux de neutralité en fonction de leur performance. Malheureusement, la distribution obtenue est peu représentative puisque l'écart-type du nombre de réseaux est important. En effet, pour une valeur de performance donnée, la taille d'un réseau de neutralité varie fortement d'une instance à l'autre. De plus, on peut constater que les paysages contiennent peu de RN de grande taille et beaucoup de RN de petite taille. Nous avons donc choisi de représenter la distribution des tailles des RN de la même manière que la distribution de la taille des villes en géographie ou que la distribution de la fréquence d'apparition d'un mot dans un texte en théorie de l'information. Le graphique consiste à représenter en échelle logarithmique en abscisse et en ordonnée la taille d'un RN en fonction du rang de cette taille parmi l'ensemble de tous les RN. Le RN de plus grande taille est de rang 1 et se situe le plus à gauche sur le graphique, le RN de plus petite taille se situe à droite du graphique et son rang correspond au nombre de RN du paysage. Les courbes obtenues sont donc décroissantes. Nous appellerons *profil rang-taille* ce graphique. La taille de l'espace de recherche est identique pour l'ensemble des paysages ( $N = 16$ ), la taille des RN peut donc être comparée entre les différents paysages. Nous n'avons pas étudié la taille des RN pour la valeur de  $N$  égale à 64, pour cette taille d'espace de recherche, il n'est pas envisageable de l'énumérer exhaustivement.

Les figures 3.9 à 3.14 présentent les profils rang-tailles des RN pour les différents paysages et paramètres. Pour chaque graphiques, les 50 profils correspondant aux 50 instances de paysages sont représentés, et la courbe en gras est la moyenne de la taille des RN de même rang. L'ensemble des 50 profils permet de juger approximativement de la variabilité de la distribution des tailles des RN selon les instances de paysages.

On peut observer deux types de profil. Pour le premier type (type 1), les profils sont composés de trois parties : une partie gauche formée de quelques RN (de quelques unités à une dizaine) de très grande taille approximativement de même valeur, une partie centrale où la taille décroît rapidement montrant qu'il existe peu de RN de tailles intermédiaires, enfin une troisième partie formée des RN de plus petites tailles (10 à 100 fois plus petits que la plus grande taille de RN) répartis suivant une droite décroissante. Ce type de profil présente un point d'inflexion dans la partie centrale. Les exemples représentatifs de ce type de profil sont les graphiques situés en haut à droite des figures 3.10 et 3.12 et en bas à droite de la figure 3.14. Pour le second type de profil (type 2), la courbe moyenne est concave. Les profils ne sont pas dominés par quelques unités de RN de très grande taille, et généralement les profils sont plus dispersés autour de la courbe moyenne que dans le type 1. Les exemples représentatifs de tels profils sont les graphiques situés dans la colonne de gauche des figures 3.9, 3.11 et 3.13.

Pour la famille des paysages  $NK_q$ , les profils sont du type 1 lorsque  $q$  est petit et  $K$  grand et du type 2 sinon. Pour une valeur de  $K$  fixée, la différence de taille entre les RN de premiers et de derniers rangs est d'autant plus grande que  $q$  est petit. Pour une valeur de  $q$  fixée, cette différence s'accroît légèrement lorsque  $K$  est grand. De la même manière pour la famille des paysages  $NK_M$ , les profils sont du type 1 lorsque  $M$  est petit et  $K$  grand et du type 2 sinon. Pour une valeur de  $K$  fixée, la différence de taille entre les RN de premiers et de derniers rangs est d'autant plus grande que  $M$  est petit. Pour une valeur de  $M$  fixée, cette différence est d'autant plus grande que  $K$  est grand. Pour la famille des paysages  $NK_p$ , peu de profils sont du type 1. Seuls les six paysages de paramètres  $p \geq 0.9$  et  $K \geq 5$  sont du type 1 où il y a des RN de grande taille en nombre au moins égale à une centaine.

Les paysages sont souvent dominés par quelques grands RN. La figure 3.15 montre la taille du plus grand RN normalisée par la taille de l'espace de recherche ( $2^{16}$ ) en fonction des paramètres des paysages. Pour les trois familles de paysages, pour une valeur de  $K$  fixée, la taille du plus grand RN décroît lorsque le degré de neutralité moyen diminue. Pour les paysages  $NK_p$  et  $NK_M$ , la valeur du paramètre  $K$  a peu d'influence sur la taille du RN excepté pour  $K = 1$  dans les paysages  $NK_M$ . Pour les paysages  $NK_q$ , pour une même valeur  $q$ , la taille du plus grand RN est plus petite pour les valeurs  $K = 1$  et  $K = 2$ , par exemple pour  $q = 2$ , la taille est de 2% pour  $K = 1$  et proche de 20% pour  $K = 3$ . La taille du plus grand RN des paysages  $NK_q$  est plus sensible à la valeur du paramètre  $K$  que pour les autres familles de paysages. Les plus grands RN sont obtenus pour les paysages  $NK_p$  avec  $p$  proche de 1. Il représente près de 85% du paysage alors que les plus grands RN pour les paysages  $NK_q$  atteignent entre 15% et 20% du paysage avec  $q = 2$  et entre 30% et 35% pour les paysages  $NK_M$  pour  $M = 16$ .

Lorsque  $K$  est petit, par exemple inférieur à 3 pour les paysages  $NK_q$ , inférieur à 2 pour les paysages  $NK_M$  ou inférieur à 3 pour les paysages  $NK_p$ , la taille du plus grand RN dépend fortement de l'instance du paysage. La figure 3.16 montre le rapport de l'écart-type par la moyenne de la taille du grand RN du paysage en fonction des paramètres. Nous appellerons variabilité ce rapport. Une valeur proche de 1 signifie que l'écart-type est de même ordre de grandeur que la moyenne, *i.e.* la variabilité de la taille du plus grand RN est grande. Pour l'ensemble des trois familles de paysages, la variabilité est la plus importante lorsque  $K = 1$  et décroît lorsque  $K$  augmente. Pour une valeur de  $K$  fixée, la valeur minimale est obtenue lorsque le paramètre de neutralité est maximal ( $q$  et  $M$  petit,  $p$  proche de 1); les variations sont différentes d'une famille de paysages à l'autre. Pour les paysages  $NK_p$ , la variabilité est maximale pour  $p = 0.8$  puis décroît linéairement jusqu'à une valeur comprise entre 0.0 et 0.2. Pour les paysages  $NK_M$ , la variabilité croît avec le paramètre  $M$  (sauf pour  $K = 1$  où la variabilité est maximale pour  $M = 32$ ). Pour les paysages  $NK_q$ , les variations dépendent de la valeur de  $K$ . Pour  $K = 5$  et  $K = 8$ , la variabilité est monotone croissante de 0.02 à environ 0.2, pour  $K = 1$ , la variabilité est importante au dessus de 1, enfin pour  $K = 2$  ou  $K = 3$ , les maxima de 0.7 et 0.43 sont atteints respectivement, pour  $q = 3$  et  $q = 4$ . Pour des valeurs de  $K$  comparables et une moyenne de degré de neutralité comparable, la variabilité la plus faible est rencontrée pour les paysages  $NK_M$  et la plus forte pour les paysages  $NK_p$ .

## Synthèse

Les profils rang-taille permettent de mieux comprendre la distribution des tailles des réseaux de neutralité, même dans le cas où la taille des RN varie fortement selon l'instance de paysage considérée. Ces profils ont permis de mettre en évidence que les paysages sont dominés par de grands RN. Ils peuvent être peu nombreux et de très grande taille (type 1) lorsque le paramètre d'épistasie  $K$  et le degré de neutralité moyen sont suffisamment grands ou plus nombreux et plus petits (type 2) lorsque le paramètre d'épistasie  $K$  et le degré de neutralité moyen sont plus petits. Dans ce dernier cas, la variabilité selon les instances de paysage est accentuée.

Pour les variantes des paysages additifs étudiés, il y a peu de différence entre les profils rang-taille des paysages selon que l'on discrétise certains termes de la somme ou le nombre de valeurs prises par la somme. Par contre, lorsque certains termes de la somme sont nuls, le nombre de très grands réseaux est important et il dépend plus fortement de l'épistasie du paysage.

Une étude analytique future de la taille des RN pourra utiliser la théorie de la percolation [123]. En effet, on peut définir la percolation d'arc entre deux solutions lorsqu'elles sont en relation par voisinage neutre. La théorie de la percolation donne des résultats concernant la plus grande composante connexe du graphe obtenu qui correspond à la taille du plus grand de réseau de neutralité du paysage. Elle pourrait peut-être expliquer le passage du type 1 ou type 2 lorsque les paramètres sont modifiés.

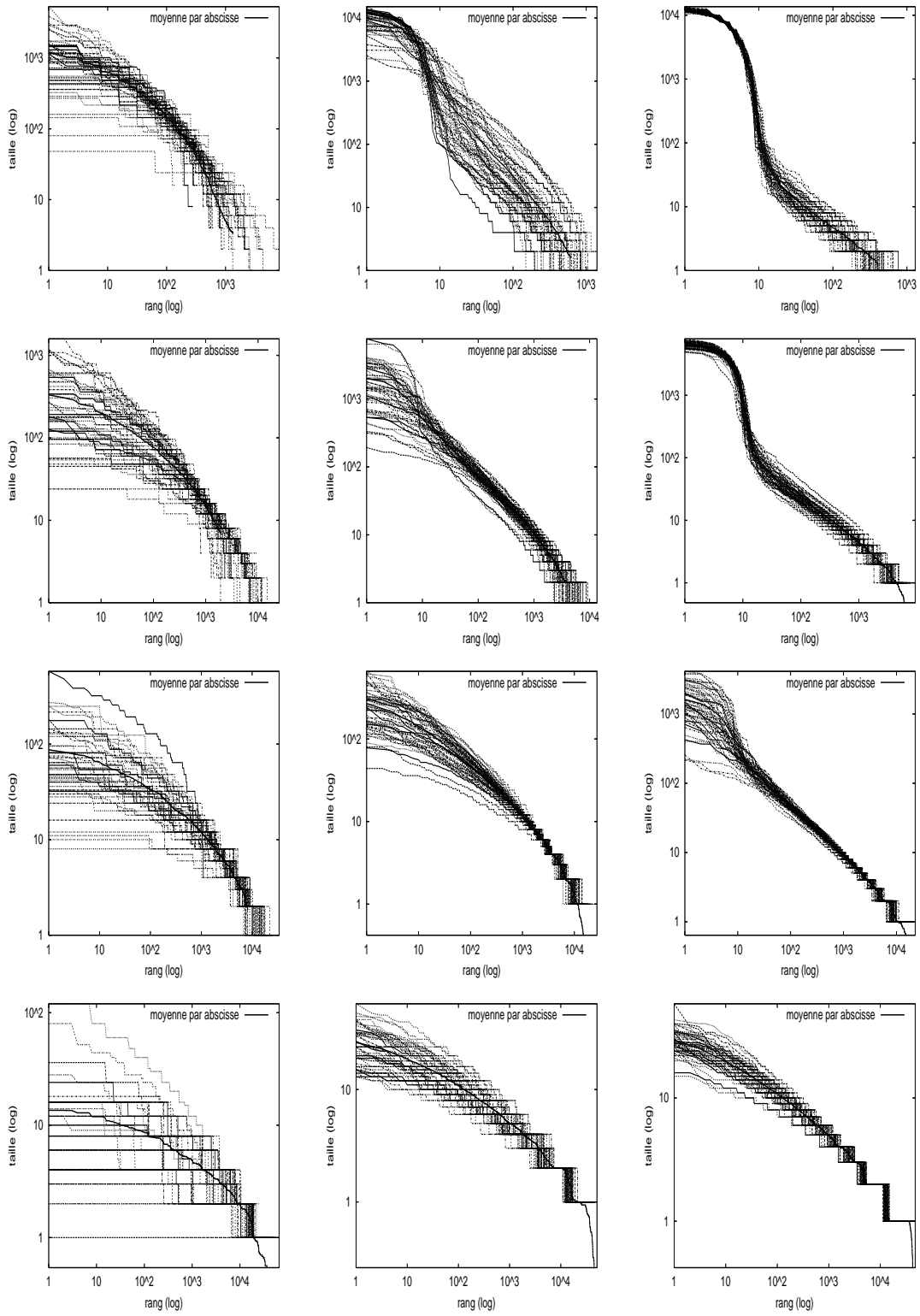


FIG. 3.9 – Profils rang-taille des RN pour les paysages  $NK_q$ . Le paramètre  $q$  prend les valeurs 2, 3, 4, 10 de haut en bas, et le paramètre  $K$  les valeurs 1, 2, 3 de gauche à droite.

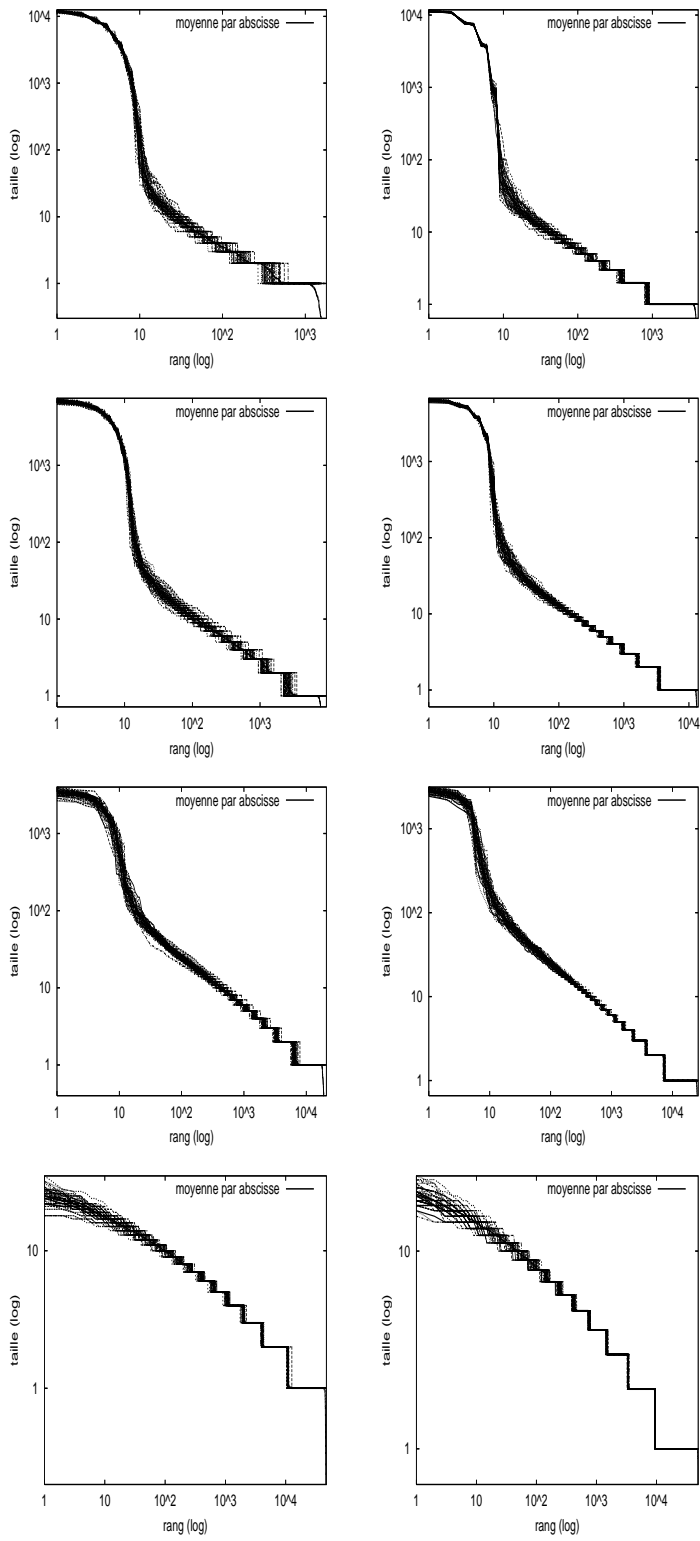


FIG. 3.10 – Profils rang-taille des RN pour les paysages  $NK_q$ . Le paramètre  $q$  prend les valeurs 2, 3, 4, 10 de haut en bas, et le paramètre  $K$  les valeurs 5, 8 de gauche à droite.

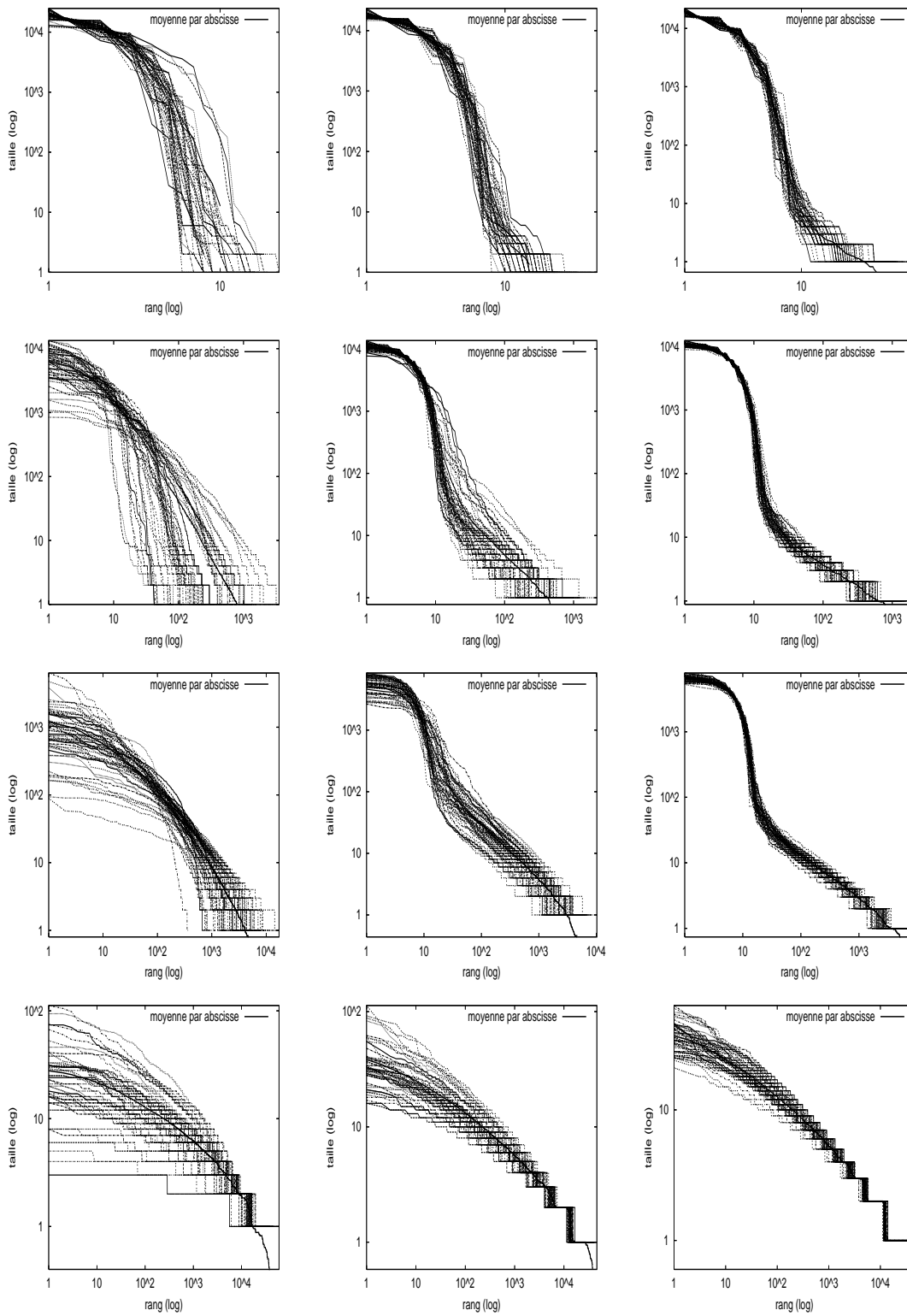


FIG. 3.11 – Profils rang-taille pour les paysages  $NK_M$ . Le paramètre  $M$  prend les valeurs 16, 32, 48, 160 de haut en bas, et le paramètre  $K$  les valeurs 1, 2, 3 de gauche à droite.

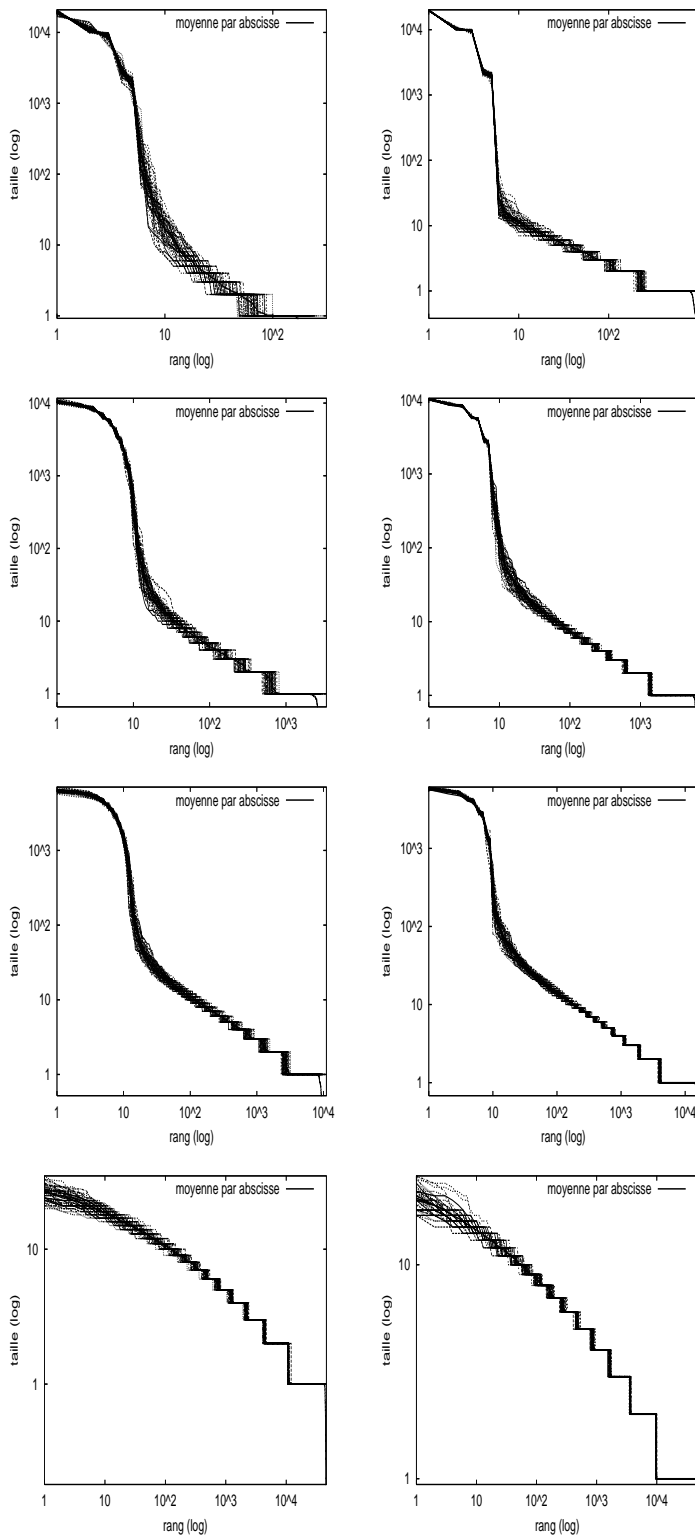


FIG. 3.12 – Profils rang-taille pour les paysages  $NK_M$ . Le paramètre  $M$  prend les valeurs 16, 32, 48, 160 de haut en bas, et le paramètre  $K$  les valeurs 5, 8 de gauche à droite.



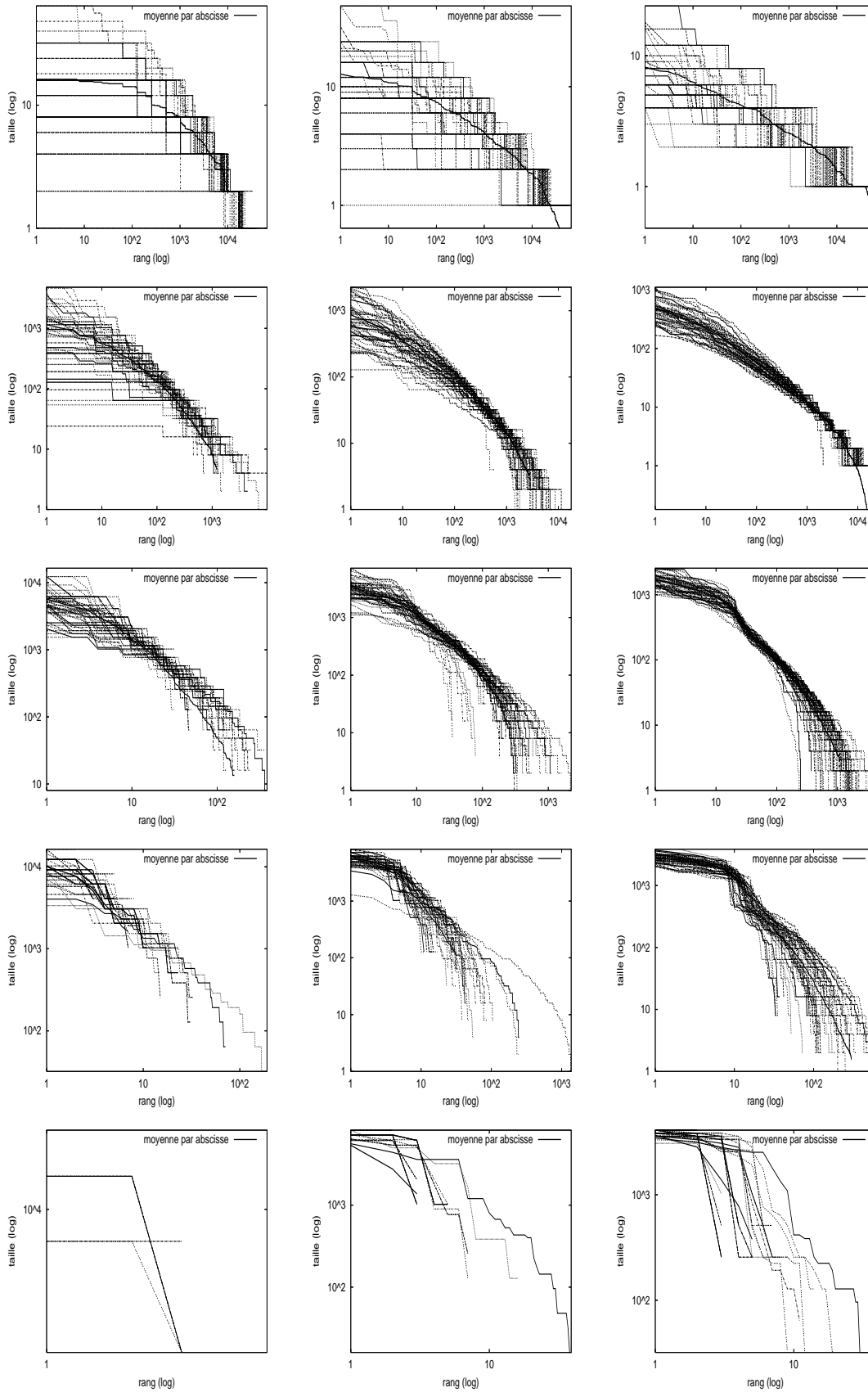


FIG. 3.13 – Profils rang-taille pour les paysages  $NK_p$ . Le paramètre  $p$  prend les valeurs 0.5, 0.8, 0.9, 0.95, 0.99 de haut en bas, et le paramètre  $K$  les valeurs 1, 2, 3 de gauche à droite.

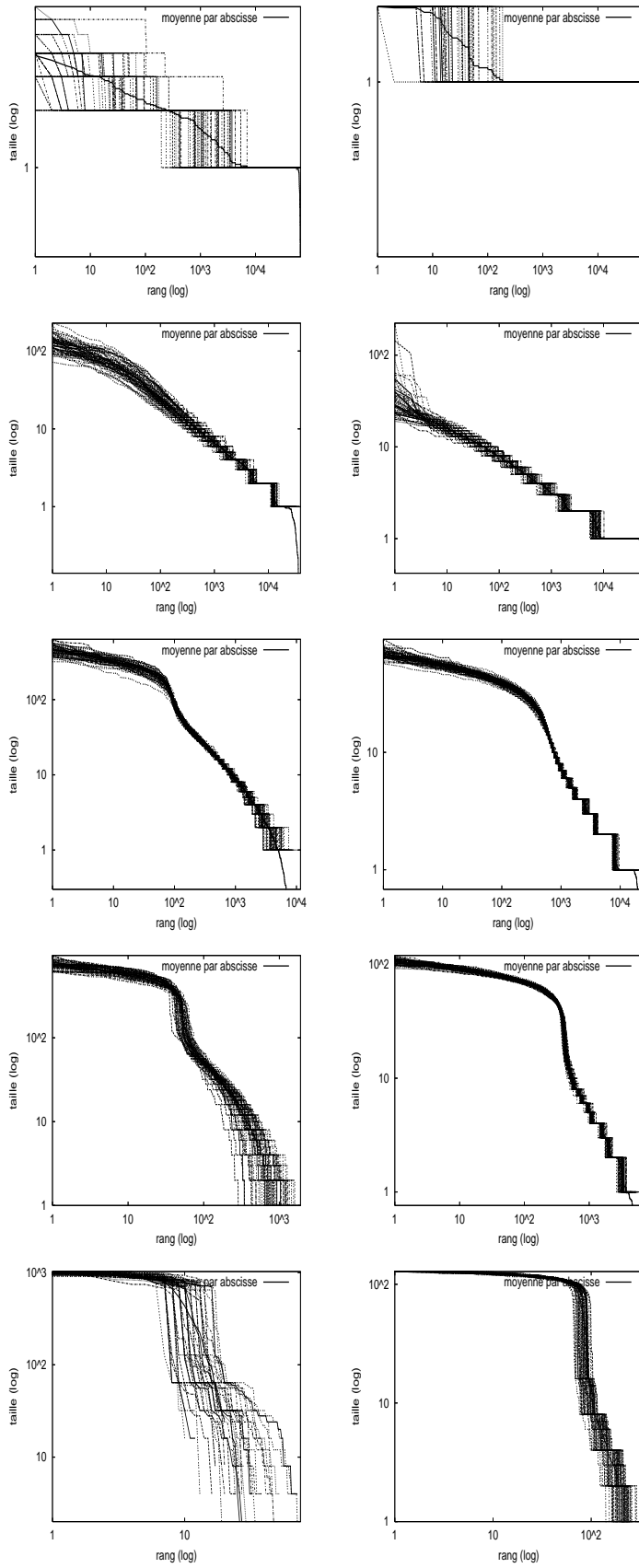


FIG. 3.14 – Profils rang-taille pour les paysages  $NK_p$ . Le paramètre  $p$  prend les valeurs 0.5, 0.8, 0.9, 0.95, 0.99 de haut en bas, et le paramètre  $K$  les valeurs 5, 8 de gauche à droite.

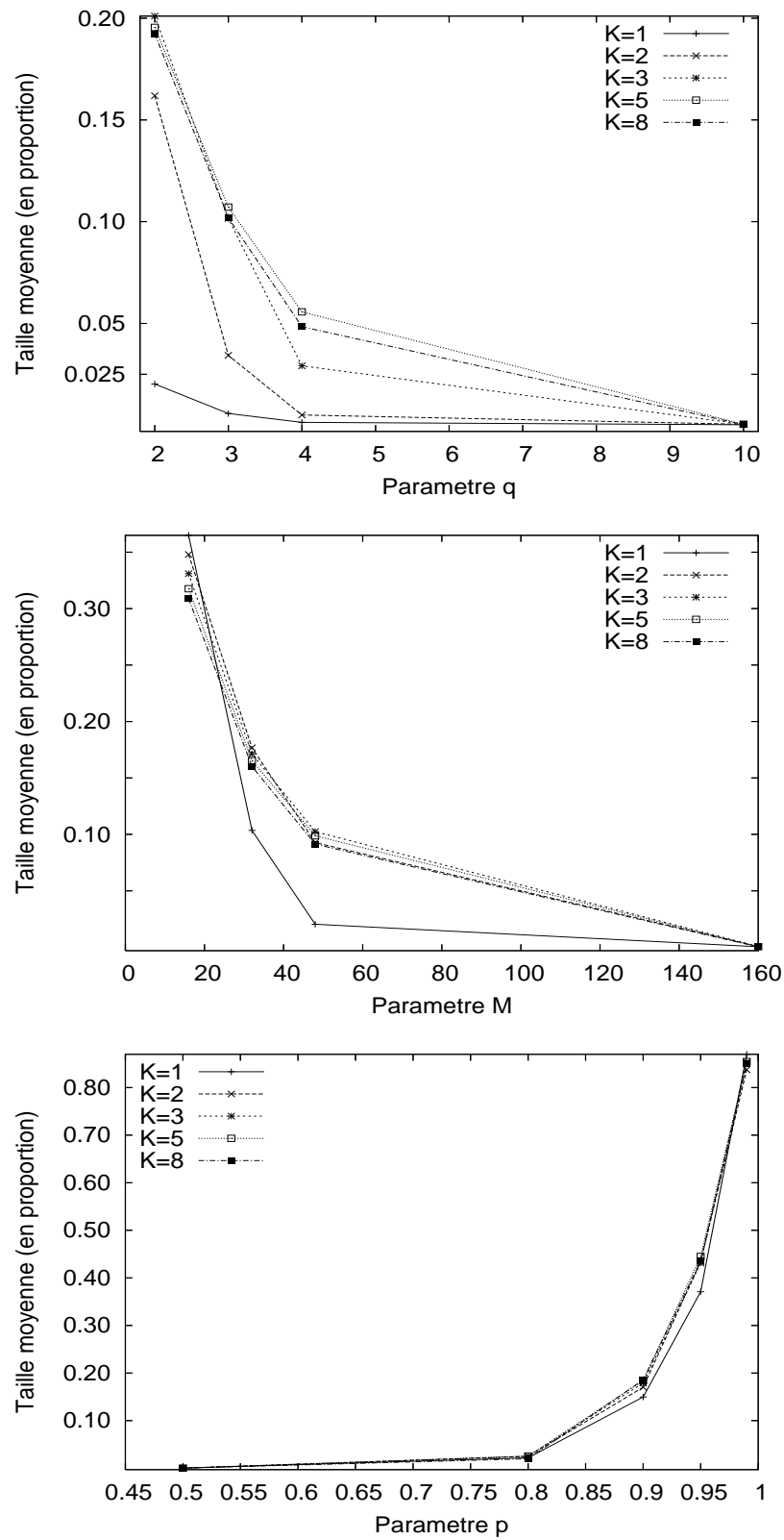


FIG. 3.15 – Taille moyenne sur les 50 instances de paysages du plus grand réseau de neutralité normalisée par la taille de l'espace de recherche de  $2^{16}$ . On trouve de haut en bas la famille des paysages  $NK_q$ ,  $NK_M$  et  $NK_p$ .

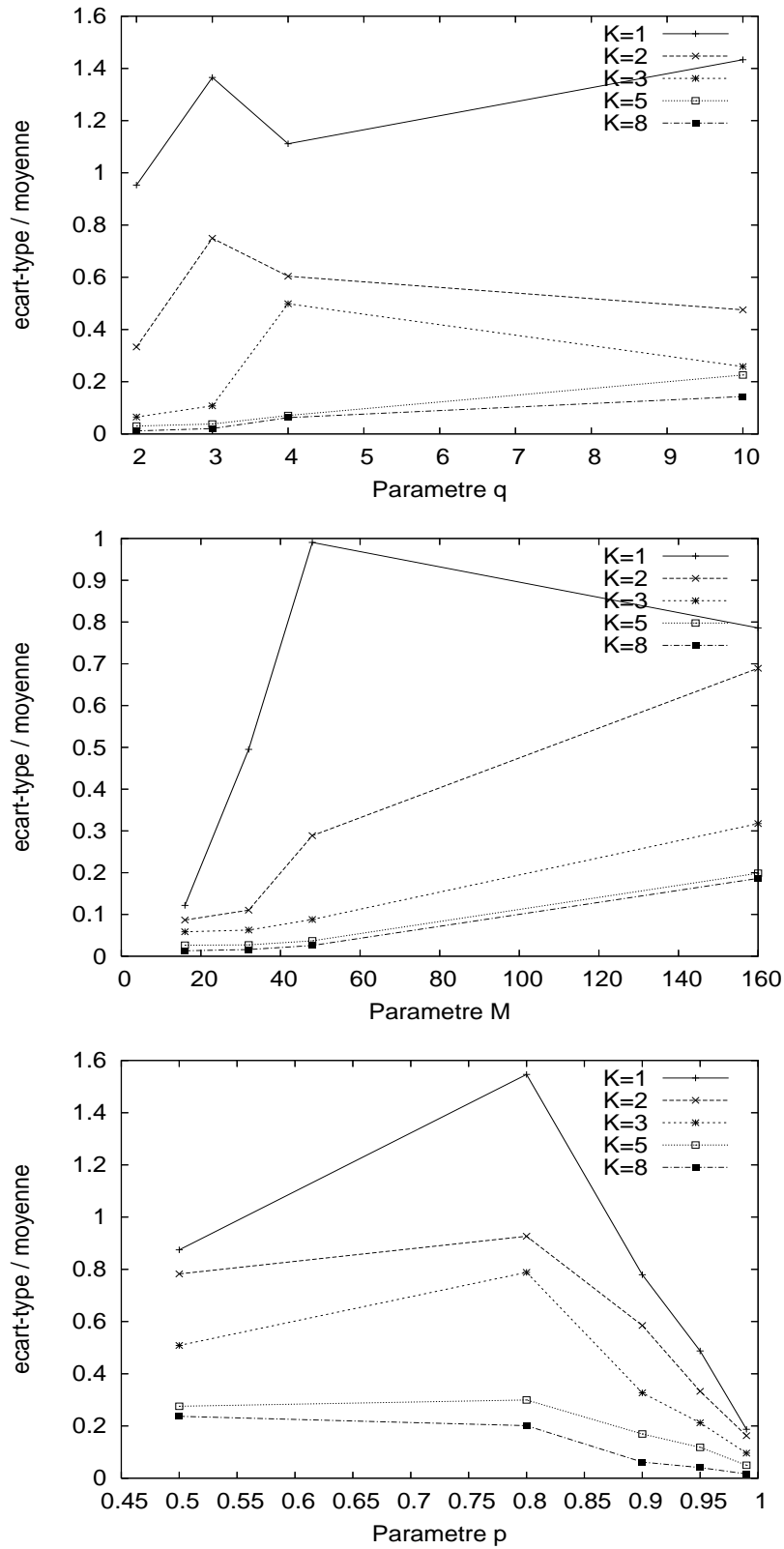


FIG. 3.16 – Rapport de l'écart-type par la moyenne de la taille du plus grand réseau de neutralité sur les 50 instances de paysages. On trouve de haut en bas la famille des paysages  $NK_q$ ,  $NK_M$  et  $NK_p$ .

### 3.2.3 Nombre de réseaux de neutralité

Dans cette sous-section, nous étudions le nombre de réseaux de neutralité par valeur de performance, pour  $N$  égal à 16, et nous comparons les concepts d'ensemble de neutralité et de réseau de neutralité.

La densité des états, noté D.O.S. (voir définition section 1.2.1), est le nombre de solutions par valeur de performance, autrement dit cette densité donne la taille des ensembles de neutralité. Les graphiques en bas à droite des figures 3.17, 3.18 et 3.19 présentent la D.O.S. pour les trois familles de paysages. Pour une valeur du paramètre de neutralité fixée, les D.O.S. sont quasiment égales, nous n'avons donc pas représenté les densités pour les différentes valeurs du paramètre  $K$ .

Pour l'ensemble des paysages, à l'exception des valeurs de performance optimale, les ensembles de neutralité contiennent plusieurs RN. Le nombre de RN varie en fonction du paramètre  $K$ , il augmente avec  $K$  pour une même valeur du paramètre de neutralité (deux exceptions pour les paysages  $NK_q$  pour  $K = 1$ ,  $q = 2$  et  $q = 3$ ). Pour une valeur du paramètre  $K$  fixée, le nombre de RN augmente lorsque le degré de neutralité moyen du paysage diminue. Les ensembles de neutralité et les RN ne partitionnent donc pas l'espace de recherche de la même manière. Lorsque  $K$  est grand, le nombre de RN est important, les notions d'ensemble et de réseau diffèrent le plus. Lorsque que la neutralité est plus importante, le nombre de RN est plus petit, les ensembles et les réseaux coïncident plus sans toutefois être identiques.

Le nombre de RN en fonction de la performance peut être interprété comme la distribution des valeurs de performance des RN. Cette distribution est généralement unimodale comme les D.O.S., le maximum de ces distributions étant situé à la même valeur de performance. Cependant, il existe certaines exceptions :

- pour les paysages  $NK_q$  : pour  $q = 2$ ,  $K = 8$  et  $q = 2$ ,  $K = 5$ ,
- pour les paysages  $NK_M$  : pour  $M = 32$ ,  $K = 8$  et pour  $M = 16$  quelque soit la valeur de  $K$ .

Dans ces cas, les distributions sont bimodales avec un minima pour la valeur de performance moyenne 0.5. Pour cette valeur de performance, les RN de neutralité sont les plus grands ce qui expliquerait le faible nombre de RN.

### Synthèse

Les ensembles et réseaux de neutralité ne définissent pas les mêmes ensembles de solutions. Un ensemble de neutralité contient plusieurs RN. Le nombre de RN par ensemble est plus grand lorsque le paramètre d'épistasie  $K$  est important et le paramètre de neutralité a une valeur qui diminue le degré de neutralité moyen du paysage.

Pour les variantes des paysages additifs étudiés, discrétiser les valeurs prises par la somme donne un nombre de RN moins important que discrétiser les termes de la somme ou rendre nulle certains termes de la somme.

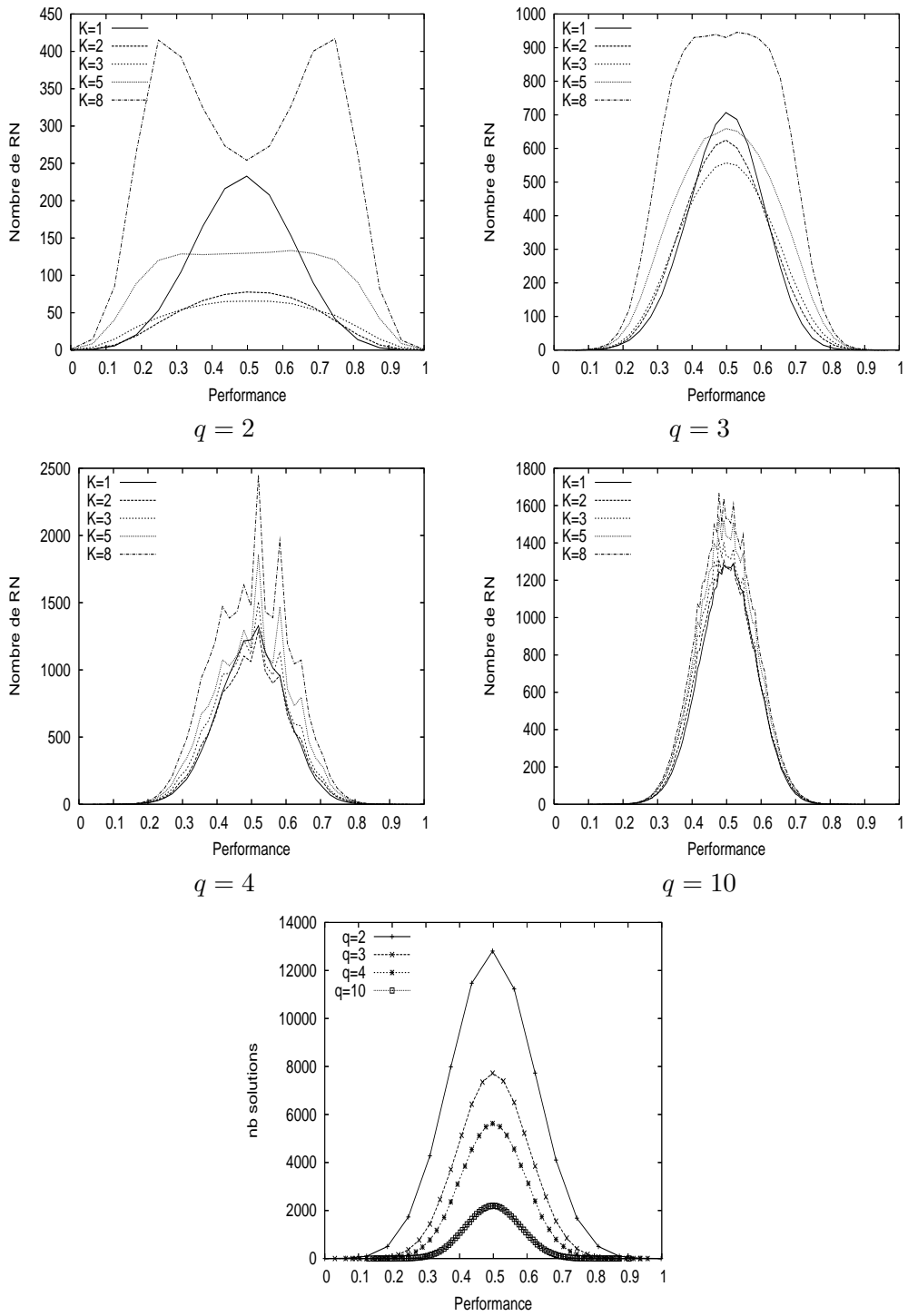


FIG. 3.17 – Nombre de réseaux de neutralité par valeur de performance (les quatre graphiques supérieurs) et densité des états (graphique inférieur) pour différentes valeurs des paramètres pour les paysages  $NK_q$ .

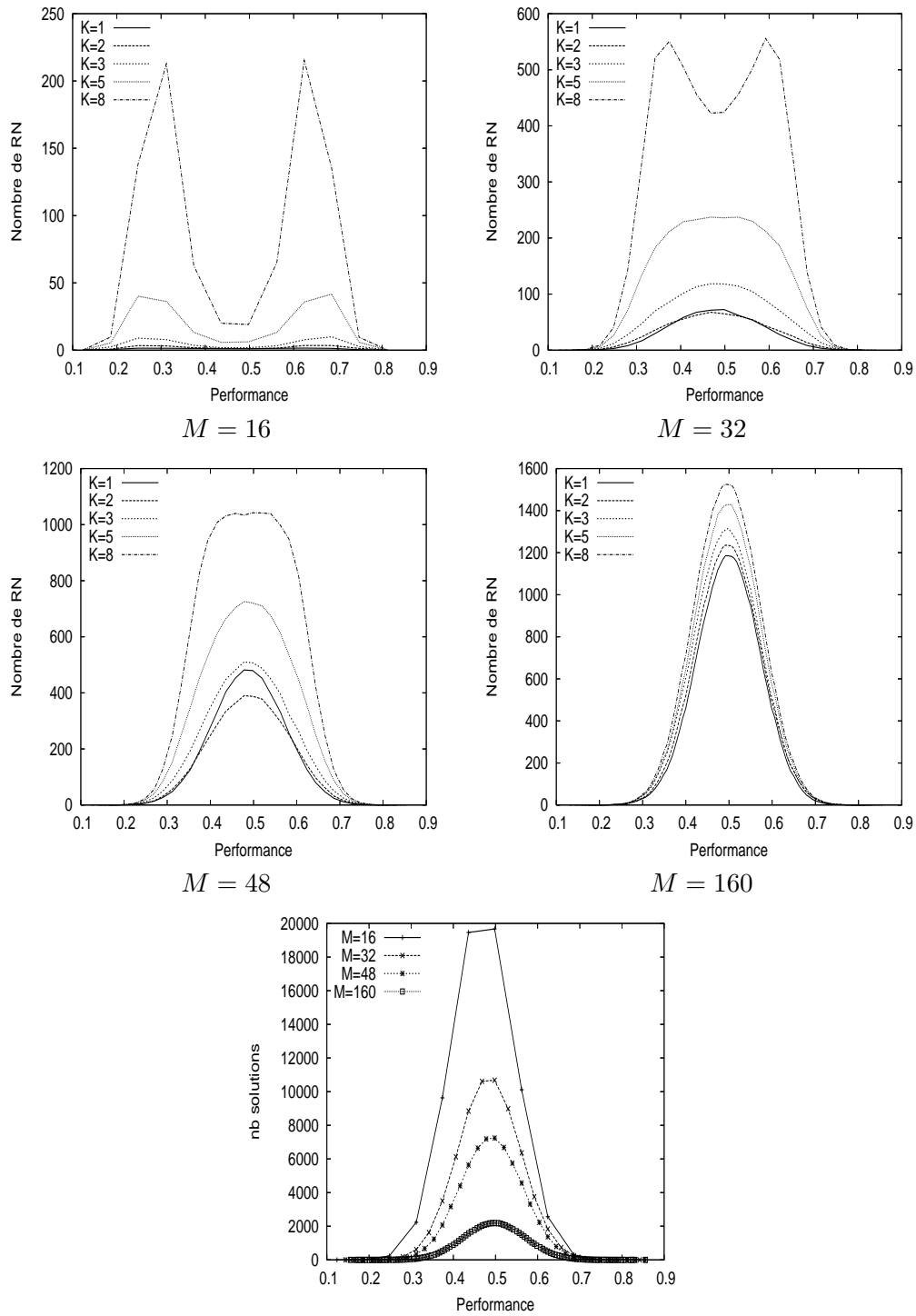


FIG. 3.18 – Nombre de réseaux de neutralité par valeur de performance (les quatre graphiques supérieurs) et densité des états (graphique inférieur) pour différentes valeurs des paramètres pour les paysages  $NK_M$ .

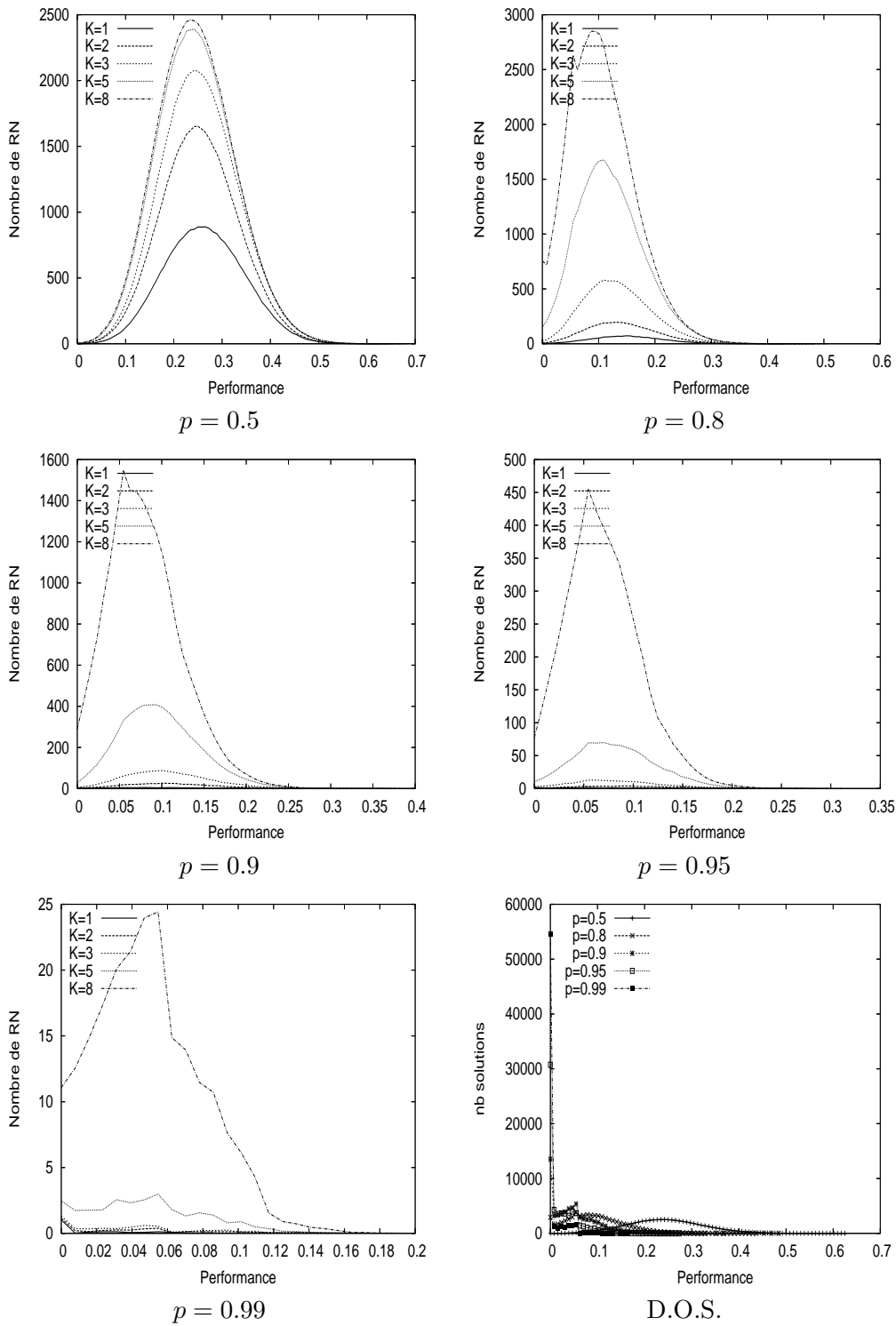


FIG. 3.19 – Nombre de réseaux de neutralité par valeur de performance (les quatre graphiques supérieurs et en bas à gauche) et densité des états (graphique en bas à droite) pour différentes valeurs des paramètres pour les paysages  $NK_p$ .



### 3.2.4 Taux d'innovation

Dans cette sous-section, nous étudions le taux d'innovation défini dans la section 3.1 pour les trois familles de paysages. Le taux d'innovation est le nombre de nouvelles valeurs de performance accessibles dans le voisinage des solutions obtenues au cours d'une marche neutre sur un RN. Nous donnerons, comme dans [60], l'*innovation cumulative neutre* noté  $C_n(t)$  *i.e.* le nombre de valeurs de performance différentes atteignables dans le voisinage au cours d'une marche neutre de longueur  $t$ . Nous comparons ce nombre à l'innovation cumulative aléatoire, noté  $C_a(t)$ , au cours d'une marche aléatoire de longueur  $t$ . La figure 3.20 montre des exemples de courbes représentatives de  $C_n$  et  $C_a$  pour un paysage  $NK_q$  et  $NK_M$ .

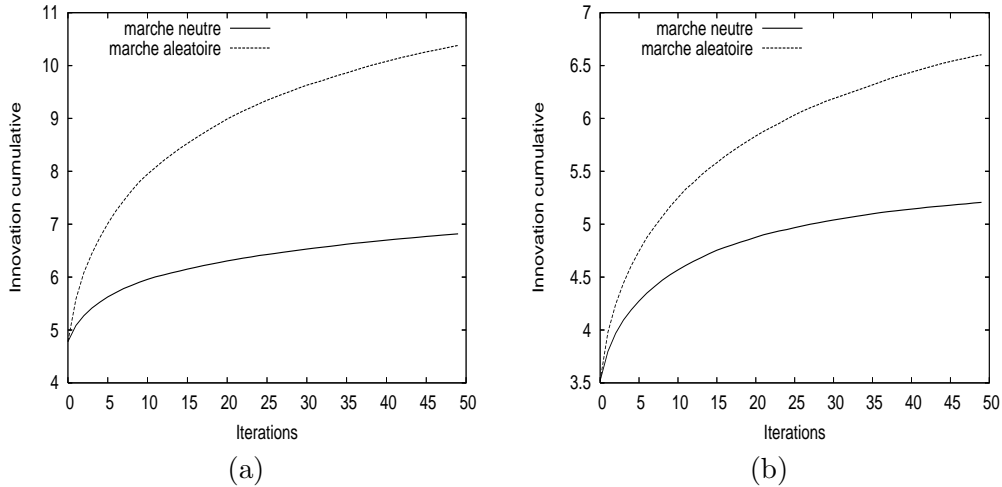


FIG. 3.20 – Exemple de courbes d'innovation cumulative lors d'une marche neutre sur un RN ( $C_n$ ), et lors d'une marche aléatoire dans l'ensemble du paysage ( $C_a$ ) pour un paysage  $NK_q$  avec  $K = 2$  et  $q = 2$  (a) et pour un paysage  $NK_M$  avec  $K = 2$  et  $M = 16$  (b).

Nous avons étudié les mêmes valeurs de paramètre que dans les sections précédentes. Pour chaque instance de paysage,  $10^2$  solutions initiales sont choisies uniformément dans le paysage parmi les solutions appartenant aux RN de taille supérieure à 50 solutions. A partir de chaque solution initiale, 10 marches indépendantes sont effectuées. Nous réalisons ainsi  $10^3$  marches neutres et  $10^3$  marches aléatoires de longueur 150. Dans les instances où aucun RN n'est de taille supérieure à 50, aucune marche n'est effectuée.

Les courbes des innovations cumulatives obtenues semblent être des courbes représentatives de fonctions puissances comme dans l'exemple 3.20. Nous avons donc calculé la régression linéaire de ces courbes en échelle logarithmique en abscisse et en ordonnée. On note respectivement  $a_n$  et  $a_a$  les coefficients directeurs, et  $b_n$  et  $b_a$  les ordonnées à l'origine des régressions obtenues de l'innovation cumulative neutre et aléatoire. Les tableaux 3.5, 3.6 et 3.7 donnent les résultats de ces expériences pour les trois paysages. Afin de comparer les innovations cumulatives aléatoire et neutre, nous définissons le nombre  $\tau = \frac{a_a}{a_n}$  égale au rapport des coefficients directeurs.

Pour l'ensemble des paysages, les coefficients de corrélation sont plus grand que 0.95, hormis pour l'innovation cumulative neutre avec  $K = 1$  où les coefficients sont un peu plus petit (0.82 pour  $p = 0.5$  pour le plus petit). Les données sont donc fortement linéairement corrélées. Statistiquement, les fonctions d'innovation cumulative  $C_n$  et  $C_a$  sont donc des fonctions puissances d'expression  $C(t) = \exp(b)t^a$ . Les taux d'innovation qui sont le nombre de nouvelles valeurs par itération sont donc la dérivée de  $C$  égale à  $\exp(b) a t^{a-1}$ . Ces taux sont décroissants.

Pour les paysages  $NK_q$ , pour une valeur de  $K$  fixée, lorsque  $q$  augmente, les coefficients

TAB. 3.5 – Régressions linéaires des innovations cumulatives neutre et aléatoire  $\log(C_n(t)) = a_n \log(t) + b_n$  et  $\log(C_a(t)) = a_a \log(t) + b_a$  pour les paysages  $NK_q$  avec  $N = 16$ . Le coefficient de corrélation est noté  $\rho$ .

K	q	Marche Neutre			Marche aléatoire			$\tau$
		$\rho$	$a_n$	$b_n$	$\rho$	$a_a$	$b_a$	
1	2	0.967	0.03296	1.4428	0.989	0.16994	1.5271	5.16
	3	0.940	0.03764	1.8105	0.988	0.19035	1.8997	5.06
	4	0.934	0.04566	1.9867	0.984	0.20865	2.1060	4.57
	10	0.907	0.07169	2.3947	0.969	0.25710	2.6370	3.59
2	2	0.999	0.07973	1.5908	0.988	0.15526	1.7223	1.95
	3	0.993	0.08385	1.9433	0.985	0.17783	2.0792	2.12
	4	0.986	0.09823	2.1388	0.981	0.19695	2.2954	2.00
	10	0.968	0.17972	2.5657	0.953	0.25109	2.8805	1.40
3	2	0.999	0.09645	1.7038	0.986	0.14565	1.8324	1.51
	3	0.995	0.10073	2.0771	0.982	0.16498	2.2087	1.64
	4	0.987	0.10929	2.2830	0.977	0.18267	2.4256	1.67
	10	0.962	0.18297	2.7147	0.952	0.25292	2.9657	1.38
5	2	0.996	0.09996	1.8773	0.981	0.12769	1.9746	1.28
	3	0.990	0.10898	2.2390	0.977	0.14868	2.3450	1.36
	4	0.981	0.11339	2.4548	0.970	0.16447	2.5642	1.45
	10	0.962	0.16933	2.9462	0.949	0.23311	3.1125	1.38
8	2	0.991	0.09913	2.0077	0.978	0.11204	2.0751	1.13
	3	0.985	0.10875	2.3589	0.971	0.13290	2.4413	1.22
	4	0.971	0.11449	2.5753	0.962	0.14998	2.6568	1.31
	10	-	-	-	-	-	-	-

$a_n$  et  $a_a$  augmentent ainsi que les coefficients  $b_n$  et  $b_a$ . Ce qui s'explique probablement par le nombre de valeurs de performance qui augmente avec  $q$ . En effet, celui-ci est de  $1 + N(q - 1)$ . Pour une valeur de  $q$  fixée, les coefficients  $b_n$  et  $b_a$  augmentent avec  $K$  et  $a_a$  décroît avec  $K$ . En revanche, le coefficient  $a_n$  a tendance à croître avec  $K$ . Lorsque  $K$  augmente, une solution a plus de valeurs de performance différentes dans son voisinage et les RN des paysages  $NK_q$  ont de meilleures capacités d'exploration de nouvelles valeurs de performance. Ceci est confirmé par la décroissance avec  $K$  du rapport  $\tau$ . La variation de  $q$  pour une valeur de  $K$  a moins d'influence sur  $\tau$  qu'une variation de  $K$ . Pour  $K$  égal à 1,  $\tau$  décroît avec l'augmentation de  $q$ ; pour les autres valeurs de  $K$ ,  $\tau$  varie peu.

Dans les paysages  $NK_M$ , pour une valeur de  $K$  fixée, les coefficients  $a_n$ ,  $a_a$ ,  $b_n$  et  $b_a$  augmentent comme le nombre de valeurs de performance  $M$  (une seule exception pour  $a_n$  quand  $K = 1$  et  $M$  entre 32 et 48). Pour une valeur de  $M$  fixée, les coefficients  $b_n$  et  $b_a$  augmentent avec  $K$  et  $a_a$  décroît avec  $K$ . Le coefficient  $a_n$  augmente entre  $K = 1$  et  $K = 3$ , puis diminue entre  $K = 3$  et  $K = 8$ . Lorsque  $K$  augmente, une solution a plus de valeurs de performance différentes dans son voisinage, ce qui explique l'accroissement de  $a_a$ ,  $b_a$  et  $b_n$ . La capacité de découverte de nouvelles valeurs de performance des RN ne varie pas de la même façon, dans un premier temps avec l'augmentation de  $K$  les RN augmentent leur capacité de découverte qui ensuite atteint une limite pour  $K = 3$ . Le rapport  $\tau$  permet de comparer les capacités de découverte des RN par rapport à l'ensemble de l'espace de recherche. Pour une valeur de  $M$  fixée,  $\tau$  décroît lorsque  $K$  augmente. Relativement à l'ensemble des solutions de l'espace de

TAB. 3.6 – Régressions linéaires des innovations cumulatives neutre et aléatoire  $\log(C_n(t)) = a_n \log(t) + b_n$  et  $\log(C_a(t)) = a_a \log(t) + b_a$  pour les paysages  $NK_M$  avec  $N = 16$ . Le coefficient de corrélation est noté  $\rho$ .

K	M	Marche Neutre			Marche aléatoire			$\tau$
		$\rho$	$a_n$	$b_n$	$\rho$	$a_a$	$b_a$	
1	16	0.997	0.08265	1.1509	0.991	0.14799	1.1928	1.79
	32	0.989	0.06392	1.6550	0.989	0.16853	1.7132	2.64
	48	0.979	0.06158	1.9252	0.987	0.18648	2.0025	3.03
	160	0.944	0.09198	2.4895	0.974	0.25501	2.6560	2.77
2	16	0.997	0.08928	1.2779	0.991	0.14009	1.3310	1.57
	32	0.998	0.09345	1.7607	0.987	0.15868	1.8655	1.70
	48	0.993	0.09398	2.0482	0.984	0.17555	2.1606	1.87
	160	0.977	0.16187	2.6391	0.965	0.25975	2.8705	1.60
3	16	0.998	0.09553	1.3672	0.990	0.13328	1.4304	1.40
	32	0.998	0.10496	1.8598	0.985	0.15105	1.9703	1.44
	48	0.994	0.10749	2.1492	0.981	0.16708	2.2688	1.55
	160	0.974	0.17528	2.7480	0.960	0.25402	2.9605	1.45
5	16	0.996	0.09498	1.4931	0.987	0.11886	1.5530	1.25
	32	0.995	0.10671	1.9985	0.982	0.13689	2.0907	1.28
	48	0.989	0.11015	2.2939	0.976	0.15205	2.3926	1.38
	160	0.965	0.16420	2.8852	0.952	0.24243	3.0682	1.48
8	16	0.994	0.09370	1.6043	0.982	0.10235	1.6548	1.09
	32	0.991	0.10521	2.1169	0.979	0.12140	2.1875	1.15
	48	0.982	0.11064	2.4028	0.970	0.13798	2.4813	1.25
	160	-	-	-	-	-	-	-

recherche, les RN augmentent leur capacité de découverte lorsque  $K$  augmente. La variation de  $M$  pour une valeur de  $K$  fixée a moins d'influence sur la valeur de  $\tau$  que la variation de  $K$ .  $\tau$  augmente entre  $M = 16$  et  $M = 48$ , *i.e.* avec la diminution du degré de neutralité moyen. Pour  $M = 160$ ,  $\tau$  diminue par rapport à  $M = 48$ .

Pour les paysages  $NK_p$ , pour une valeur de  $K$  fixée, lorsque  $p$  augmente *i.e.* le degré de neutralité moyen du paysage augmente, les coefficients  $b_n$  et  $b_a$  décroissent jusqu'à une valeur faible d'environ 0.5. Les coefficients  $a_n$  et  $a_a$  sont aussi décroissants. Ce qui s'explique peut-être par le nombre de valeurs de performance qui diminue lorsque  $p$  augmente. Pour une valeur de  $p$  fixée entre  $p = 0.8$  et  $p = 0.95$ , les coefficients  $b_n$  et  $b_a$  augmentent avec de  $K$  et pour  $p = 0.99$ , ces coefficients restent constants. Les coefficients  $a_n$  et  $a_a$  croissent également avec  $K$ . Le coefficient  $a_a$  croît moins vite que le coefficient  $a_n$  puisque le rapport  $\tau$  diminue lorsque  $K$  augmente pour une valeur de  $p$  fixée. Lorsque  $K$  augmente, une solution a plus de valeurs de performance différentes dans son voisinage et les RN des paysages  $NK_p$  ont de meilleures capacités d'exploration de nouvelles valeurs de performance. La variation de  $p$  pour une valeur de  $K$  fixée a moins d'influence sur  $\tau$  que la variation de  $K$ . Pour  $K$  fixé,  $\tau$  décroît fortement avec l'augmentation de  $p$ . Le paramètre de neutralité  $p$  est prépondérant sur le paramètre  $K$  quant à la valeur du rapport  $\tau$ .

TAB. 3.7 – Régressions linéaires des innovations cumulatives neutre et aléatoire  $\log(C_n(t)) = a_n \log(t) + b_n$  et  $\log(C_a(t)) = a_a \log(t) + b_a$  pour les paysages  $NK_p$  avec  $N = 16$ . Le coefficient de corrélation est noté  $\rho$ .

K	p	Marche Neutre			Marche aléatoire			$\tau$
		$\rho$	$a_n$	$b_n$	$\rho$	$a_a$	$b_a$	
1	0.5	0.821	0.06652	2.6677	0.999	0.99015	2.0537	14.88
	0.8	0.939	0.06027	2.3013	0.999	0.78206	1.9380	12.98
	0.9	0.949	0.05828	1.8182	0.997	0.50506	1.6642	8.67
	0.95	0.953	0.05341	1.3473	0.990	0.28343	1.3169	5.31
	0.99	0.948	0.02213	0.3850	0.910	0.03440	0.4109	1.55
2	0.5	-	-	-	-	-	-	-
	0.8	0.982	0.18358	2.4237	0.999	0.87603	2.0118	4.77
	0.9	0.991	0.17782	1.9815	0.999	0.67088	1.7178	3.77
	0.95	0.989	0.15316	1.4487	0.998	0.46591	1.2604	3.04
	0.99	0.956	0.07438	0.6005	0.992	0.16189	0.5196	2.18
3	0.5	-	-	-	-	-	-	-
	0.8	0.997	0.33790	2.4505	0.999	0.93410	2.1284	2.76
	0.9	0.998	0.32052	2.0209	0.999	0.77395	1.7886	2.41
	0.95	0.997	0.28397	1.5531	0.999	0.59556	1.3648	2.10
	0.99	0.976	0.15054	0.6089	0.993	0.23303	0.5112	1.55
5	0.5	-	-	-	-	-	-	-
	0.8	1.000	0.57227	2.4869	1.000	0.96949	2.3636	1.69
	0.9	1.000	0.55358	2.0833	1.000	0.87582	1.9804	1.58
	0.95	0.999	0.51204	1.6121	0.999	0.75018	1.4881	1.47
	0.99	0.995	0.38616	0.5748	0.997	0.46643	0.4859	1.21
8	0.5	-	-	-	-	-	-	-
	0.8	1.000	0.64800	2.6792	1.000	0.97025	2.5617	1.50
	0.9	0.999	0.82032	2.1225	1.000	0.94084	2.2060	1.15
	0.95	0.999	0.71993	1.7767	0.999	0.87738	1.7284	1.22
	0.99	0.998	0.67755	0.5297	0.997	0.72897	0.4948	1.08

TAB. 3.8 – Régressions linéaires des innovations cumulatives neutre et aléatoire  $\log(C_n(t)) = a_n \log(t) + b_n$  et  $\log(C_a(t)) = a_a \log(t) + b_a$  pour les paysages  $NK_q$  avec  $N = 64$  Le coefficient de corrélation est noté  $\rho$ .

K	q	Marche Neutre			Marche aléatoire			$\tau$
		$\rho$	$a_n$	$b_n$	$\rho$	$a_a$	$b_a$	
2	2	0.993	0.06182	1.8312	0.999	0.23081	1.8649	3.73
	3	0.997	0.06076	2.2580	0.999	0.23763	2.3063	3.91
	4	0.999	0.05237	2.5275	0.999	0.24447	2.5772	4.67
	10	0.935	0.01993	3.2637	0.997	0.27077	3.3547	13.58
4	2	0.998	0.08482	2.0481	0.998	0.20945	2.1233	2.47
	3	0.999	0.09190	2.4585	0.998	0.21871	2.5497	2.38
	4	1.000	0.09594	2.7116	0.997	0.22527	2.8197	2.35
	10	0.987	0.05967	3.4257	0.993	0.25368	3.5818	4.25
8	2	0.999	0.09171	2.3297	0.994	0.17906	2.4080	1.95
	3	0.998	0.09983	2.7492	0.993	0.18790	2.8337	1.88
	4	0.997	0.10629	3.0005	0.992	0.19551	3.0955	1.84
	10	-	-	-	-	-	-	-
12	2	0.997	0.09077	2.5023	0.990	0.15602	2.5799	1.72
	3	0.995	0.10029	2.9095	0.988	0.16617	2.9972	1.66
	4	0.995	0.10641	3.1538	0.986	0.17349	3.2584	1.63
	10	0.994	0.10145	3.6839	0.974	0.20646	3.9848	2.04
16	2	0.995	0.09103	2.6093	0.987	0.14089	2.6855	1.55
	3	0.994	0.10006	3.0129	0.984	0.15079	3.1022	1.51
	4	0.993	0.10624	3.2491	0.982	0.15834	3.3609	1.49
	10	0.992	0.09573	3.7345	0.967	0.19323	4.0745	2.02

### Influence du paramètre $N$

Afin de mettre en évidence l'influence de la taille de l'espace de recherche, nous avons étudié les innovations cumulatives lorsque  $N$  est égal à 64, et nous exposons les principaux résultats.

Pour les variantes de paysages, les fonctions innovations cumulatives sont également des fonctions puissances, les coefficients de corrélation sont tous significatifs. Pour une valeur de  $K$  fixée, les coefficients  $b_n$  et  $b_a$  ainsi que le coefficient  $a_a$  augmentent avec la diminution du degré moyen du paysage, comme pour  $N = 16$ . Pour une valeur du paramètre de neutralité, les variations de  $a_n$  reste inchangée. En revanche, pour une valeur du paramètre  $K$  fixée, les variations de  $a_n$  et de  $\tau$  diffèrent avec  $N = 16$ . Pour les paysages  $NK_q$ ,  $\tau$  augmente avec  $q$  lorsque  $K = 4$  ou  $K = 2$  et est quasiment constant ou diminue avec  $q$  pour  $K$  égale à 8, 12 ou 16. alors qu'il augmentait avec  $q$  pour  $N = 16$ . L'ordre de grandeurs de  $\tau$  reste le même entre  $N = 16$  et  $N = 64$ . Pour les paysages  $NK_M$ , pour  $N = 16$ ,  $a_n$  augmente avec  $q$  alors que pour  $N = 64$ ,  $a_n$  est maximal pour  $q = 4$ . En conséquence, pour  $N = 64$ , les variations de  $\tau$  ne sont plus constantes mais conserve le même ordre de grandeur que dans le cas  $N = 16$ . Pour les paysages  $NK_p$ , les variations de  $a_n$  sont différentes mais les variations  $\tau$  est reste les mêmes (décroissante lorsque  $p$  augmente).

TAB. 3.9 – Régressions linéaires des innovations cumulatives neutre et aléatoire  $\log(C_n(t)) = a_n \log(t) + b_n$  et  $\log(C_a(t)) = a_a \log(t) + b_a$  pour les paysages  $NK_M$  avec  $N = 64$ . Le coefficient de corrélation est noté  $\rho$ .

K	M	Marche Neutre			Marche aléatoire			$\tau$
		$\rho$	$a_n$	$b_n$	$\rho$	$a_a$	$b_a$	
2	16	0.993	0.06182	1.8312	0.999	0.23081	1.8649	3.73
	32	0.997	0.06076	2.2580	0.999	0.23763	2.3063	3.91
	48	0.999	0.05237	2.5275	0.999	0.24447	2.5772	4.67
	160	0.935	0.01993	3.2637	0.997	0.27077	3.3547	13.58
4	16	0.998	0.08482	2.0481	0.998	0.20945	2.1233	2.47
	32	0.999	0.09190	2.4585	0.998	0.21871	2.5497	2.38
	48	1.000	0.09594	2.7116	0.997	0.22527	2.8197	2.35
	160	0.987	0.05967	3.4257	0.993	0.25368	3.5818	4.25
8	16	0.999	0.09171	2.3297	0.994	0.17906	2.4080	1.95
	32	0.998	0.09983	2.7492	0.993	0.18790	2.8337	1.88
	48	0.997	0.10629	3.0005	0.992	0.19551	3.0955	1.84
	160	-	-	-	-	-	-	-
12	16	0.997	0.09077	2.5023	0.990	0.15602	2.5799	1.72
	32	0.995	0.10029	2.9095	0.988	0.16617	2.9972	1.66
	48	0.995	0.10641	3.1538	0.986	0.17349	3.2584	1.63
	160	-	-	-	-	-	-	-
16	16	0.995	0.09103	2.6093	0.987	0.14089	2.6855	1.55
	32	0.994	0.10006	3.0129	0.984	0.15079	3.1022	1.51
	48	0.993	0.10624	3.2491	0.982	0.15834	3.3609	1.49
	160	-	-	-	-	-	-	-

TAB. 3.10 – Régressions linéaires des innovations cumulatives neutre et aléatoire  $\log(C_n(t)) = a_n \log(t) + b_n$  et  $\log(C_a(t)) = a_a \log(t) + b_a$  pour les paysages  $NK_p$  avec  $N = 64$ . Le coefficient de corrélation est noté  $\rho$ .

K	p	Marche Neutre			Marche aléatoire			$\tau$
		$\rho$	$a_n$	$b_n$	$\rho$	$a_a$	$b_a$	
2	0.5	0.832	0.01410	4.1558	1.000	0.98840	4.0433	70.08
	0.8	0.995	0.15373	3.6139	0.999	0.95774	3.4651	6.23
	0.9	0.989	0.19253	2.9948	0.998	0.90323	2.7167	4.69
	0.95	0.987	0.18768	2.3623	0.993	0.79534	1.9055	4.24
	0.99	0.987	0.14540	1.0715	0.983	0.39228	0.7419	2.70
4	0.5	-	-	-	-	-	-	-
	0.8	0.998	0.22497	3.8894	1.000	0.97858	3.8401	4.35
	0.9	0.989	0.37035	3.1318	0.999	0.94583	3.2035	2.55
	0.95	0.986	0.38424	2.5026	0.997	0.87784	2.4222	2.28
	0.99	0.984	0.34050	1.1295	0.986	0.58416	0.8872	1.72
8	0.5	-	-	-	-	-	-	-
	0.8	-	-	-	-	-	-	-
	0.9	0.994	0.53541	3.4657	1.000	0.97529	3.7260	1.82
	0.95	0.990	0.60887	2.7723	0.999	0.93491	3.0405	1.54
	0.99	0.985	0.59851	1.2846	0.992	0.75612	1.2959	1.26
12	0.5	-	-	-	-	-	-	-
	0.8	-	-	-	-	-	-	-
	0.9	0.997	0.54092	3.7715	1.000	0.98598	3.9531	1.82
	0.95	0.994	0.71327	3.0674	1.000	0.95928	3.3986	1.34
	0.99	0.990	0.73002	1.5207	0.995	0.83507	1.6411	1.14
16	0.5	-	-	-	-	-	-	-
	0.8	-	-	-	-	-	-	-
	0.9	0.995	0.48416	3.8257	1.000	0.99023	4.0532	2.05
	0.95	0.996	0.75684	3.3286	1.000	0.97088	3.6312	1.28
	0.99	0.993	0.80250	1.7806	0.997	0.87923	1.9278	1.10

## Synthèse

Les innovations cumulatives neutre et aléatoire sont des fonctions puissances dont les paramètres ont été donnés pour les trois variantes de paysage. Il est possible de déduire de ces fonctions les taux d'innovation. La comparaison entre l'innovation neutre et aléatoire permet d'en déduire que les RN de neutralité des paysages étudiés ont la possibilité de découvrir un grand nombre de valeurs de performance dans leur voisinage.

Cette modélisation par une fonction puissance permet d'estimer le nombre de nouvelles valeurs de performance dans le voisinage des solutions des RN. Cette information peut aider à la conception de métaheuristique adaptée. En effet, il est possible ajuster le temps de recherche sur un RN en fonction de la probabilité de trouver une nouvelle valeur de performance. Si cette probabilité est trop faible, la recherche sur un RN peut être arrêtée.

Généralement, les taux d'innovation neutre et aléatoire augmentent lorsque l'épistasie diminue et le degré de neutralité moyen du paysage augmente, mais le taux d'innovation neutre augmentent dans une moindre proportion. La taille de l'espace de recherche influence, dans une moindre mesure, le taux d'innovation.

La différence de taux d'innovation entre les différentes des variantes des paysages est faible au vu des résultats. les variantes ne se différencient pas par leur taux d'innovation.



### 3.2.5 Autocorrélation de l'évolvabilité maximale

Dans cette section, nous allons étudier sur les paysages  $NK_q$ ,  $NK_M$  et  $NK_p$  l'autocorrélation de l'évolvabilité définie section 3.1.2 relativement à l'opérateur HC (cf section 2.3.1) qui sélectionne une solution voisine dont la performance est la plus grande du voisinage. Nous ajoutons la condition que cette performance maximale doit être différente de celle de la solution initiale. Cette condition permet de garantir que la performance est hors du RN. Nous appellerons *évolvabilité maximale* cette mesure d'évolvabilité, elle permet d'étudier les RN de grande performance connecté à un RN donné.

Nous avons considéré les mêmes valeurs de paramètre que dans les sections précédentes. Pour chaque instance de paysage,  $10^2$  solutions initiales sont choisies uniformément dans le paysage parmi les solutions appartenant aux RN de taille supérieure à 50 solutions. A partir de chaque solution initiale, 10 marches neutres indépendantes sont effectuées. Dans les instances où aucun RN n'est de taille supérieure à 50, aucune marche n'est effectuée.

Les figures 3.21, 3.22 et 3.23 présentent les fonctions d'autocorrélation de l'évolvabilité maximale et la valeur du coefficient d'ordre 1 pour les paysages quand  $N = 16$ . Les résultats se commentent de la même manière que pour les fonctions d'autocorrélation des degrés de neutralité,

Pour les paysages  $NK_q$ , le coefficient d'autocorrélation  $\rho(1)$  d'ordre 1 décroît lorsque le paramètre de neutralité  $q$  augmente. Excepté pour  $K = 1$ , pour une valeur du paramètre  $q$  fixée,  $\rho(1)$  est décroissant avec  $K$ . Pour  $K = 1$ ,  $\rho(1)$  décroît fortement entre les valeur  $q = 3$  et  $q = 4$ . Pour les différentes valeurs des paramètres,  $\rho(1)$  est au dessus de 0.35, voir au dessus de 0.6 pour plus de la moitié des valeurs. La corrélation n'est pas nulle, les valeurs de performance au dessus des RN de neutralité ne sont pas distribuées aléatoirement. Les fonctions d'autocorrélation  $\rho$  sont de deux types. Lorsque le degré de neutralité moyen du paysage est plus grand, les fonctions d'autocorrélation sont des fonctions décroissantes (par exemple toutes les fonctions d'autocorrélation relative à  $K = 1$  et  $K = 2$ ). Lorsque le degré de neutralité moyen est plus faible, les fonctions d'autocorrélation alternent entre une plus grande et une plus faible valeur pour la fonction  $\rho$ . Pour  $K = 1$ , les fonctions d'autocorrélation ne décroissent pas jusqu'à une valeur proche de 0. En effet, pour cette valeur du paramètre  $K$ , les évolvabilités maximales sont quasiment constantes durant une marche neutre, à peu près 145 sur 150 pas de marche sont de même valeur d'évolvabilité. La corrélation d'évolvabilité entre deux solutions de la marche ne peut donc pas décroître. Pour les autres valeurs du paramètre  $K$ , le nombre de valeurs d'évolvabilité maximale est plus important et les fonctions d'autocorrélation décroissent.

Pour les paysages  $NK_M$ , les fonctions d'autocorrélation sont toutes du second type qui alternent une valeur haute et une valeur basse d'autocorrélation. Les fonctions décroissent vers la valeur nulle. Pour une valeur du paramètre  $M$  fixée, le coefficient d'autocorrélation  $\rho(1)$  décroît lorsque  $K$  augmente. Pour une valeur de  $K$  fixée,  $\rho(1)$  atteint un maximum pour la valeur intermédiaire de  $M$  égale à 32. La corrélation entre performance maximale n'est pas nulle. Toutefois, la corrélation est plus petite que dans le cas des paysages  $NK_q$  et reste comprise entre 0.35 et 0.68.

Pour les paysages  $NK_p$ , les fonctions d'autocorrélation sont majoritairement du premier type où les fonctions sont décroissantes. Les deux fonctions pour  $p = 0.99$ ,  $K = 1$  et  $K = 2$  sont toujours supérieures à 0.6 et 0.3 respectivement. Ceci s'explique de nouveau par la constance de l'évolvabilité maximale au cours de la marche neutre. Le coefficient  $\rho(1)$  est croissant avec le paramètre  $p$  excepté pour  $K = 8$  où un maximum est atteint pour la valeur  $p = 0.9$ .  $\rho(1)$  est toujours supérieur à 0.4. La corrélation n'est pas nulle, les valeurs de performance ne sont pas réparties aléatoirement autour des RN. La variation de  $\rho(1)$  en fonction du paramètre  $K$  n'est pas constante, en effet les valeurs pour  $K = 1$  coupent l'ensemble des courbes. Cependant, hors mis pour  $K = 1$ ,  $\rho(1)$  est décroissant avec  $K$ .

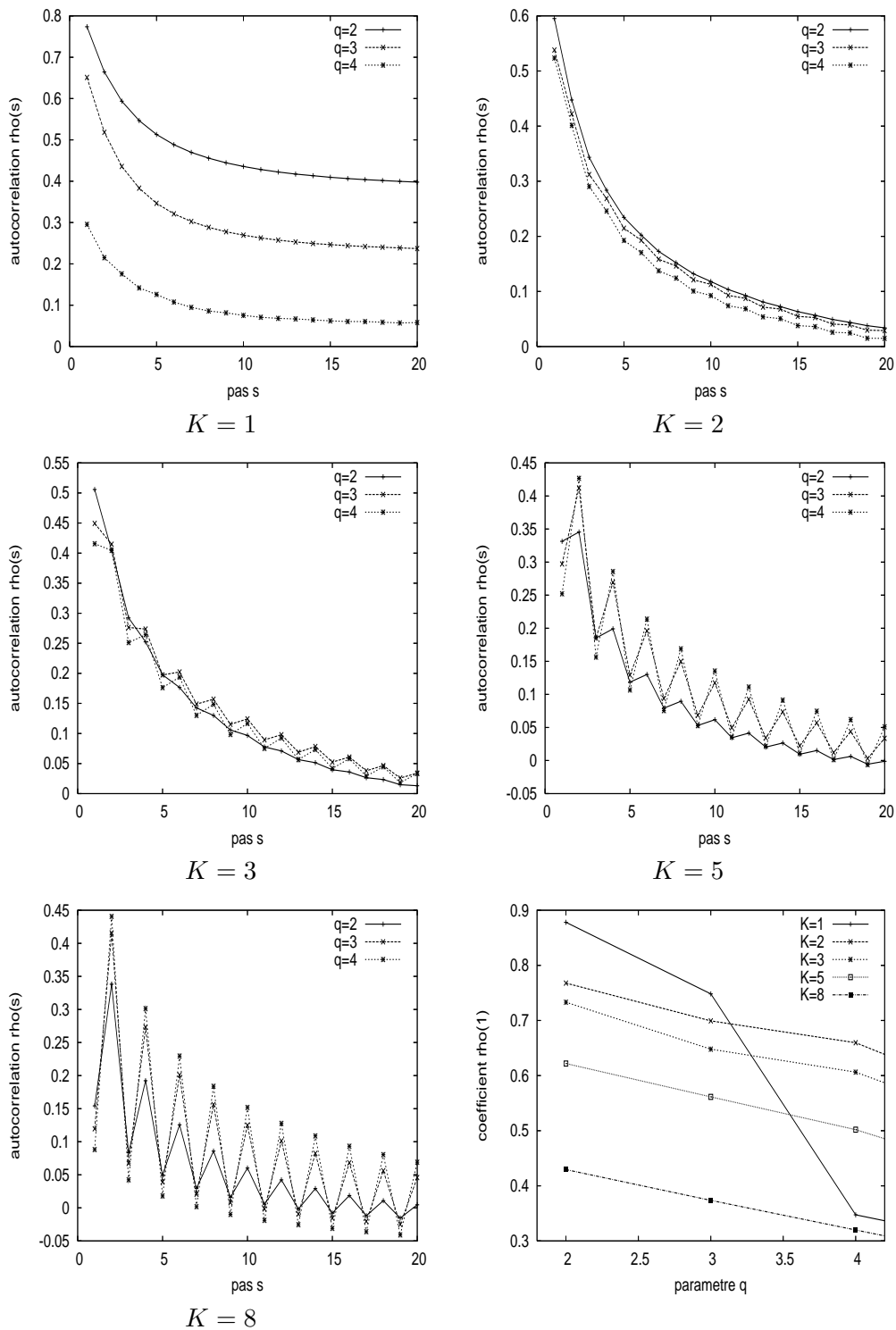


FIG. 3.21 – Fonctions d'autocorrélation de l'évolvabilité maximale pour différentes valeurs du paramètre  $K$  et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages  $NK_q$  avec  $N = 16$ .

### Influence du paramètre $N$

Afin de mesurer l'influence de la taille de l'espace de recherche, nous avons étudiés les fonctions d'autocorrélation de l'évolvabilité maximale pour  $N = 64$ . Nous avons effectué les

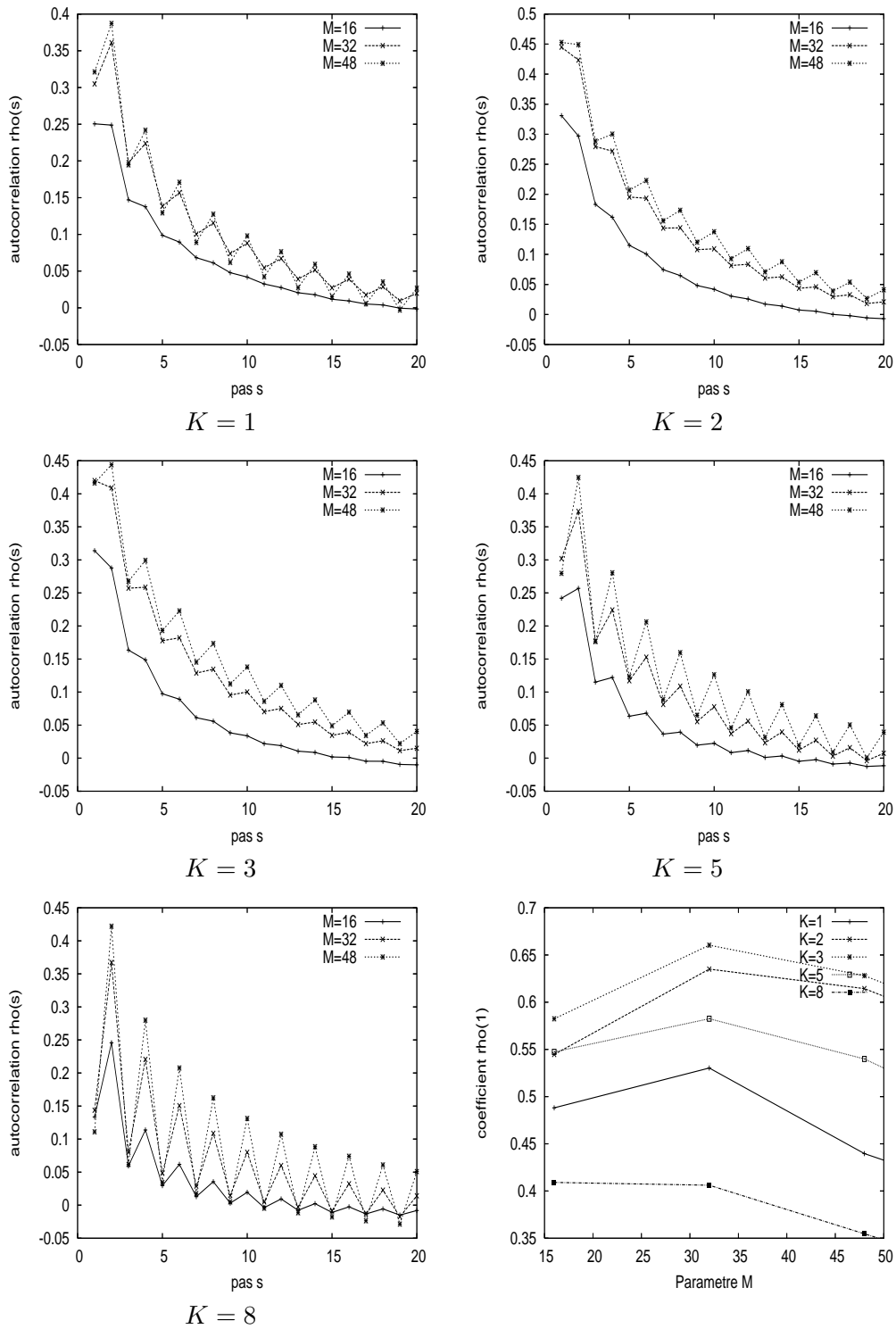


FIG. 3.22 – Fonctions d'autocorrélation de l'évolvabilité maximale pour différentes valeurs du paramètre  $K$  et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages  $NK_M$  avec  $N = 16$ .

marches neutres de la même façon que dans la section 3.2.1. La figure 3.24 présente le coefficient d'autocorrélation d'ordre 1 de l'évolvabilité maximale pour  $N = 64$ .

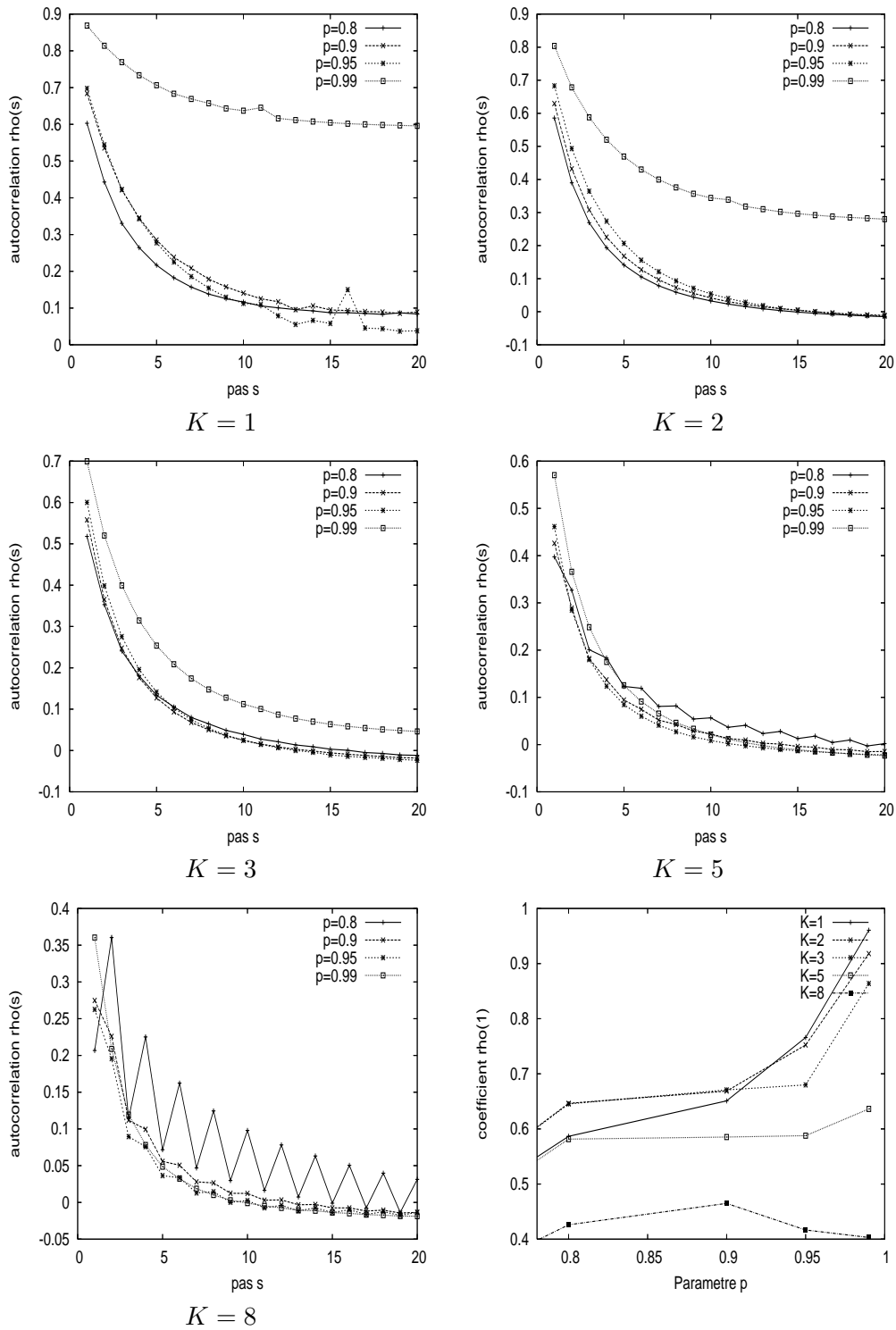


FIG. 3.23 – Fonctions d'autocorrélation de l'évolvabilité maximale pour différentes valeurs du paramètre  $K$  et coefficient d'autocorrélation d'ordre 1 (en bas à droite) pour les paysages  $NK_p$  avec  $N = 16$ .

Pour les paysages  $NK_q$ , les variations de  $\rho(1)$  sont les mêmes que pour  $N = 16$ .  $\rho(1)$  est plus faible pour les plus grandes valeur de  $K$  que précédemment. Pour les paysages  $NK_M$ , les

valeurs de  $\rho(1)$  sont aussi plus petites que pour  $N = 16$ . Alors que la valeur  $M = 32$  est un maximum pour  $N = 16$ , la valeur  $M = 32$  est un minimum pour  $N = 64$ . Nous n'avons pas d'explication sur cette différence. Pour les paysages  $NK_p$ , la valeur  $p = 0.9$  est un minimum pour  $\rho(1)$  pour une valeur de  $K$  fixée à  $K = 12$  ou à  $K = 16$ . Les valeurs de  $\rho(1)$  sont du même ordre de grandeur que pour  $N = 16$ .

## Synthèse

Les corrélations d'évolvabilité maximale entre solutions voisines dans un réseau de neutralité ne sont pas nulles. Elle est la plus forte pour les paysages  $NK_p$ . Cette corrélation peut être la conséquence du faible nombre de valeurs de l'évolvabilité maximale lorsque le degré moyen de neutralité du paysage est important. Le degré de neutralité moyen n'est pas directement lié à la présence de corrélation puisque les paysages  $NK_q$  et  $NK_M$  ont des degrés de neutralité différents et des corrélations du même ordre.

Pour les variantes des paysages additifs étudiés, la corrélation de l'évolvabilité est plus forte lorsque certains termes de la somme sont discrétisés ou rendu nuls que lorsque la somme elle-même est discrétisée. De plus, l'influence du paramètre d'épistasie est plus marquée dans le premier cas. L'augmentation de la taille de l'espace de recherche semble diminuer la corrélation d'évolvabilité maximale entre solutions voisines d'un RN mais reste significative.

Les variations de l'évolvabilité maximale sont plus faibles lorsque le degré de neutralité moyen est important et le paramètre d'épistasie est faible. Les valeurs de performances autour des RN ne sont donc pas réparties aléatoirement. Dans le chapitre suivant, nous allons proposer une nouvelle métaheuristique qui exploite cette information. La méthode de recherche consiste à optimiser l'évolvabilité sur chaque RN pour ensuite changer de RN. Nous examinerons dans ce chapitre les potentialités de cette nouvelle métaheuristique.

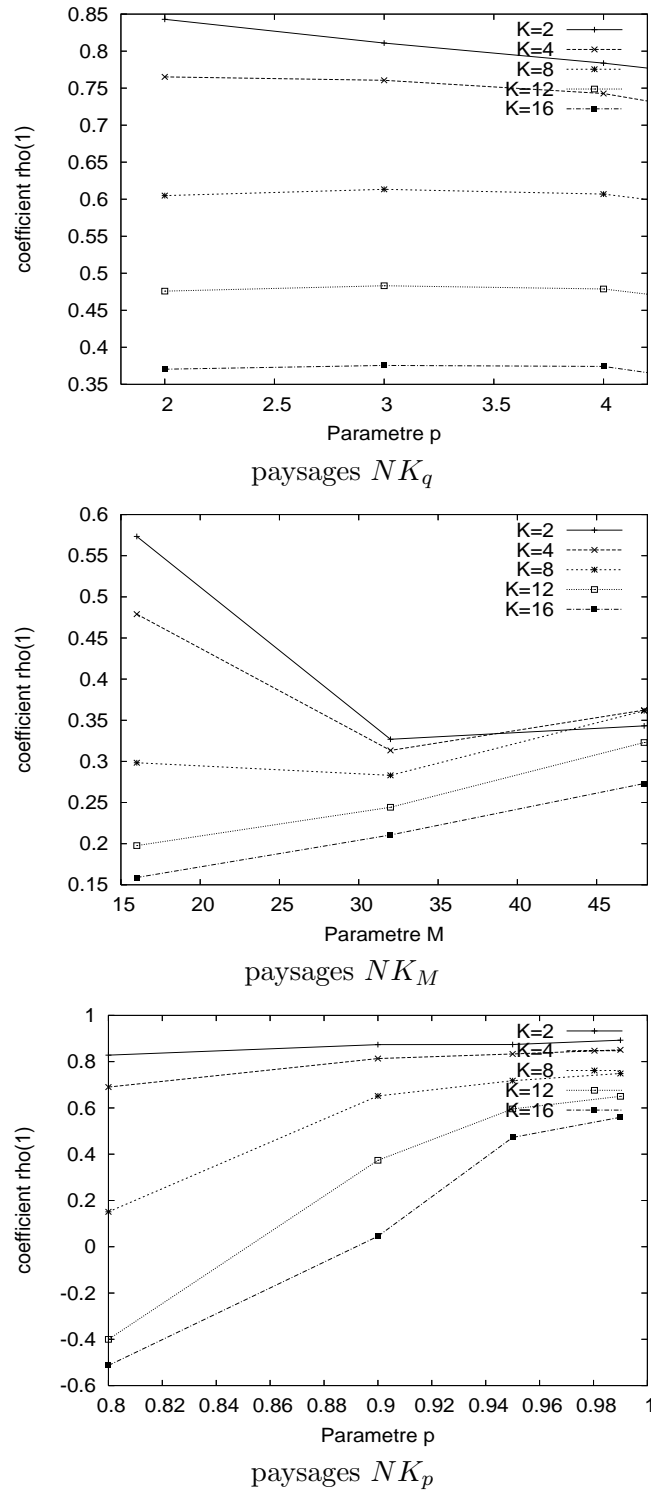


FIG. 3.24 – Coefficient d'autocorrélation d'ordre 1 de l'évolvabilité maximale pour les paysages  $NK_q$ ,  $NK_M$  et  $NK_p$  avec  $N = 64$ .

### 3.3 Autres paysages adaptatifs neutres

Dans cette section, nous étudions deux autres paysages adaptatifs neutres. L'analyse du problème classique MAX-SAT, que nous avons étudié du point de vu du nuage adaptatif dans le chapitre 2, permet mettre en évidence ces réseaux de neutralité du paysage. Le problème des *routes épistatiques* que nous avons défini dans l'article [30], présente un exemple un paysage académique de neutralité et épistasie ajustables, dont les solutions sont de taille variable.

#### 3.3.1 MAX-SAT

Le paysage MAX-SAT a été défini dans la section 1.4.3. Afin de pouvoir mener une étude exhaustive, nous allons étudier de petites instances du problème MAX-3-SAT où le nombre de littéraux par clause est égale à 3 et le nombre de variables est fixé à  $N = 16$ . Le nombre de clauses  $m$  décrit l'ensemble  $\{39, 59, 64, 69, 74, 79, 99\}$ , ainsi le rapport  $\frac{m}{N}$  varie de 2.44 à 6.19 et la valeur pour  $m = 69$  est proche de la valeur critique  $\alpha_c = 4.3$ . Pour les paysages MAX-SAT aléatoires, la probabilité de changer la valeur d'une clause est petite lorsqu'on modifie la valeur d'une variable. La probabilité de mutation neutre devrait donc être plus faible lorsque le nombre de clauses est plus petit. Pour chaque valeur des paramètres, 50 instances indépendantes de paysages aléatoires sont générées.

#### Distribution des degrés de neutralité

La figure 3.25-a montre un exemple de distribution des degrés de neutralité pour une instance du paysage lorsque  $m = 69$ . En effet, pour l'ensemble des valeurs de paramètres étudiées, les distributions des degrés de neutralité sont unimodales sans toutefois être des distributions binômiales puisque les tests du chi2 (non présentés ici) sont négatifs. La figure 3.25-b donne le degré de neutralité moyen des paysages ainsi que l'écart-type autour de cette moyenne. Comme attendu le degré de neutralité moyen décroît lorsque le nombre de clauses  $m$  augmente. La décroissance est presque linéaire de 5.82 pour  $m = 39$  à 3.84 pour  $m = 99$ . L'écart-type décroît également de 1.79 à 1.51. Il ne semble pas avoir de discontinuité dans la moyenne du degré de neutralité moyen autour de la valeur critique  $\alpha_c$ .

Les fonctions d'autocorrélation des degrés de neutralité au cours d'une marche neutre ont été réalisées suivant le même protocole que dans la section précédente 3.2.1. La figure 3.26 montre les résultats obtenus. Toutes les fonctions sont décroissantes vers la valeur 0. Il n'y pas d'alternance entre une valeur haute et une valeur basse comme rencontré pour certaines variantes paysages des paysages NK ou le degré de neutralité moyen est plus faible. La décroissance de la fonction est d'autant plus grande que le coefficient d'autocorrélation d'ordre 1 est grand. Le tableau 3.11 donne les valeurs des coefficients d'autocorrélation d'ordre 1. La valeur de  $\rho(1)$  est décroissante lorsque le nombre de clauses augmente.

En résumé, le degré de neutralité moyen et la corrélation des degrés est plus faible lorsque le nombre de clauses est grand. Selon l'échelle empirique établie dans la section 3.2.1, la corrélation des degrés de neutralité est dans la zone de forte corrélation. Le graphe des RN des paysages MAX-3-SAT pour les valeurs des paramètres étudiés n'est pas aléatoire et les variations du degrés de neutralité entre solutions voisines sur les RN sont faibles.

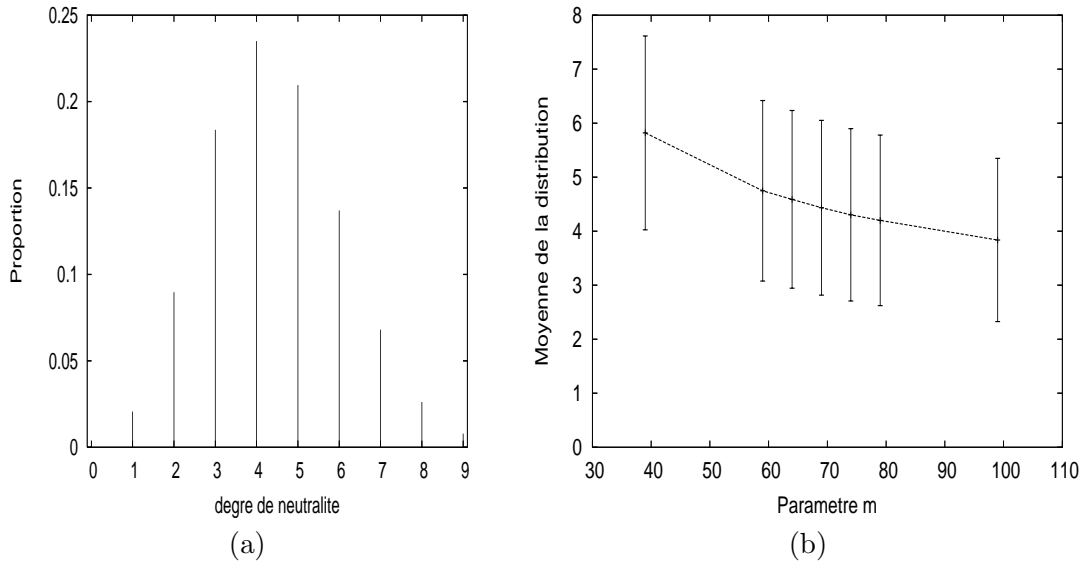


FIG. 3.25 – Distribution des degrés de neutralité pour  $m = 69$  (a) et moyenne et écart-type des distributions de degré de neutralité (b) pour les paysages MAX-3-SAT pour  $N = 16$ .

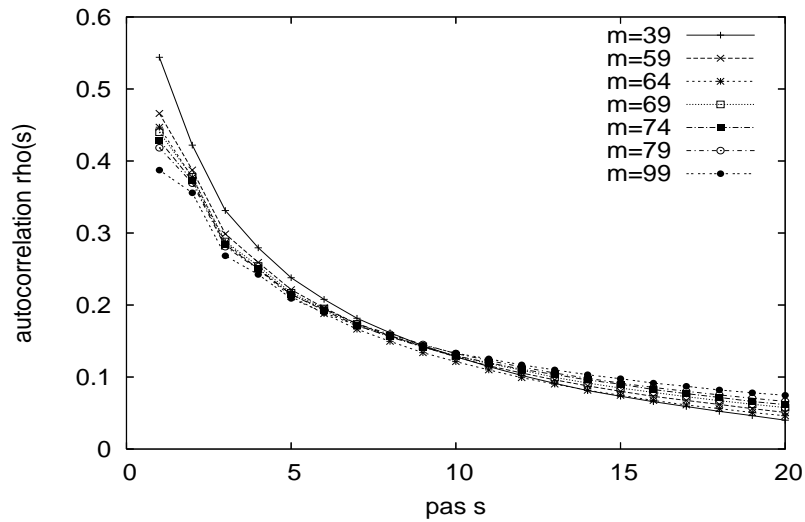


FIG. 3.26 – Fonctions d'autocorrélation des degrés de neutralité pour différentes valeurs du paramètre  $m$  pour les paysages MAX-3-SAT pour  $N = 16$ .

TAB. 3.11 – Valeur du coefficient d'autocorrélation d'ordre 1 pour les paysages MAX-3-SAT pour  $N = 16$ .

$m$	39	59	64	69	74	79	99
$\rho(1)$	0.544	0.466	0.446	0.439	0.428	0.418	0.387



TAB. 3.12 – Régressions linéaires des innovations cumulatives neutre et aléatoire  $\log(C_n(t)) = a_n \log(t) + b_n$  et  $\log(C_a(t)) = a_a \log(t) + b_a$  pour les paysages MAX-3-SAT. Le coefficient de corrélation est noté  $\rho$ .

m	Marche Neutre			Marche aléatoire			$\tau$
	$\rho$	$a_n$	$b_n$	$\rho$	$a_a$	$b_a$	
39	0.999	0.08735	1.6608	0.985	0.15195	1.7974	1.74
59	0.998	0.09233	1.8292	0.985	0.16323	1.9665	1.77
64	0.998	0.09379	1.8538	0.985	0.16623	1.9915	1.77
69	0.997	0.09134	1.8969	0.985	0.16682	2.0288	1.83
74	0.997	0.09085	1.9221	0.985	0.16844	2.0570	1.85
79	0.997	0.09188	1.9401	0.984	0.16992	2.0756	1.85
99	0.996	0.08928	2.0216	0.984	0.17477	2.1656	1.96

### Taille des RN

La figure 3.27 représente les profils rang-taille établis de la même façon que dans la section 3.2.2 pour les différentes valeurs du paramètre  $m$ .

Les profils pour toutes les valeurs du paramètre  $m$  sont du premier type décrit dans la section 3.2.2. Les paysages sont dominés par un petit nombre de très grands RN, il y a peu de RN de tailles intermédiaires enfin les nombreux petits RN ont des tailles suivant une loi puissance en fonction de leur rang. La variabilité selon l'instance du paysage est faible.

La moyenne du plus grand RN est présenté sur la figure 3.28-a normalisée par la taille de l'espace de recherche  $2^{16}$ . Les tailles sont au-dessus de 10%, les plus grands RN dominent l'ensemble des paysages. La taille du plus grand RN est décroissante lorsque le nombre de clauses augmente *i.e.* le degré de neutralité moyen diminue. La faible variation de l'écart-type normalisé par la moyenne (figure 3.28-b) montre que les variabilités des profils sont presque identiques.

En résumé, le nombre de clauses  $m$  a peu d'influence sur les profils rang-taille. Ceux-ci sont du premier type où les grands RN dominent le plus le paysage et où la variabilité est faible selon l'instance. Ces profils sont à rapprocher des profils des paysages  $NK_q$  lorsque  $K = 3$  et  $q = 2$  ou des paysages  $NK_M$  lorsque  $K = 3$  et  $M = 32$ .

### Taux d'innovation

Les innovations cumulatives neutre ( $C_n$ ) et aléatoire ( $C_a$ ) ont été calculées suivant le même protocole que dans la section 3.2.4. De même que pour les variantes des paysages NK, nous trouvons que les courbes sont les courbes représentatives d'une fonction puissance. Cela est confirmé par la régression des courbes en échelle logarithmique. La table 3.12 donne les résultats de ces régressions et la valeur du rapport  $\tau = \frac{C_a}{C_n}$ .

Les coefficients de corrélation sont supérieurs à 0.98, statistiquement, les innovations cumulatives sont des fonction puissances. Les valeurs de  $b_n$  et  $b_a$  sont croissantes avec le nombre de clauses  $m$ . Le nombre de valeurs de performance accessibles dans le voisinage d'une solution augmente avec la diminution de la neutralité moyenne du paysage. Le coefficient  $a_a$  augmente avec  $m$ , la découverte de nouvelles valeurs de performance au cours d'une marche aléatoire augmente lorsque la neutralité moyenne du paysage diminue. La variation du coefficient  $a_n$  n'est pas monotone.  $a_n$  croît entre  $m = 39$  et  $m = 64$ , et pour  $m > 64$ , la variation n'est plus monotone. Toutefois, le rapport  $\tau$  est croissant avec le nombre de clauses. Sa valeur maximale est alors de 1.96.

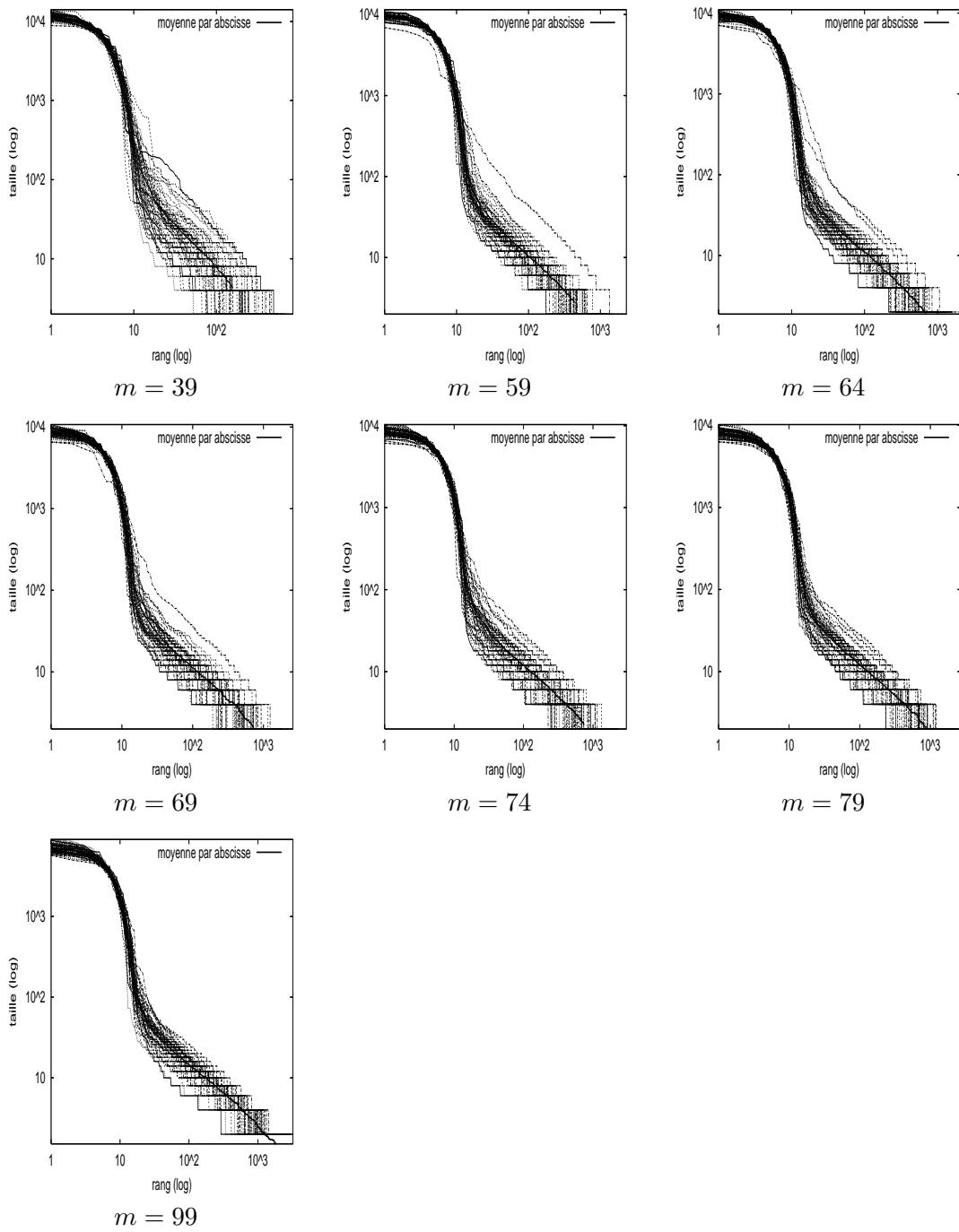


FIG. 3.27 – Profils rang-taille des RN pour les paysages MAX-3-SAT pour  $N = 16$  et différentes valeur de  $m$ .

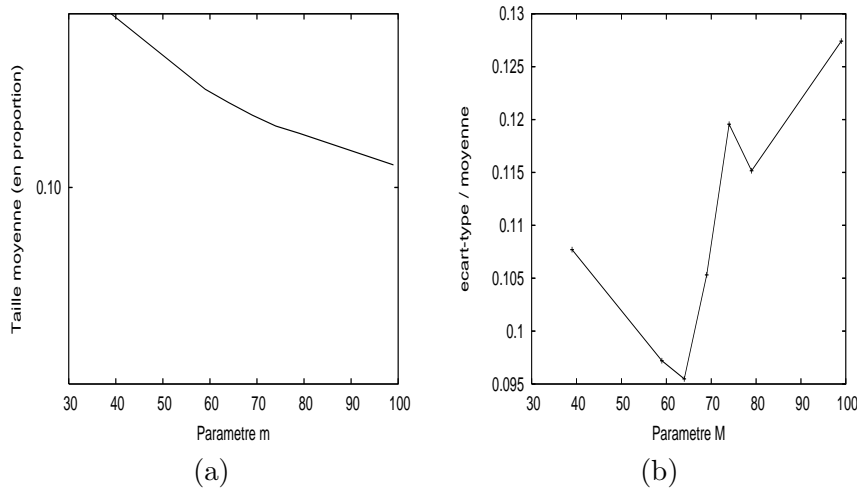


FIG. 3.28 – Moyenne (a) et écart-type (b) du plus grand RN pour les paysages MAX-3-SAT pour  $N = 16$ .

TAB. 3.13 – Valeur des coefficient d'autocorrélation d'ordre 1 de l'évolvabilité maximale pour les paysages MAX-3-SAT pour  $N = 16$ .

$m$	39	59	64	69	74	79	99
$\rho(1)$	0.627	0.605	0.604	0.602	0.609	0.602	0.598

En résumé, les innovations cumulatives sont des fonctions puissances permettant de calculer le nombre de valeurs de performance nouvelles rencontrées au cours de marches neutres et aléatoires. Relativement à l'ensemble des solutions de l'espace de recherche, les RN possèdent une capacité de découverte importante et qui augmente avec le nombre de clauses.

### Autocorrélation de l'évolvabilité maximale

L'autocorrélation de l'évolvabilité maximale a été mesurée suivant le même protocole que dans la section 3.2.5. La figure 3.26 présente les fonctions d'autocorrélation et la table 3.11 donne les valeurs des coefficients d'ordre 1 pour les différentes valeurs du paramètre  $m$ .

Ces fonctions sont décroissantes de limite nulle. Elles sont du premier décrit dans la section 3.2.5. Pour toutes les valeurs du paramètre  $m$ , les coefficients  $\rho(1)$  ont des valeurs très proches autour de 0.6. Les fonctions d'autocorrélation se différencient par leur vitesse de décroissance. Les fonctions décroissent d'autant plus vite que le nombre de clauses est petit.

Les corrélations d'évolvabilité maximale entre solutions voisines dans un réseau de neutralité ne sont pas nulles. Les valeurs de performances autour des RN ne sont donc pas réparties aléatoirement. Les variations de l'évolvabilité maximale sont plus petites lorsque le nombre de clauses est moins important.

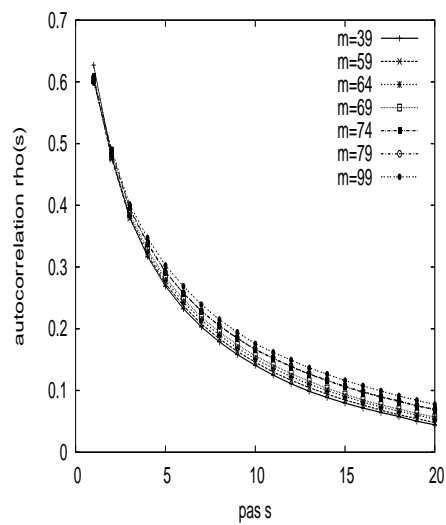


FIG. 3.29 – Fonctions d’autocorrélation de l’évolvabilité maximale pour différentes valeurs du paramètre  $m$  pour les paysages MAX-3-SAT pour  $N = 16$ .

### 3.3.2 Routes épistatiques

Dans cette sous-section, nous étudions la neutralité du paysage des Routes Épistatiques que nous avons proposé dans [30], où les solutions sont de taille variable. Ce paysage ajoute au paysage des routes royales [101] des interactions épistatiques entre les solutions sous-optimales appelées blocs. Nous présentons tout d'abord, les paysages des Routes Royales (RR) issue du domaine de la programmation génétique linéaire afin de permettre l'étude des opérateurs de recombinaison ; en particulier, nous mettons en évidence leurs effets destructifs (ou constructifs) sur les blocs. Le nombre entier strictement positif  $b$  est un paramètre du paysage indiquant la taille d'un bloc. L'espace de recherche  $\mathcal{S}$  est constitué des solutions de taille variable sur le vocabulaire  $\Sigma = \{a_1, \dots, a_N\}$  fini de taille  $N$ . Le voisinage d'une solution de taille  $\lambda$  est l'ensemble des solutions dont la taille est  $\lambda - 1$  (délétion d'une lettre),  $\lambda$  (altération d'une lettre) et  $\lambda + 1$  (ajout d'une lettre). La fonction  $B_b$  indique la présence d'un bloc contigu de taille  $b$  d'une même lettre  $a \in \Sigma$  dans une chaîne de longueur  $\lambda$ .  $B_b(s, a) = 1$  si  $\exists i \in [0, \lambda - b[ \forall j \in [0, b - 1] s_{i+j} = a$  et  $B_b(s, a) = 0$  sinon. La performance d'une chaîne  $s$  est alors le nombre de blocs restant à construire parmi les  $N$  possibles :

$$f_{Nb}(s) = N - \frac{1}{N} \sum_{i=1}^N B_b(s, a_i)$$

Le paysage des Routes Épistatiques (RE) est donc caractérisé par les trois paramètres  $N$ ,  $K$  et  $b$ . Ce paysage utilise la présence ou de l'absence d'un bloc comme dans le paysage des Routes Royales et ajoute de l'épistasie entre ces blocs à l'aide d'une fonction d'adaptation d'un paysage  $NK$  :

$$f_{NKb}(s) = \frac{1}{N} \sum_{i=1}^N f_i(B_b(s, a_i), B_b(s, a_{i_1}), \dots, B_b(s, a_{i_K}))$$

Les fonctions  $f_i$  sont les fonctions de contributions du paysage NK (définition 1.4.3). Une permutation de l'espace du paysage NK est effectuée de manière à obtenir l'optimum globale pour  $1^N$ , où tous les blocs sont présents.

Le neutralité moyenne est d'autant plus grande que la taille des blocs  $b$  est grande. Le paramètre  $K$  ajuste l'épistasie, son influence sur l'épistasie est d'autant plus faible que la neutralité moyenne est importante. Lorsque  $K = 0$ , le paysage RE est sans épistasie correspond au paysage RR, lorsque  $K = N - 1$ , l'épistasie est maximale, la construction successive des blocs rencontre beaucoup d'optima locaux. La thèse [100] p. 84 propose une étude des paysages RR et RE du point de vue de l'épistasie.

Dans la suite, nous menons une étude de la neutralité sur ce type de paysage. Le nombre de bloc  $N$  décrit l'ensemble  $\{8, 10, 16\}$ , le paramètre d'épistasie  $K$  l'ensemble  $\{0, 2, 4, 7\}$  et la taille des blocs  $b$  l'ensemble  $\{1, 2, 3, 4\}$ . Pour chaque valeurs des paramètres, nous générons de manière indépendante 10 instances de paysages NK. Il n'est évidemment pas possible de générer exhaustivement toutes les solutions de l'espace de recherche. Pour chaque instance de paysage, nous générons un échantillon de solutions initiales. Les solutions initiales sont de taille choisie uniformément entre  $N b$  et  $3N b$  et chaque locus est choisi uniformément dans  $\Sigma$ . De même, le voisinage de chaque solution est trop large pour être entièrement explorer. Pour chaque solution nous explorons un échantillon de 300 solutions voisines dont la taille est choisie de manière équiprobable parmi  $\{\lambda - 1, \lambda, \lambda + 1\}$ .

#### Distribution des degrés de neutralité

L'échantillon de solutions pour estimer la distribution des degrés de neutralité est de taille  $10^3$ . Nous ne pouvons pas ici calculer le degré de neutralité (nombre de solutions voisines de

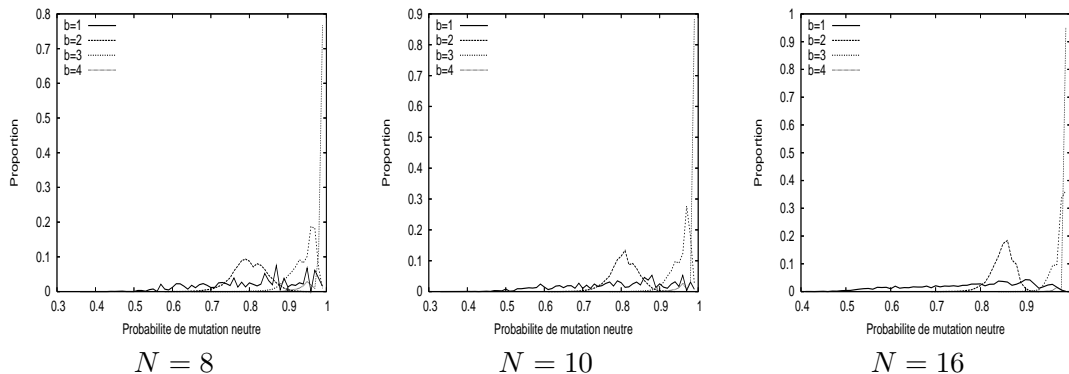


FIG. 3.30 – Distribution des degrés de neutralité pour différentes valeurs des paramètres des paysages RE.

TAB. 3.14 – Espérance ( $E$ ) et écart-type ( $\sigma$ ) des distributions du degré de neutralité pour différentes valeurs des paramètres des paysages RE.

$b$	$N = 8$		$N = 10$		$N = 16$	
	$E$	$\sigma$	$E$	$\sigma$	$E$	$\sigma$
1	0.805	0.122	0.788	0.130	0.785	0.126
2	0.804	0.044	0.811	0.036	0.854	0.025
3	0.946	0.031	0.960	0.025	0.982	0.014
4	0.990	0.016	0.994	0.011	0.998	0.004

même performance) puisque la taille du voisinage est de taille variable. Nous calculerons pour une solution, la probabilité qu’une solution voisine soit de même performance. La figure 3.30 présente les distributions de ces probabilités.

Ces distributions de probabilité ne dépendent pas du paramètre  $K$ . Nous avons donc donné uniquement les résultats pour  $K = 0$ . En effet, la performance d’une solution est modifiée si et seulement le nombre de blocs est modifié. La valeur du paramètre  $K$  ajuste seulement la corrélation entre deux solutions voisines lorsque le nombre de blocs présents dans chacune d’entre elles est différent. Les distributions sont des distributions unimodales, exceptées quand la valeur de  $b$  égale à 1.

La table 3.14 donne les moyennes et les écart-types des distributions précédentes. La neutralité moyenne des paysages augmente fortement avec le paramètre  $b$  et diminue dans une moindre mesure lorsque le paramètre  $N$  augmente. La probabilité qu’une solution voisine ait la même performance est très élevée (supérieure à 0.78). Pour  $b = 4$ , les solutions voisines (probabilité supérieure à 0.99) ont quasiment toutes la même performance.

L’autocorrélation des degrés de neutralité (figure 3.31) a été calculée à partir de 50 marches neutres de longueur 150 pour chaque instance de paysage. De nouveau, les fonctions sont indépendantes du paramètre  $K$ . Elles sont décroissantes jusqu’à la valeur nulle. La décroissance est d’autant plus accentuée que le coefficient d’ordre 1,  $\rho(1)$ , est grand. Pour une valeur du paramètre  $b$ , le coefficient  $\rho(1)$  décroît lorsque  $N$  augmente. Remarquons que, contrairement aux variantes des paysages NK ou aux paysages MAX-SAT, la corrélation diminue lorsque le degré de neutralité moyen augmente.

En résumé, le degré de neutralité moyen est indépendant du paramètre épistatique  $K$  et augmente fortement avec la taille d’un bloc  $b$  (jusqu’à 99% de solutions voisines de même

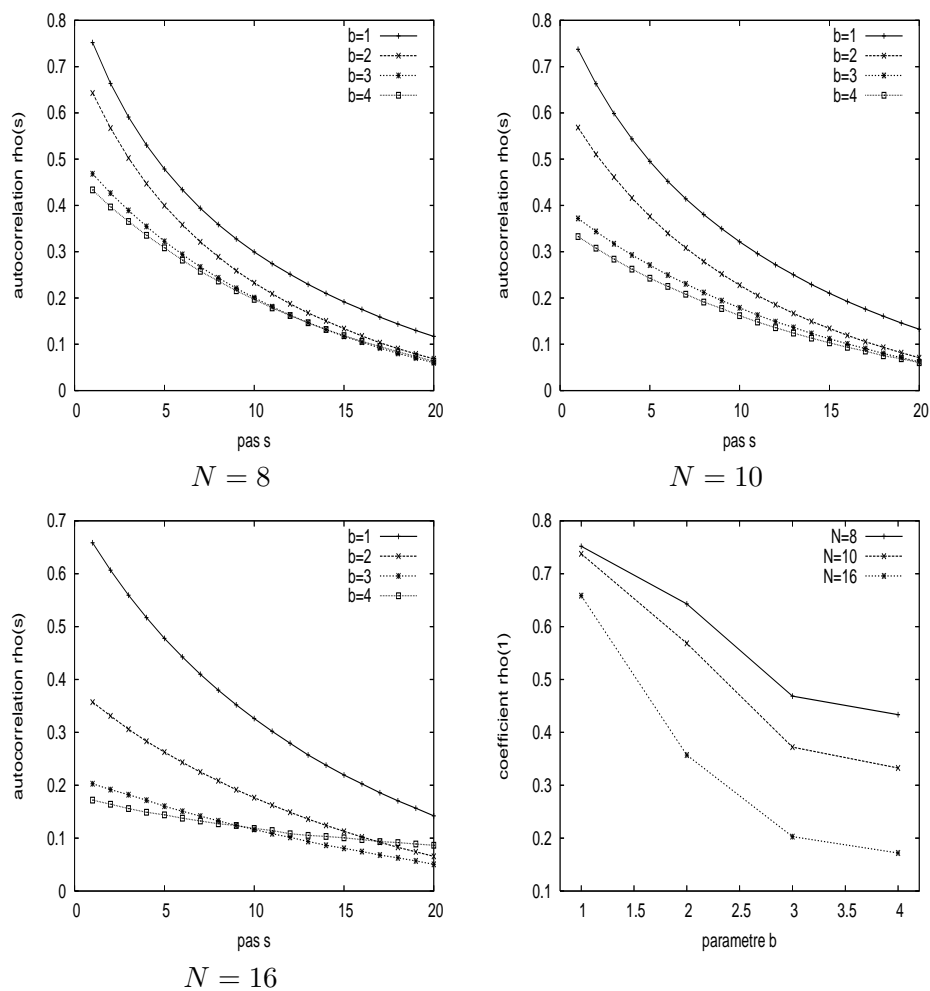


FIG. 3.31 – Fonctions d'autocorrélation et coefficient d'autocorrélation d'ordre 1 (en bas à droite) des degrés de neutralité pour différentes valeurs des paramètres des paysages RE.

TAB. 3.15 – Régressions linéaires des innovations cumulatives neutre et aléatoire  $\log(C_n(t)) = a_n \log(t) + b_n$  et  $\log(C_a(t)) = a_a \log(t) + b_a$  pour les paysages RE. Le coefficient de corrélation est noté  $\rho$ .

N	K	b	Marche Neutre			Marche aléatoire			$\tau$
			$\rho$	$a_n$	$b_n$	$\rho$	$a_a$	$b_a$	
8	0	1	0.986	0.17458	1.9042	0.994	0.64191	1.2803	3.68
		2	0.989	0.17522	2.0247	0.996	0.57601	1.7496	3.29
		3	0.996	0.13046	1.6188	0.990	0.39908	1.2194	3.06
		4	0.991	0.28299	0.4311	0.986	0.32750	0.3557	1.16

performance pour  $b = 4$ ). Le nombre de blocs maximal  $N$  a peu d'influence. La corrélation des degrés est faible lorsque la taille des blocs est grande. Selon l'échelle empirique établie dans la section 3.2.1, la corrélation des degrés de neutralité est dans la zone de forte corrélation excepté lorsque le degré de neutralité moyen est important. Le graphe des RN des paysages RE pour les valeurs de paramètres étudiées n'est pas aléatoire et les variations du degré de neutralité entre solutions voisines sur les RN sont faibles.

### Taux d'innovation

Comme pour les distributions de probabilité précédentes, les innovations cumulatives ne dépendent pas du paramètre  $K$ . Nous avons donc donné seulement les résultats pour  $K = 0$ .

Lorsque le degré de neutralité moyen du paysage augmente, les coefficients  $b_n$ ,  $b_a$  ainsi que  $a_a$  diminuent. La variation de  $a_n$  n'est pas monotone en fonction de la taille de bloc  $b$ . Toutefois, le coefficient  $\tau$  croît avec la taille de bloc. Par contre, ce nombre de blocs  $N$  influence peu les valeurs des coefficients.

En résumé, nous avons observé la corrélation inverse comparée à celles des paysages  $NK_q$ ,  $NK_M$ ,  $NK_p$  et MAX-SAT, à savoir que le taux d'innovation augmente avec la neutralité.

### Autocorrélation de l'évolvabilité maximale

L'autocorrélation de l'évolvabilité maximale a été calculée à partir des mêmes marches neutres que pour l'autocorrélation des degrés de neutralité. La figure 3.33 montre un exemple parmi toutes les fonctions d'autocorrélation obtenues. Pour l'ensemble des paramètres testés, les formes des courbes représentatives sont identiques à l'exemple donné, seule la valeur du coefficient d'autocorrélation d'ordre 1 varie : les fonctions sont décroissantes de limite nulle.

Les coefficients d'autocorrélation d'ordre 1 sont donnés figure 3.33. Pour des valeurs de  $K$  et  $b$  fixées, le coefficient est décroissant lorsque  $N$  augmente. Quelque soient  $N$  et  $K$ , la valeur  $\rho(1)$  pour  $b = 1$  (au dessus de 0.5) domine celles pour les autres valeurs de  $b$ . Le coefficient  $\rho(1)$  est faible pour  $b > 1$ .

En résumé, l'autocorrélation de l'évolvabilité maximale est plus faible (hormis le cas limite où la taille du bloc est 1) que dans le cas des paysages  $NK_q$ ,  $NK_M$ ,  $NK_p$  ou MAX-SAT.



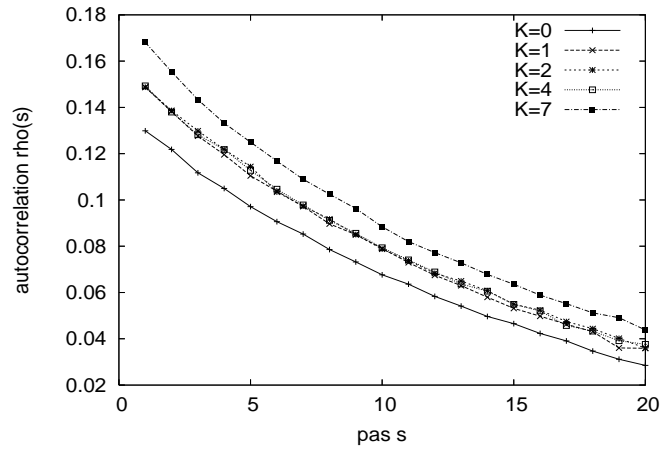


FIG. 3.32 – Fonctions d'autocorrélation de l'évolvabilité maximale pour différentes valeurs du paramètre  $K$  sur les paysages RE avec  $N = 16$  et  $b = 2$ .

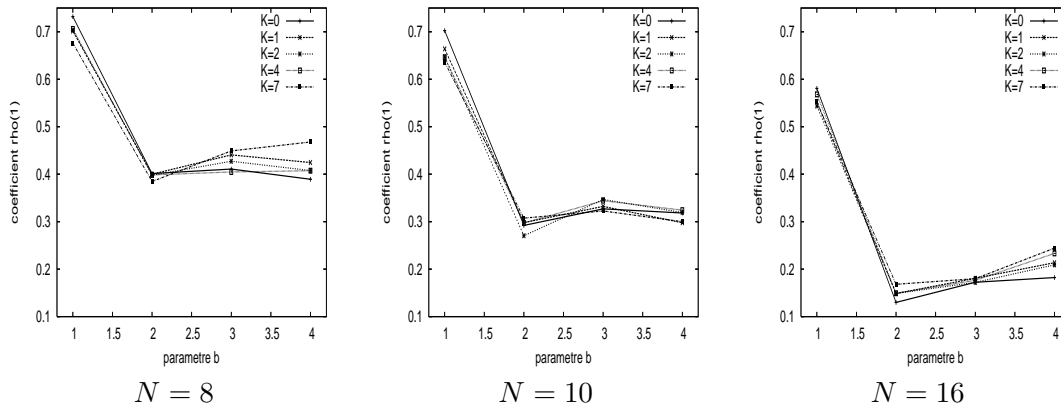


FIG. 3.33 – Coefficient d'autocorrélation de l'évolvabilité maximale d'ordre 1 pour différentes valeurs des paramètres des paysages RE.

### 3.4 Synthèse du chapitre

Dans ce chapitre, nous avons présenté les mesures existantes qui permettent de caractériser les réseaux de neutralité (RN) d'un paysage adaptatif : distribution de degré de neutralité, autocorrélation des degrés de neutralité, taille des RN, taux d'innovation. Nous avons défini une nouvelle mesure, l'*autocorrélation de l'évolvabilité*, qui mesure la corrélation de l'évolvabilité au cours d'une marche sur un RN. Cette mesure permet de compléter la description des paysages neutres à proximité des RN.

Nous avons analysé à l'aide des mesures trois familles de paysages pour lesquelles la neutralité est ajustable. Ces trois variantes des paysages NK, les paysages  $NK_q$ ,  $NK_M$  et  $NK_p$ , sont représentatives de la manière d'obtenir de la neutralité dans un paysage additif :

- pour les paysages  $NK_q$ , les termes de la somme sont discrétisés entre 0 et  $q - 1$ . Le degré de neutralité moyen est alors limité mais les RN sont structurés.
- pour les paysages  $NK_p$ , les termes de la somme sont nulles avec une probabilité  $p$ . Le degré de neutralité moyen est alors fortement contrôlé par le paramètre  $p$ , et les RN sont structurés.
- pour les paysages  $NK_M$ , la somme est discrétisée en  $M$  valeurs. Le degré moyen du paysage dépend fortement de  $M$ , mais les RN sont moins structurés que pour les autres variantes.

Nous avons analysé les RN du problème académique MAX-SAT et du problème des Routes Épistatiques que nous avons proposé dans [30] dont les solutions sont de tailles variables.

Dans ces études, nous avons proposé une représentation originale de la distribution des tailles à l'aide d'un profil rang-taille et calculé la corrélation des degrés de neutralité qui n'avait jamais été auparavant utilisée dans le domaine de l'optimisation combinatoire. Nous avons pu proposer une échelle permettant de classer les problèmes selon la valeur de cette corrélation.

L'accumulation de ces données peut maintenant servir de référence dans l'étude de nouveaux paysages adaptatifs neutres. La mise en évidence de nouvelles caractéristiques dans des paysages de référence, autocorrélation des degrés, autocorrélation de l'évolvabilité maximale, permet à la fois de compléter l'image d'un paysage adaptatif neutre et d'imaginer de nouvelles métaheuristiques d'optimisation.



## Chapitre 4

# Dynamique et Métaheuristiques dans les problèmes neutres

Dans ce chapitre, nous allons d’abord présenter la dynamique des algorithmes évolutionnaires (AE) dans les paysages adaptatifs neutres. Cette dynamique des *équilibres ponctuels* fût initialement découverte en biologie de l’évolution moléculaire, puis adaptée au domaine de l’optimisation combinatoire. Nous généraliserons au mode de sélection par tournoi, un résultat énonçant la dynamique des AE sur un RN.

Les mesures sur les paysages adaptatifs neutres ont mis en lumière de nouvelles caractéristiques des réseaux de neutralité : les RN sont structurés et leurs interactions peuvent être utilisées pour guider une recherche. Nous proposerons ici une nouvelle métaheuristique, appelé *recherche périscopique* (RP), adaptée aux paysages neutres utilisant la notion d’évolvabilité pour guider la recherche sur les RN. Les résultats expérimentaux sur les variantes neutres des paysages NK confirmeront l’intérêt de la RP et surtout la corrélation entre ses performances et les mesures de neutralité réalisées au chapitre 3.

Le problème de majorité est un problème d’apprentissage difficile de la tâche de classification réalisée par un automate cellulaire. Le paysage adaptatif relatif à ce problème est massivement neutre : cela est du en partie à l’erreur d’évaluation de la performance d’une règle. Nous allons étudier le paysage dans son ensemble et montrer que le nombre important de solutions de performance nulle ne permet pas une étude statistique. Ensuite, nous allons étudier ce paysage par le “haut” en considérant les meilleurs optima locaux connus à ce jour. Nous définirons le sous-espace Olympe des similarités entre ces différents optima et des symétries du problème. L’étude de ce sous-espace et l’analyse de ses RN permettra de montrer qu’une optimisation par l’AE restreinte à l’Olympe prenant en compte la neutralité est plus facile et permet de trouver à moindre coût des solutions de qualité équivalente aux meilleurs.

### 4.1 Dynamique des algorithmes évolutionnaires

Dans cette section, nous allons commencer par décrire l’évolution par équilibres ponctuels, puis nous exposerons la dynamique d’évolution des algorithmes évolutionnaires sur un réseau de neutralité.

En paléontologie, les travaux de Eldge et Gould [34] ont mis en évidence un type d’évolution appelé *équilibres ponctuels*. L’évolution se déroule selon l’alternance de deux phases. Pendant la première, l’adaptation d’une espèce n’augmente pas et l’adaptation moyenne de la population stagne. Dans une seconde phase, l’adaptation augmente brusquement : on observe alors un saut qualitatif significatif ; la descendance du nouvel individu trouvé se répand rapidement dans la population (voir figure 4.1). Ce schéma d’évolution d’une espèce dans un paysage multimodale a

été explicitement proposé par S. Wright [147] et modélisé par C. M. Newmann [90] et R. Lande [78]. La population forme un nuage localisé autour d'un optimum local : elle reste dans cet état jusqu'à ce qu'un mutant puisse traverser une "vallée" afin d'atteindre un nouvel optimum. Ce type d'évolution peut également trouver une explication dans le contexte des paysages neutres.

Dans le domaine de l'optimisation par algorithme évolutionnaire, on rencontre les mêmes phases d'évolution qui sont dues à la neutralité du paysage. Pendant la première phase la population se diffuse aléatoirement sur le réseau avec une probabilité faible et constante de trouver une porte ; puis dans la seconde phase, une porte conduisant à un réseau de neutralité de meilleure performance est empruntée. La question de savoir si un algorithme évolutionnaire effectue une recherche aléatoire pendant la première phase est toujours ouverte. Quelques éléments de réponse ont été apportés par l'étude de la dynamique sur un réseau de neutralité.

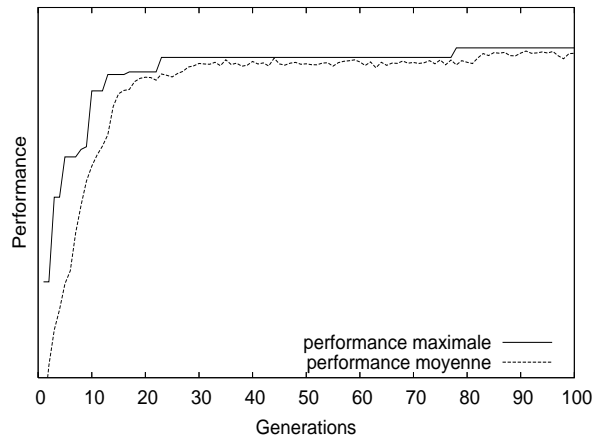


FIG. 4.1 – Exemple d'évolution de performance d'une population lors d'une dynamique d'équilibres ponctués.

Derrida [31] a réalisé une étude détaillée de l'évolution d'une population de taille finie sur un paysage dit "plat", où toutes les solutions ont la même performance. Dans ce cas, et contrairement à l'intuition, la dynamique évolutionnaire demeure très complexe. La population se divise en sous-populations où les solutions partagent une même généalogie. Des travaux [13] [94] [92], à la croisée de la biologie moléculaire et de l'optimisation, ont étudiés la convergence d'une population sur un réseau de neutralité. Le modèle prend en compte une population infinie soumise à une sélection proportionnelle à la performance et à une mutation d'un bit par locus. Wilke [142] généralise cette étude à un mode de mutation plus général. Dans tous les cas, la distribution limite de la population sur le réseau de neutralité est uniquement déterminée par la topologie du réseau. Elle est indépendante de la proportion de solutions sur le RN ou du taux de mutation. Le degré de neutralité moyen de la population est supérieur au degré de neutralité moyen du RN, *i.e.* la distribution de la population converge vers les régions découvertes où les solutions ont les plus grands degrés de neutralité. En formulant l'hypothèse que la probabilité de mutation augmentant la performance est faible pour l'ensemble des solutions du RN, la probabilité de dégrader la performance des solutions par mutation, *i.e.* la probabilité de diminution de performance d'une solution, est plus faible pour la population limite que pour une population répartie aléatoirement sur le RN. L'évolution de la population sur un RN augmente, sous cette hypothèse, la robustesse (la non dégradation de performance par mutation) vis-à-vis des mutations [114].

Dans chacun des cas précédents, seule une sélection proportionnelle à la performance est

considérée. Cependant, il ne faut pas oublier que la définition du voisinage neutre dépend du mode de sélection. Par exemple, avec une sélection proportionnelle à la performance, deux valeurs de performance très proches auront quasiment le même taux de reproduction alors qu'avec une sélection par tournoi, elles seront considérées comme strictement différentes. Le choix du mode de sélection influence la nature de la neutralité et ainsi la dynamique de la recherche. Par ailleurs la sélection par tournoi étant la plus largement utilisée pour les algorithmes évolutionnaires, nous proposons donc de montrer, en adaptant la preuve de Nimwegen [94], que les résultats précédents sont également valides pour ce type de sélection :

**Proposition :**

Pour une sélection par tournoi, la distribution limite de la population sur un RN est uniquement déterminée par la topologie du réseau. Elle est indépendante de la proportion de solutions sur le RN ou du taux de mutation. Le degré de neutralité moyen de la population est supérieur au degré de neutralité moyen du RN.

*Preuve :*

Nous considérons un algorithme évolutionnaire limité à l'itération de sélection par tournoi puis de mutation. L'espace de recherche considéré est l'ensemble des chaînes binaires de longueur  $L$ . La mutation altère un seul bit par chaîne avec une probabilité<sup>18</sup>  $\mu$ . Soit  $P$  la proportion de la population (infinie) sur le réseau de neutralité  $G$  de performance  $\sigma$  ; on suppose que le reste de la population a des performances inférieures. La sélection par tournoi de taille  $t$  choisit au hasard  $t$  solutions dans la population et sélectionne la meilleure comme géniteur. La proportion moyenne sur le réseau après le tournoi est de  $1 - (1 - P)^t$ , soit  $P\alpha$  avec  $\alpha = \sum_{k=0}^{t-1} \binom{t}{k+1} (-P)^k$ . Après mutation, une proportion  $\langle \nu \rangle$  reste sur le réseau et une proportion  $1 - \langle \nu \rangle$  voit sa performance se dégrader. Soit  $Q$  la proportion des solutions de performances inférieures qui atteignent le réseau  $G$  par mutation, nous avons à l'équilibre  $P = \alpha \langle \nu \rangle P + Q$ . En général,  $Q$  est négligeable devant  $P$  ; si bien que nous obtenons un équilibre entre les individus quittant le réseau par mutation et la pression de sélection :  $\alpha \langle \nu \rangle = 1$ . Maintenant exprimons la relation entre le degré de neutralité  $d_s$  d'une solution  $s \in G$  et la probabilité  $\nu_s$  de rester sur  $s$  :  $\nu_s = 1 - \mu(1 - \frac{d_s}{L})$ . Asymptotiquement  $\langle \nu \rangle = \sum_{s \in G} \nu_s \frac{P_s}{P}$  et le degré moyen de neutralité  $\langle d \rangle$  de la population est égale à  $\sum_{s \in G} d_s \frac{P_s}{P}$ . Pour une population sur  $G$ , nous obtenons la relation entre la neutralité apparente de la population et la proportion d'individus sur le réseau :

$$\langle d \rangle = L(1 - \frac{\alpha - 1}{\mu\alpha}) \quad (4.1)$$

Ainsi connaissant la proportion de solutions de même performance dans la population, on peut estimer le degré de neutralité moyen de la population. En considérant que pour des solutions de même performance, le tournoi sélectionne un individu avec une probabilité uniforme, la proportion  $P_s$  de solutions  $s \in G$  est le produit de la proportion de solutions sur  $G$  et de la probabilité que cette solution soit  $s$ , soit  $P_s = \frac{P_s}{P}(1 - (1 - P)^t) = \alpha P_s$ . La distribution asymptotique vérifie donc  $\forall s \in G, P_s = (1 - \mu)\alpha P_s + \frac{\mu}{L} \sum_{t \in \mathcal{V}(s) \cap G} \alpha P_t$ . Ainsi en utilisant l'équation 4.1, nous obtenons

$$\langle d \rangle \vec{P} = \mathbf{G} \vec{P} \quad (4.2)$$

où  $\mathbf{G}$  est la matrice d'adjacence du graphe induit par  $G$  et  $\vec{P}$  le vecteur de distribution  $P_s$  pour tout  $s \in G$ . À l'aide du théorème de Perron-Frobenius, nous pouvons conclure comme dans [94] que la distribution asymptotique  $\vec{P}$  est indépendante du taux de mutation et du niveau d'adaptation et ne dépend que de la topologie du réseau. De même, le degré moyen de la population est plus grand que la moyenne sur le réseau, ce qui signifie que la population converge vers les zones de plus grand degré de neutralité.

---

<sup>18</sup>La preuve se généralise aux chaînes sur un alphabet de taille quelconque ainsi qu'à d'autres modes de mutation

Cette preuve repose sur les hypothèses d'équilibre de proportion d'une part entre le réseau et les autres niveaux d'adaptation et d'autre part en tout point du réseau de neutralité. Notons que la vitesse et le mode de convergence ne sont pas exposés, ni les variations stochastiques autour de la distribution asymptotique qui peuvent être déterminante pour l'efficacité de la recherche.

## 4.2 Métaheuristique dans les paysages neutres : Recherche Périscopique

Dans cette section, nous proposons une nouvelle métaheuristique, que nous avons défini dans les articles [21, 138], adaptée aux paysages neutres utilisant la notion d'évolvabilité pour guider la recherche sur les réseaux de neutralité. Cette métaheuristique, appelé *recherche périscopique* (RP), est imagée par la métaphore de la nage avec périscope. Elle consiste en l'itération de deux étapes, la première optimise une mesure d'évolvabilité sur un réseau de neutralité, la suivante réalise un saut qualitatif de performance en sélectionnant une solution voisine adéquate.

Après avoir défini l'algorithme de la recherche, nous comparons les potentialités de cette métaheuristique à celles existantes sur les variantes des paysages NK étudiées au chapitre 3

### 4.2.1 Algorithme

Dans le chapitre 3, nous avons mis en évidence, pour un certain nombre de paysages, que la corrélation de l'évolvabilité maximale entre solutions voisines sur un réseaux de neutralité est non nulle. Nous pouvons en déduire que le sous-paysage réduit aux solutions du RN, de voisinage induit par la relation de voisinage neutre et dont la fonction à optimiser est l'évolvabilité maximale est peu rugueux. On peut donc espérer optimiser l'évolvabilité maximale sur un RN par recherche locale afin d'obtenir une solution potentiellement de meilleure performance. Le principe de la métaheuristique de recherche périscopique utilise ce principe. La RP itère deux phases : durant la première, elle optimise sur un RN, à l'aide d'un opérateur local, une mesure d'évolvabilité jusqu'à ce qu'une condition d'arrêt termine cette phase ; la seconde étape consiste à appliquer un autre opérateur local afin d'obtenir une solution de meilleure performance.

Dans un premier temps, l'algorithme de Recherche Périscopique Générique (RPG) est défini sans particulariser ni la définition de l'évolvabilité sur le RN, ni les opérateurs de recherche locale entre RN. Soient  $op_{f,\mathcal{V}}$  un opérateur de recherche local et une mesure d'évolvabilité  $evol : \mathcal{S} \rightarrow \mathbb{R}$  (définition 1.3.2) relative à l'opérateur  $op_{f,\mathcal{V}}$ . Soit  $op_{evol,\mathcal{V}_n}$  un autre opérateur de recherche local qui sélectionne en fonction de l'évolvabilité  $evol$  une solution voisine dans le voisinage neutre. L'algorithme 3 donne l'algorithme de la RPG.  $Cond$  et  $Cond_n$  sont des conditions d'arrêt de l'optimisation par l'opérateur  $op_{f,\mathcal{V}}$  et de l'optimisation de l'évolvabilité par l'opérateur  $op_{evol,\mathcal{V}_n}$ .

---

#### Algorithme 3 Recherche Périscopique Générique

---

Choisir une initiale solution  $s \in \mathcal{S}$

**répéter**

**tant que** not  $Cond_n(s)$  **faire**

$s \leftarrow op_{evol,\mathcal{V}_n}(s)$

**fin tant que**

$s \leftarrow op_{f,\mathcal{V}}(s)$

**jusqu'à**  $Cond_n(s)$

---

La métaphore d'une recherche à la surface d'un lac est souvent utilisée pour "visualiser" la

recherche sur les réseaux de neutralité. On parle alors de “dérive”, de “nageur de réseaux”, etc. La recherche périscopique complète la métaphore du nageur sur un RN en équipant le nageur d’un périscope qui lui permet de se guider en “regardant” au-dessus du RN. Le périscope est la représentation métaphorique de la mesure d’évolvabilité.

Nous exposons maintenant la métaheuristique de recherche périscopique dans le cas où l’opérateur de recherche local est l’opérateur HC (voir section 2.3.1) et la mesure d’évolvabilité est l’évolvabilité maximale (voir section 3.2.5).

**Définition:** L’évolvabilité maximale d’une solution  $s$  est la fonction  $evol_{max}$  qui associe à tout  $s \in \mathcal{S}$  la plus grande performance du voisinage  $\mathcal{V}(s) : \forall s \in \mathcal{S}, evol_{max}(s) = \max\{f(s') \mid s' \in \mathcal{V}(s)\}$ .

La définition du prédicat *isLocal* (voir section 1.3.1) permet de définir les conditions d’arrêt de l’algorithme sur des solutions localement optimale. Un maximum local est une solution où toutes les solutions voisines sont de performance strictement plus petite et un *maximum local neutre* est une solution où toutes les solutions du voisinage neutre sont d’évolvabilité maximale strictement plus petite :

**Définition:** Une solution  $s$  est un *maximum local* ssi  $isLocal(s, f, \mathcal{V})$  et une solution  $s$  est une *maximum local neutre* ssi  $isLocal(s, evol_{max}, \mathcal{V}_n)$

L’algorithme 4 est l’algorithme de recherche périscopique associé à l’opérateur HC et les figures 4.2 et 4.3 illustrent son principe.

---

#### Algorithme 4 Recherche Périscopique

---

```

Choisir une solution initiale  $s \in \mathcal{S}$ 
répéter
  tant que non  $isLocal(s, evol_{max}, \mathcal{V}_n)$  faire
     $M = \max\{evol_{max}(s') \mid s' \in \mathcal{V}_n(s) - \{s\}\}$ 
    si  $evol_{max}(s) < M$  alors
      choisir  $s' \in \mathcal{V}_n(s)$  telle que  $evol_{max}(s') = M$ 
    fin si
  fin tant que
  choisir  $s' \in \mathcal{V}(s) - \mathcal{V}_n(s)$  telle que  $f(s') = evol_{max}(s)$ 
   $s \leftarrow s'$ 
jusqu’à  $isLocal(s, f, \mathcal{V})$ 

```

---

### 4.2.2 Algorithmes de comparaisons

Afin de mettre en évidence les potentialités de la métaheuristique de recherche périscopique, la RP est testée sur les variantes des paysages NK et cinq algorithmes sont utilisés pour comparer ces performances : deux algorithmes adaptés aux paysages neutres, le Nageur de Réseau défini section 1 (NR) et l’Extrema Sélection (ES) défini section 1.4.4 ; un algorithme évolutionnaire simple (AES) réduit aux opérateurs de sélection et de mutation (algorithme 6) et deux algorithmes d’exploitation maximale du voisinage, le Hill Climbing (HC) (voir algorithme 7) et le Hill Climbing dont le rayon du voisinage est de longueur de Hamming 2 (HC2) (voir algorithme 8).

Le nageur de réseau teste une solution voisine aléatoirement et la sélectionne seulement si la performance est supérieure ou égale. Le NR explore aléatoirement les RN. Cette recherche permet de tester si une recherche aléatoire est préférable à la RP.

L’extrema sélection est un algorithme évolutionnaire (voir algorithme 5) particulier où l’on utilise une performance endogène. Celle-ci est égale à 0 si la solution est inférieure au produit d’un seuil et de la meilleur performance de la population, et est égale à la distance au centroïde



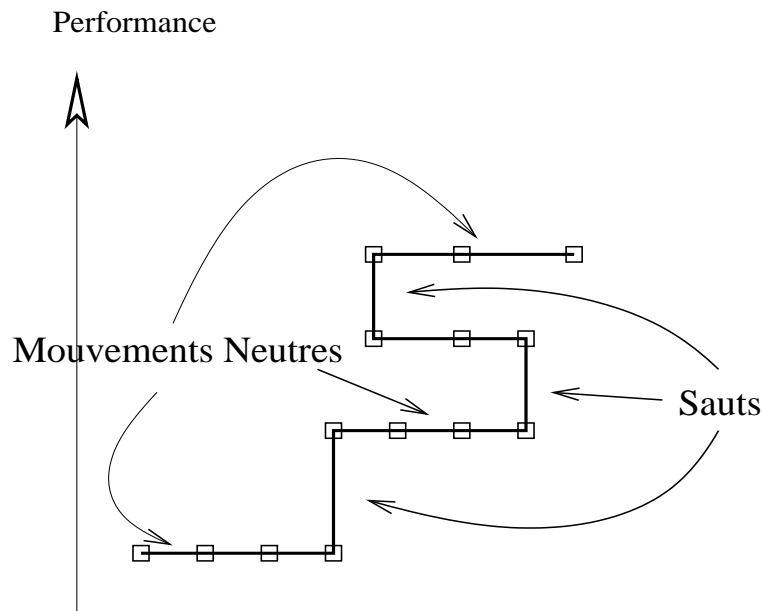


FIG. 4.2 – Illustration de la recherche périscopique : la recherche périscopique alterne une phase de mouvements neutres jusqu’à trouver un maximum local neutre avec un saut qualitatif de performance.

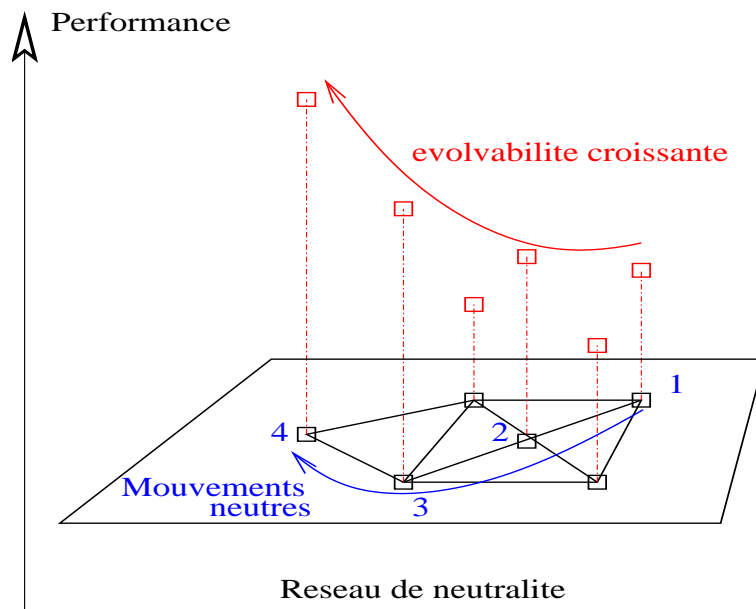


FIG. 4.3 – Illustration de la recherche périscopique : pendant la phase de mouvements neutres, l’algorithme sélectionne la solution voisine du réseau de neutralité dont l’évolvabilité est la plus grande.

de la population dans le contraire. Selon Stewart [128], sur le problème NKp, les meilleures performances sont obtenus pour un algorithme sans opérateur de croisement et où la sélection pour le remplacement est “Steady-state” : à chaque itération une solution est sélectionnée par tournoi selon la performance endogène et remplace la solution de moins bonne performance de

la population. Nous utiliserons les mêmes valeurs de paramètre que dans [128] : une population de taille 100, un tournoi de taille 2, un seuil égale à 0.99 et une probabilité de mutation par bit de 0.01.

---

**Algorithme 5** Extrema sélection (ES)

---

```

step ← 0
Choisir une population initiale de solutions  $\mathcal{P} = (s_i)_{0 < i \leq sizePop}$ 
tant que step ≤ stepMax faire
  Evaluer les performances :  $\forall i \varphi_i \leftarrow f(s_i)$ 
   $M = \max\{\varphi_i \mid 0 < i \leq sizePop\}$ 
  Calculer le centroïde  $C$  de la population
  pour  $i = 1$  to  $sizePop$  faire
    si  $\varphi_i < seuil \times M$  alors
       $\varphi_i \leftarrow 0$ 
    sinon
       $\varphi_i \leftarrow \text{distance}(C, s_i)$ 
    fin si
  fin pour
   $\mathcal{P}' \leftarrow \text{Select}(\mathcal{P}, \varphi)$ 
   $\mathcal{P} \leftarrow \text{Mutation}(\mathcal{P}')$ 
  step ← step + 1
fin tant que

```

---

L'algorithme évolutionnaire simple utilisé est décrit par l'algorithme 6. Ces spécifications sont les mêmes que l'algorithme d'extrema sélection : une population de taille 100, une probabilité de mutation par bit de 0.01 et une sélection "steady-state" réalisée par un tournoi de taille 2.

---

**Algorithme 6** Algorithme évolutionnaire simple (AES)

---

```

step ← 0
Choisir une population initiale de solutions  $\mathcal{P} = (s_i)_{0 < i \leq sizePop}$ 
tant que step ≤ stepMax faire
  Evaluer les performances :  $\forall i \varphi_i \leftarrow f(s_i)$ 
   $\mathcal{P}' \leftarrow \text{Select}(\mathcal{P}, \varphi)$ 
   $\mathcal{P} \leftarrow \text{Mutation}(\mathcal{P}')$ 
  step ← step + 1
fin tant que

```

---

L'algorithme HC est décrit par l'algorithme 7. Il consiste en l'itération de l'opérateur défini dans la section 2.3.1. L'algorithme HC2 exploite un voisinage plus large que l'algorithme HC. A chaque itération, il sélectionne la solution de meilleure performance dans le voisinage de taille 2. Avant de présenter cet algorithme, nous devons donner les définitions suivantes :

**Définition:** Le *voisinage étendu*<sup>19</sup> de  $\mathcal{V}$  est la fonction définie par  $\mathcal{V}^2(s) = \cup_{s_1 \in \mathcal{V}(s)} \mathcal{V}(s_1)$  et  $evol^2$  est la fonction qui associe à chaque solution  $s \in \mathcal{S}$  la performance maximale du voisinage étendu  $\mathcal{V}^2(s) : \forall s \in \mathcal{S}, evol^2(s) = \max\{f(s') \mid s' \in \mathcal{V}^2(s)\}$  L'algorithme HC explore moins de solutions du voisinage et l'algorithme HC2 en explore plus que la recherche périscopique. Ces algorithmes permettrons de savoir si les performances de RP sont une conséquence de la taille du voisinage exploré.

---

<sup>19</sup>remarquons que  $\mathcal{V}(s) \subset \mathcal{V}^2(s)$

---

**Algorithme 7** Hill Climbing (HC)

---

```
step ← 0
Choisir une solution initiale  $s \in \mathcal{S}$ 
répéter
  choisir  $s' \in \mathcal{V}(s)$  telle que  $f(s') = \text{evol}_{\max}(s)$ 
   $s \leftarrow s'$ 
  step ← step + 1
jusqu'à  $\text{isLocal}(s, f, \mathcal{V})$ 
```

---

---

**Algorithme 8** Hill Climbing étendu (HC2)

---

```
step ← 0
Choisir une solution initiale  $s \in \mathcal{S}$ 
répéter
  si  $\text{evol}(s) = \text{evol}^2(s)$  alors
    choisir  $s' \in \mathcal{V}(s)$  telle que  $f(s') = \text{evol}_{\max}^2(s)$ 
  sinon
    choisir  $s' \in \mathcal{V}(s)$  telle que  $\text{evol}_{\max}(s') = \text{evol}_{\max}^2(s)$ 
  fin si
   $s \leftarrow s'$ , step ← step + 1
jusqu'à  $\text{isLocal}(s, f, \mathcal{V}^2)$ 
```

---

### 4.2.3 Résultats expérimentaux

Les six métaheuristiques sont appliquées sur les mêmes instances des paysages  $NK_q$ ,  $NK_M$  et  $NK_p$  dont la neutralité a été étudié au chapitre 3. Nous étudions deux tailles d'espace de recherche,  $N = 16$  et  $N = 64$ . Pour  $N = 16$ , le paramètre  $K$  décrit l'ensemble  $\{1, 2, 3, 5, 8\}$ . Les trois paramètres  $q$ ,  $M$  et  $p$  ajustant la neutralité décrivent les ensembles :  $q \in \{2, 3, 4, 10\}$ ,  $M \in \{16, 32, 48, 160\}$  et  $p \in \{0.5, 0.8, 0.9, 0.95, 0.99\}$ . Pour chaque valeur des paramètres, 50 instances indépendantes de paysages sont générées. Pour  $N = 64$ , le paramètre  $K$  décrit l'ensemble  $\{2, 4, 8, 12, 16\}$ . Les trois paramètres  $q$ ,  $M$  et  $p$  ajustant la neutralité décrivent les mêmes ensembles que précédemment. Pour chaque valeur des paramètres, 10 instances indépendantes de paysage sont générées.

Pour chaque instance,  $10^2$  exécutions indépendantes de chaque algorithme sont réalisées. Pour le NR, la constante stepMax est égale à  $300^{20}$ . Pour les algorithmes évolutionnaires, le nombre d'itérations est fixé à 2000.

#### Performances moyennes

Pour chaque instance, nous calculons la performance moyenne et l'écart-type de chaque algorithme, puis calculons de nouveau la moyenne et l'écart-type de ces résultats sur l'ensemble des instances. Les figures 4.7, 4.8 et 4.9 présentent les moyennes des performances obtenues pour les trois familles de paysages pour  $N = 16$  et les figures 4.7, 4.8 et 4.9 pour  $N = 64$ . Le détail des résultats expérimentaux sont présentées en annexe .

Les performances sont différentes selon la taille de l'espace de recherche. Lorsque l'espace est petit avec  $N = 16$ , les algorithmes évolutionnaires AE et SE obtiennent les meilleures performances sur toutes les instances de paysage. L'AE simple a toujours des performances supérieures ou égales à l'ES. Pour les paysages  $NK_q$ , l'algorithme HC est le moins performant. Lorsque la neutralité est plus importante et l'épistasie faible, le NR et la RP ont de meilleure performance

---

<sup>20</sup>Dans nos expériences, l'algorithme converge avant cette limite

que le HC2. Pour les paysages  $NK_M$ , le HC et la RP ont quasiment les mêmes performances. Pour toutes les valeurs des paramètres, les algorithmes peuvent être classés par ordre décroissant de leurs performances : AE, ES, HC2, RP, HC, NR. Pour les paysages  $NK_p$ , Les algorithmes évolutionnaires sont les plus performants. La RP est la moins performante en particulier lorsque l'épistasie est faible. Les performances du RN sont supérieures à celles du HC2 lorsque l'épistasie est faible. Pour une taille d'espace de recherche trop petite, la recherche aléatoire peut suffire à trouver de bonnes solutions. Pour ces espaces, le rapport entre les caractéristiques du paysage et les capacités des algorithmes n'est donc pas mis en relief et ne permet pas de conclure sur les potentialités de chaque algorithme.

Lorsque l'espace de recherche est de grande taille avec  $N = 64$ , les résultats sont différents. Pour les paysages  $NK_q$ , la Recherche Périscopique obtient les meilleures performances pour toutes les valeurs des paramètres d'épistasie  $K$  et de neutralité  $q$ . La différence est d'autant plus importante que l'épistasie est importante. Pour les paysages  $NK_M$ , la recherche périscopique obtient de faibles performances excepté lorsque la neutralité est la plus faible pour  $M = 160$ . Les HC et NR obtiennent les meilleures performances lorsque le paramètre d'épistasie est strictement inférieur à 16. Dans ce dernier cas, l'ES adapté à la neutralité a la meilleure performance. Pour les paysages  $NK_p$ , la RP obtient les meilleures moyennes, excepté lorsque la neutralité est trop faible (par exemple pour  $p = 0.5$ , l'AE a les meilleures performances), ou bien lorsque la neutralité est trop importante ( $p = 0.99$  et  $K = 2, 4$ , l'ES a les meilleures performances).

La recherche périscopique est moins efficace lorsque le degré de neutralité moyen est trop important, en effet dans ce cas l'évolvabilité maximale est constante et la RP ne peut se déplacer sur les RN. Mais, cet argument n'est pas suffisant pour rendre compte des performances, par exemple, les paysages  $NK_M$  avec  $K = 8$  et  $M = 48$  et  $NK_p$  avec  $K = 4$  et  $p = 0.9$  ont le même degré moyen de neutralité, pourtant la RP a de meilleure performance sur le paysage  $NK_p$ .

Les performances de la RP ne s'expliquent pas non plus par une plus grande exploration du voisinage. À chaque itération, la RP explore  $(d + 1)N$  solutions voisines où  $d$  est le degré de neutralité de la solution courante. Le HC2 explore  $N(N - 1)/2$  solutions voisines, or le HC2 a de moins bonnes performances. On peut en déduire que la RP réalise un meilleur compromis entre exploration et exploitation du voisinage en concentrant la recherche sur les solutions voisines de même performance.

Ces résultats confirment les mesures sur la structure des RN. Pour les paysages  $NK_M$ , la structure des RN est plus faible (l'autocorrélation des degrés et de l'évolvabilité maximale est plus faible), la recherche périscopique obtient de moins bonnes performances. En revanche, lorsque la structure des RN est plus forte, comme dans les paysages  $NK_q$  et  $NK_p$ , la recherche périscopique devient plus performante.

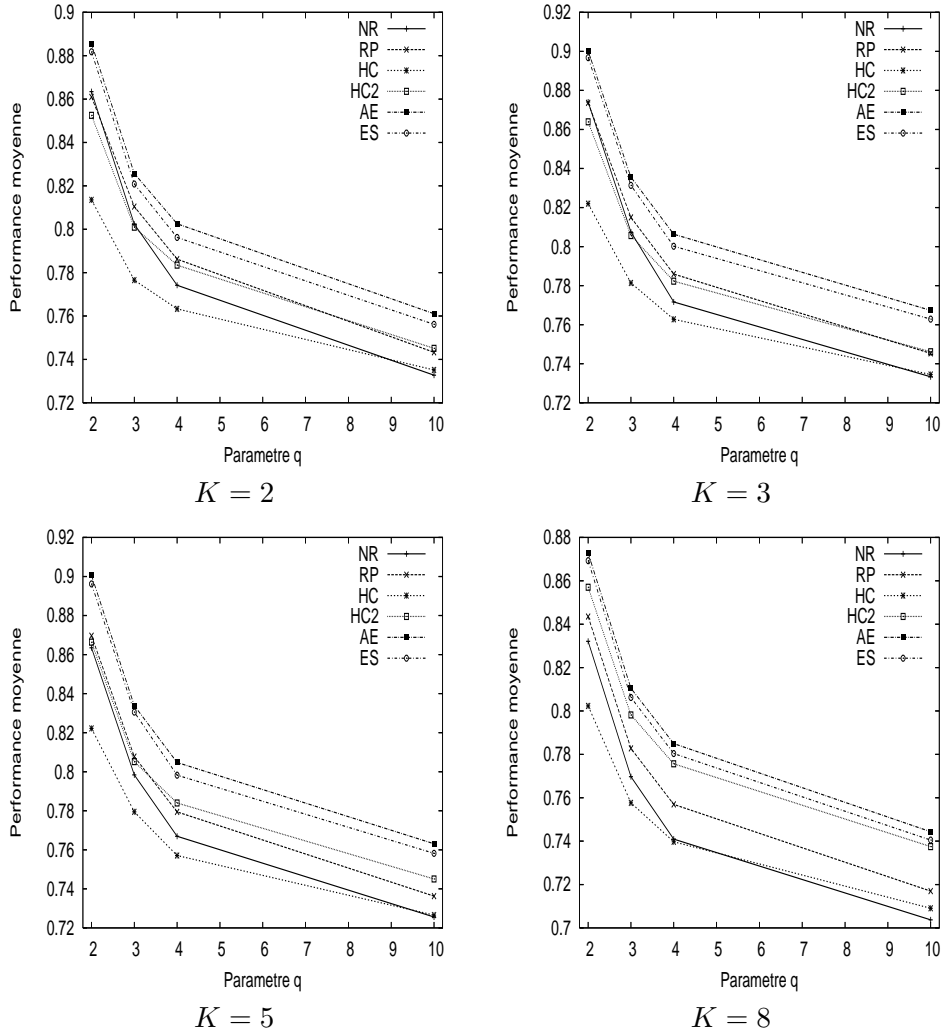


FIG. 4.4 – Performances moyennes pour les différentes métaheuristiques pour les paysages  $NK_q$  avec  $N = 16$ .

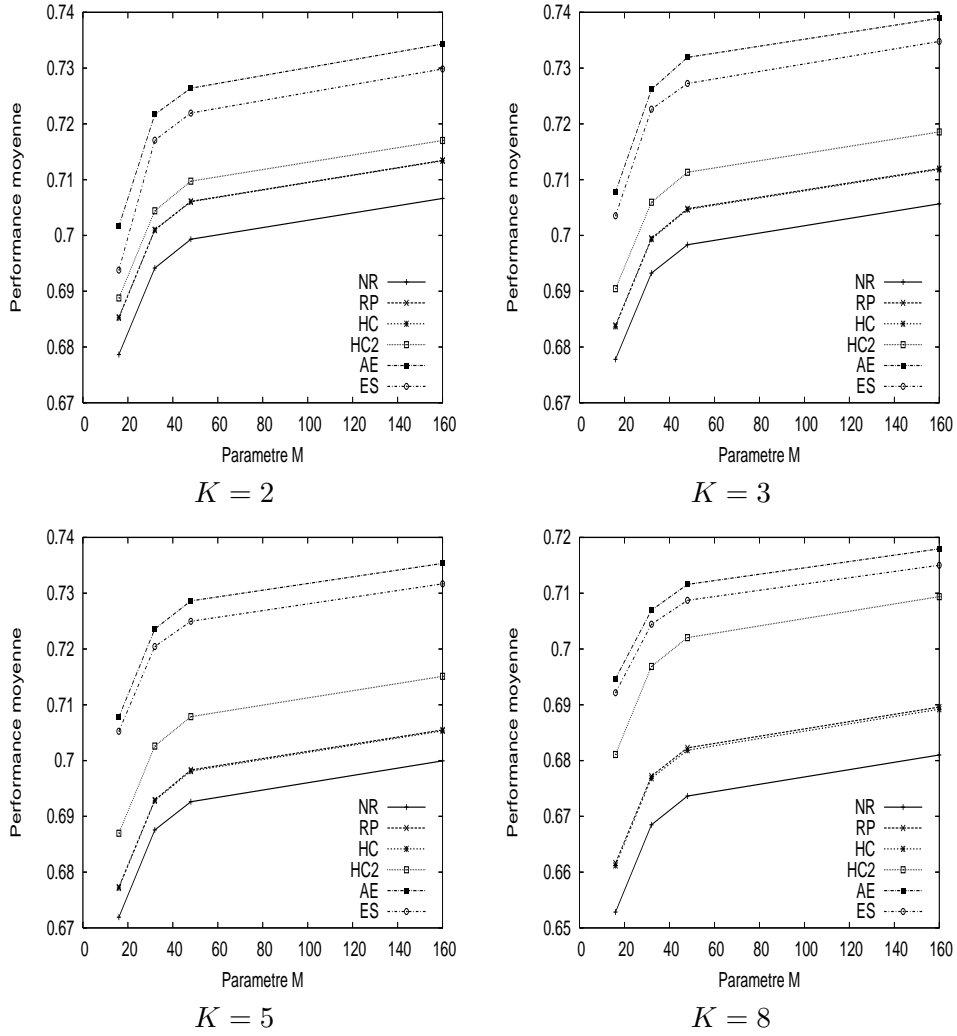


FIG. 4.5 – Performances moyennes pour les différentes métaheuristiques pour les paysages  $NK_M$  avec  $N = 16$ .

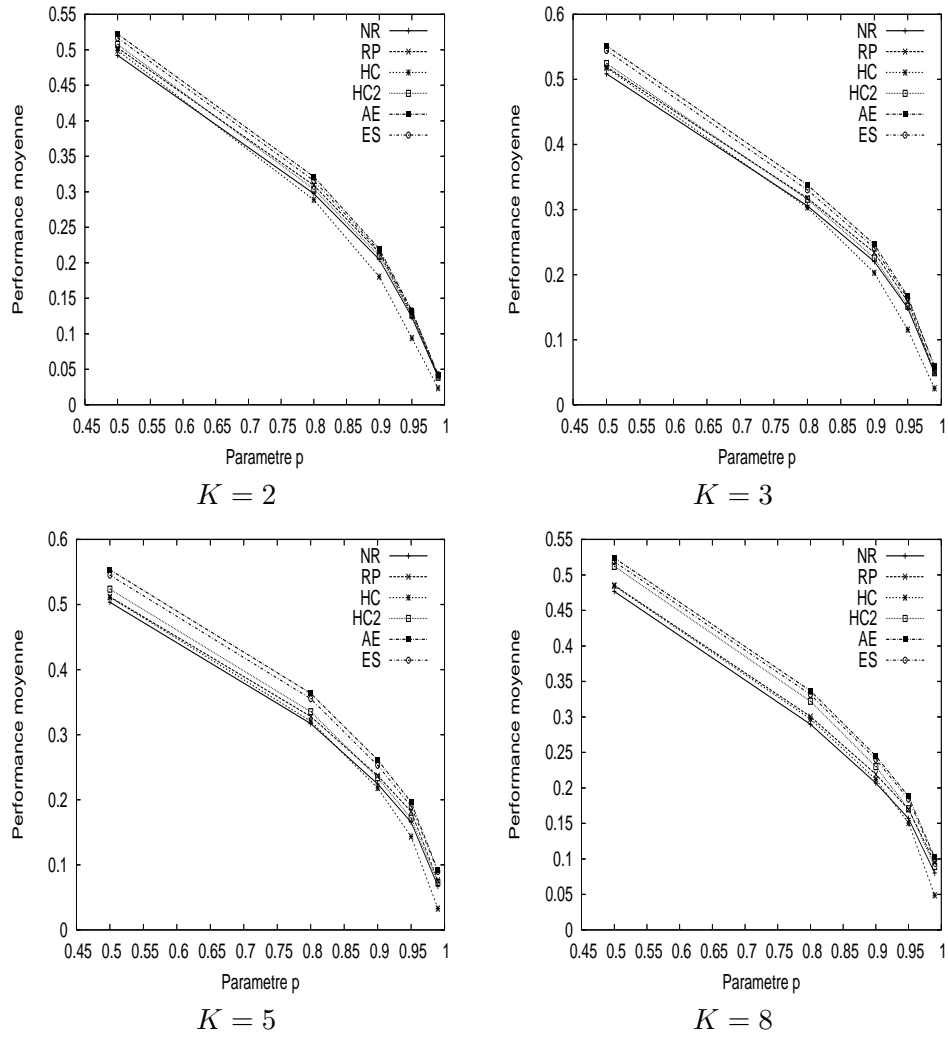


FIG. 4.6 – Performances moyennes pour les différentes métaheuristiques pour les paysages  $NK_p$  avec  $N = 16$ .

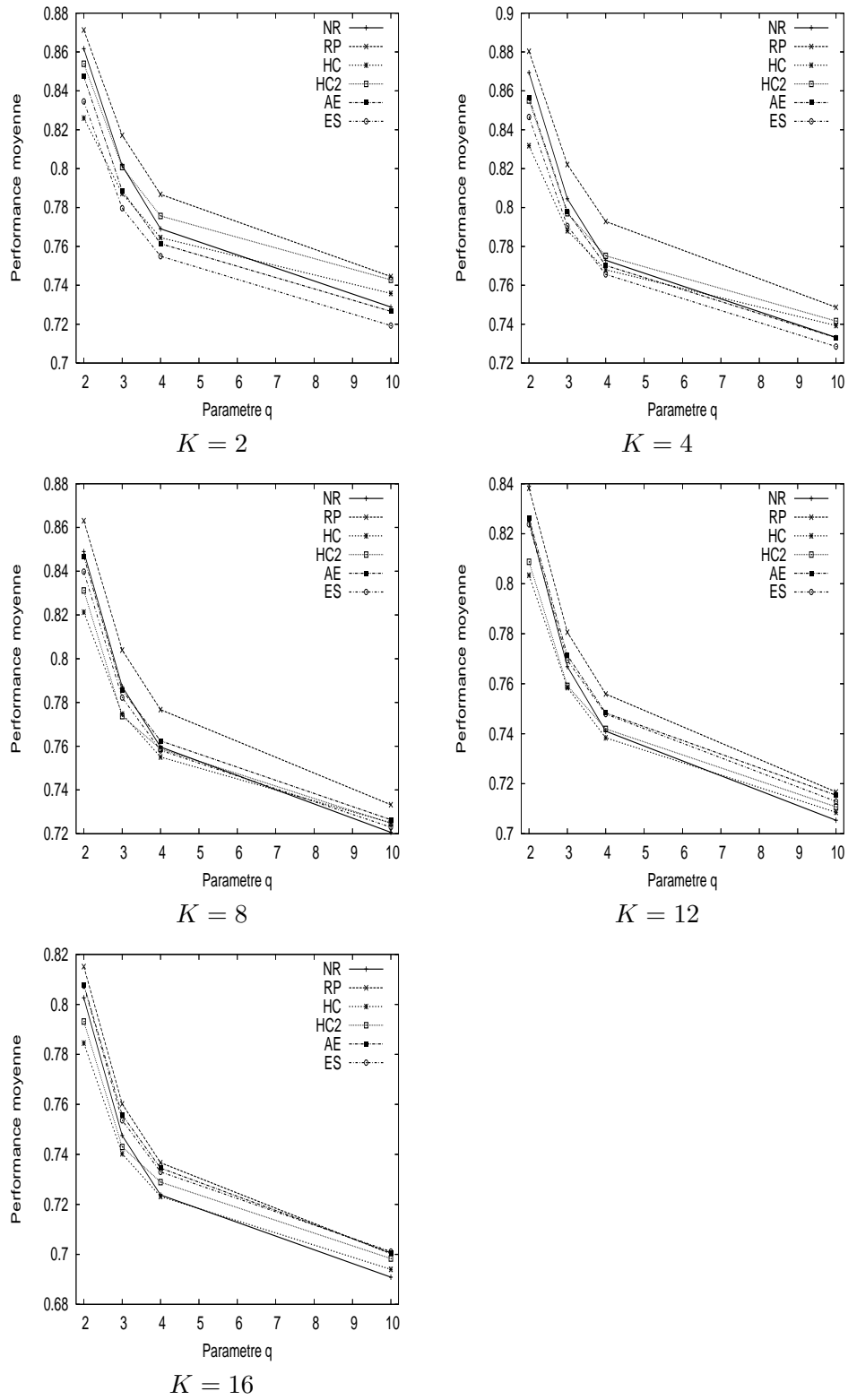


FIG. 4.7 – Performances moyennes pour les différentes métaheuristiques pour les paysages  $NK_q$  avec  $N = 64$ .



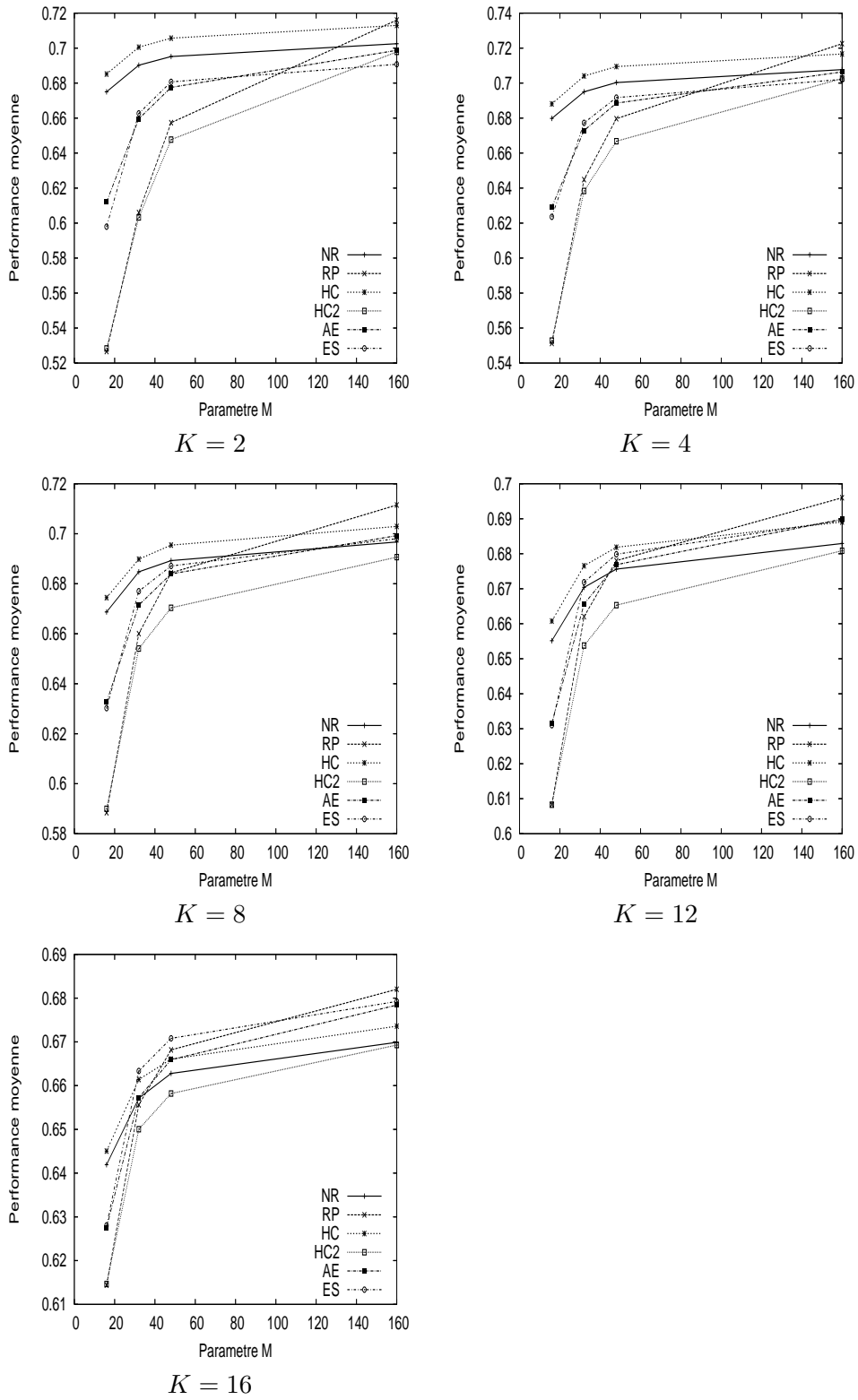


FIG. 4.8 – Performances moyennes pour les différentes métaheuristiques pour les paysages  $NK_M$  avec  $N = 64$ .

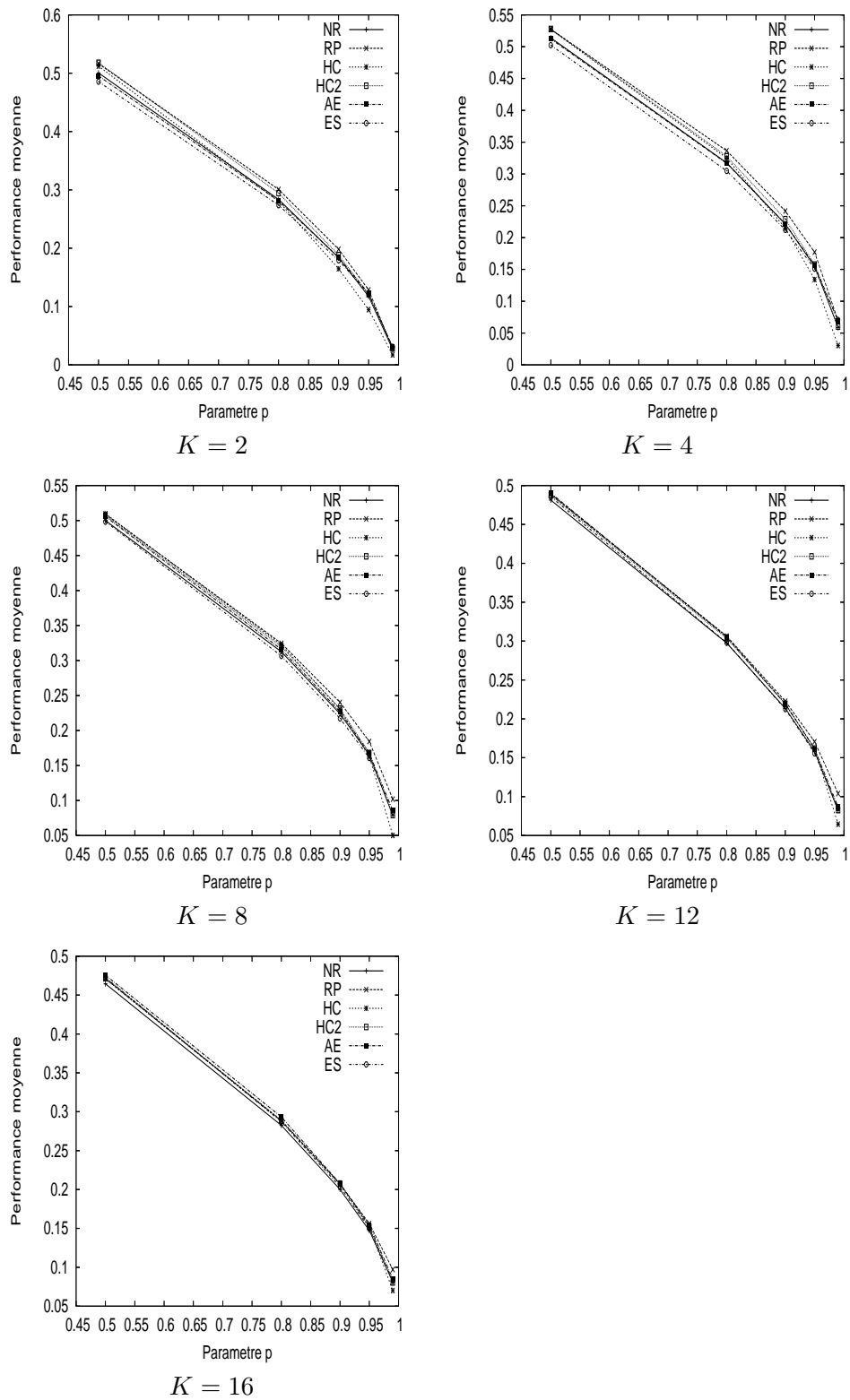


FIG. 4.9 – Performances moyennes pour les différentes métaheuristiques pour les paysages  $NK_p$  avec  $N = 64$ .

### 4.3 Problème massivement neutre : AC du problème de majorité

Dans cette section, nous analysons le paysage adaptatif relatif au problème de majorité. Le problème de majorité consiste à trouver la règle d'un automate cellulaire qui réalise la tâche de classification des configurations initiales par la densité. Les automates cellulaires sont des calculateurs universaux et leur dynamique peut être complexe et imprédictible. Cette tâche est un bon exemple d'un phénomène d'émergence dans un système complexe.

Ce paysage est un exemple représentatif de paysage neutre où la neutralité est particulièrement importante, beaucoup de solutions ont des performances égales. Le problème de majorité est un problème d'apprentissage puisqu'il consiste à trouver une bonne règle dont la performance est évaluée à partir d'un échantillon de test. Nous allons montrer que l'erreur d'évaluation amène à définir une notion de neutralité qui peut être utile à l'optimisation par métaheuristique.

Dans un premier temps, nous allons étudier le paysage dans son ensemble et montrer que le nombre important de solution de performance nulle ne permet pas une étude statistique. Ensuite, nous étudierons ce paysage par le "haut" en considérant les meilleurs optima locaux connus. L'exploitation des similarités entre ces différents optima et des symétries du problème, permet de définir un sous-espace, appelé *Olympe*, à partir des six symétries des meilleurs optima connus, les "dieux" de notre Olympe. Ce sous-espace Olympe sera analysé. Enfin, nous montrerons que l'optimisation par l'algorithme évolutionnaire dans ce sous-espace en considérant la neutralité du paysage est plus facile.

#### Introduction

Les automates cellulaires (AC) sont des systèmes dynamiques discrets qui sont étudiés depuis de nombreuses années à cause de leur description simple et leur large spectre de dynamiques possibles [18, 143]. Les AC sont des calculateurs universels et leur dynamique peut être complexe et imprédictible, bien que beaucoup d'entre eux présentent des dynamiques simples comme des points fixes ou des cycles attracteurs. Dans cette section, nous étudions un AC qui réalise une tâche de calcul "simple" qui consiste à classer les configurations initiales d'un AC à deux états. Il s'agit de savoir si une configuration initiale contient une majorité de '1'. En dépit de son apparente simplicité, il est difficile pour un système à base de règles locales d'effectuer un "calcul" global ; cela nécessite de réaliser la circulation d'information à travers tout l'automate. Comme tel, il est un parfait exemple de paradigme d'un phénomène d'émergence dans un système complexe. En effet, la configuration finale de l'AC est une propriété émergente d'un système d'agents en interaction locale. En fait, il a été prouvé qu'aucun AC ne peut réaliser tâche cette parfaitement sur l'ensemble de toutes les configurations initiales possibles [77]. Toutefois, plusieurs AC performants qui réalisent la tâche de densité ont été construit "à la main" ou trouvé à l'aide de métaheures, en particulier à l'aide d'algorithmes évolutionnaires [89, 87, 118, 4, 66]. Pour une revue récente sur ces travaux depuis dix ans, on pourra consulter [25].

Tous ces travaux ont montré empiriquement qu'il était difficile de trouver un AC réalisant la tâche de densité. Toutefois, il n'y a pas eu d'étude, à notre connaissance, sur les raisons particulières qui rendent ce paysage d'adaptation difficile. Dans la suite, nous allons étudier les caractéristiques du paysage de la tâche de densité. Cette étude complète les travaux de Hordijk [56] à propos d'un autre problème sur les AC : la tâche de synchronisation [27].

### 4.3.1 Automate cellulaire et problème de majorité

#### Automate cellulaire

Les AC sont des système dynamiques pour lesquels le temps et l'espace sont discrets. Un AC standard consiste en un ensemble de cellules indexées par  $Z^d$  (les cellules sont disposées sur une grille). Chaque cellule peut être dans un nombre fini d'états actualisé de manière synchrone par pas de temps discret, selon une règle locale identique pour toute les cellules. Dans ce travail, nous considérons seulement les AC booléens pour lesquels les états des cellules sont 0 ou 1. Le cellules sont disposées selon un tableau de dimension 1 (AC linéaire), une cellule est connectée aux  $2r + 1$  plus proches cellules voisines (en incluant la cellule elle-même), qui forment le voisinage de la cellule centrale.  $r$  est appelé le *rayon* du voisinage. La *règle de transition locale* utilisée par chaque cellule est spécifiée par une table dont les entrées sont formées par toutes les combinaisons possibles d'états du voisinage. L'état d'une cellule au pas suivant de temps est déterminé par les états courants des cellules du voisinage. Ainsi, pour un AC linéaire de rayon  $r$ , la règle de transition peut s'écrire par :

$$s_{t+1}^i = \phi(s_t^{i-r}, \dots, s_t^i, \dots, s_t^{i+r}),$$

où  $s_t^i$  est l'état de la cellule  $i$  au temps  $t$ ,  $\phi$  représente la règle de transition local.

Le terme *configuration* désigne une affectation de 0 et 1 à toutes les cellules à un pas de temps donné. Elle peut être décrite par  $\mathbf{s}^t = (s_t^0, s_t^1, \dots, s_t^{N-1})$ , où  $N$  est la taille de la grille. Les AC ici sont linéaires avec des conditions aux bords périodiques  $s_t^{N+i} = s_t^i$  *i.e.* , ils ont donc une topologie en anneau.

Une règle de transition globale  $\Phi$  peut être définie en appliquant à chaque cellule en parallèle la règle de transition locale

$$\mathbf{s}_{t+1} = \Phi(\mathbf{s}_t).$$

Cette règle globale  $\Phi$  définit ainsi l'évolution temporelle de toutes les cellules de l'AC. Pour visualiser la dynamique de l'AC, on peut utiliser un diagramme espace-temps, où l'axe horizontal représente la configuration  $\mathbf{s}^t$  à un certain pas de temps  $t$  et l'axe vertical représente les pas successifs de temps de haut en bas (par exemple, voir la figure 4.10).

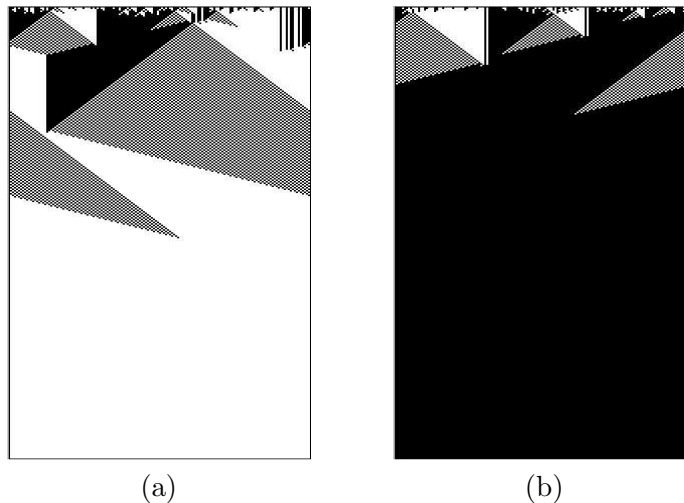


FIG. 4.10 – Diagramme espace-temps pour la règle GLK. La densité de 0 est 0.476 pour (a) et 0.536 pour (b). L'état 0 est représenté en blanc et le 1 en noir.

## Le problème de majorité

Le tâche de densité est un problème de calcul distribué typique pour les AC. Pour un AC de taille finie  $N$ , il est défini comme suit : Soit  $\rho_0$  la proportion de cellules dans l'état 1 dans la configuration initiale (CI)  $s_0$ . La tâche consiste à déterminer si  $\rho_0$  est plus petit ou plus grand que  $1/2$ . Dans cette version, le problème est aussi appelé le *problème de majorité*. Si  $\rho_0 > 1/2$  alors l'AC doit converger, après un nombre de pas de l'ordre de la taille de la grille  $N$ , vers une configuration point-fixe constituée uniquement d'états 1, que nous notons  $(1)^N$  ; dans le cas contraire, l'AC doit converger vers la configuration point-fixe  $(0)^N$ . Ici,  $N$  est égale à 149, cette valeur a été choisie habituellement dans les recherches sur la tâche de densité <sup>21</sup>. Le problème de densité est trivial pour un ordinateur qui possède une mémoire centrale. Il lui suffit juste de parcourir l'ensemble des cellules et de compter les états à 1. Cependant, il devient non trivial pour un AC linéaire de rayon petit où un AC ne peut seulement transférer une information localement à une vitesse finie alors que la densité est une propriété globale de la configuration [89].

Il a été démontré que la tâche de densité ne peut être pas résolue parfaitement à l'aide d'un AC de rayon fini [77] <sup>22</sup>. Elle a été également parfaitement résolue à l'aide d'une combinaison d'AC [40].

## Résultats des travaux précédents

L'absence d'une solution parfaite au problème n'empêche pas de rechercher des solutions imparfaites de qualité que l'on mesure par le taux de CI bien classées, la meilleure possible. En général, étant donné un comportement global désiré pour un AC, il est extrêmement difficile de construire la règle locale de l'AC qui donne le comportement global attendu. Ceci en raison des non-linéarités et des effets collectifs à grande échelle qui ne peuvent pas en général être prévus. Bien que l'évaluation sur toutes les règles possibles sont hors de portée, excepté pour un AC élémentaire ( $r = 1$ ), une résolution possible du problème peut être effectuée à l'aide d'un algorithme évolutionnaire (AE), comme le proposa le premier Packard [96] et plus tard développé par Mitchell *et al.* [87, 89].

Mitchell *et al.* ont réalisé de nombreuses études sur l'émergence de stratégies de synchronisation des cellules dans l'AC (avec  $N = 149$ ) durant l'évolution d'un AE [87, 89]. Leurs résultats sont significatifs puisqu'ils représentent un des quelques exemples où la dynamique du calcul émerge dans des systèmes spatialement étendus et complexes peut être comprise. En résumé, ces résultats peuvent être subdivisés en deux : ceux qui concernent l'histoire de l'évolution de l'AE qui mène à une solution de bonne qualité et ceux qui concernent l'analyse de la stratégie de l'AC obtenue finale. Pour les premiers, il a été observé, lorsque l'évolution d'un AE mène à une bonne règle, que la dynamique est celle des équilibres ponctuelle décrite section 4.1. A chaque saut qualitatif, la stratégie de la meilleure règle se complexifie par rapport aux précédentes. Concernant, les résultats sur la règle finale obtenue, il a été observé que la plupart des exécutions d'un AE trouvent des AC de stratégie plus simple telle que les stratégies d'expansion de blocs adjacents de 0 ou de 1. Ces stratégies non sophistiquées utilisent de manière trop forte l'information locale pour décider de la densité globale, ayant pour conséquence, que seuls les CI de forte et faible densité sont correctement classés. Ces AC ont des performances autour de 0.6, ce qui signifie que le taux de configurations correctement bien classée est de 60%.

Quelques exécutions d'AE donnent des règles aux caractéristiques plus sophistiquées aux performances autour 0.77. Toutefois, seuls neuf exécutions sur trois cents d'AE fournissent de bonnes règles. Ce qui suggère que le paysage adaptatif relatif au problème de majorité est très

---

<sup>21</sup>si  $N$  est impair, la valeur  $\rho_0 = 0.5$  pour laquelle le problème n'est pas définie, est impossible

<sup>22</sup>bien qu'une version légèrement modifiée de cette tâche peut être résolue parfaitement [17]

difficile à optimiser. Les règles performantes obtenues utilisent des signaux qui communiquent à travers la grille des informations spatiales et temporelles sur la densité locale. Un exemple d’une telle stratégie est donné figure 4.10, où le comportement de la règle dénommée GLK est représenté [89]. La règle GLK a été construite “à la main” mais son comportement est similaire aux meilleures règles trouvées par AE. Crutchfield *et al* ont développé des méthodes pour étudier la communication par signaux et l’émergence de calcul dans un AC, regroupée sous le nom de “mécanismes de calcul”<sup>23</sup>. Ils décrivent le calcul intrinsèque réalisé par l’automate en terme de domaines réguliers, particules et d’interaction de particules. Pour plus de détail, on pourra consulter [53, 57, 25].

Andre *et al.* [4] ont pu trouver une très bonne règle en utilisant la programmation génétique. Mais, les meilleures règles actuellement connues ont été trouvées par Juillé et Pollack [66] en utilisant un AE qui utilise la coévolution des règles et l’échantillon de CI de test. La performance de leur règle est d’environ 0.86.

### 4.3.2 Analyse du paysage adaptatif du problème de majorité

#### Définition du paysage

Comme Mitchell [89], nous considérons des AC sont de rayon  $r = 3$  et des configurations de taille  $\lambda = 149$ . L’ensemble  $\mathcal{S}$  des solutions potentielles est l’ensemble des chaînes binaires représentant toutes les règles possibles des AC. La relation de voisinage  $\mathcal{V}$  est le voisinage de Hamming de taille 1. La taille de  $\mathcal{S}$  est donc  $2^{2^{2r+1}} = 2^{128}$ , et chaque automate peut être tester sur  $2^{149}$  configurations initiales (IC) possibles, ce qui donne  $2^{277}$  calculs pour une énumération exhaustive de l’espace de recherche. L’énumération exhaustive est donc trop large pour être effectuée. La performance d’une règle peut être définie de différentes manières, qui conduit à différentes possibilités de performance pour les solutions et donc de définition de paysage adaptatif. Dans ce travail, nous utilisons une performance basée sur la proportion de configurations correctement classées parmi un échantillon de configurations initiales de taille  $n$ . Nous appelons *performance standard* (ou plus simplement performance lorsqu’il n’y a pas d’ambiguïté) la performance relative à un échantillon où les CI sont choisies de manière équiprobable sur l’ensemble de toutes les configurations possibles (chaque cellule à une probabilité 1/2 d’être dans l’état 0). La performance standard est une mesure difficile car il y a une prédominance de CI de densité proche de 0.5.

L’erreur d’évaluation de la performance conduit à introduire de la neutralité dans le paysage. La performance standard ne peut pas être connue parfaitement à cause des variations stochastiques de l’échantillon de CI. Si les CI sont choisies de manière indépendante, alors la performance  $f$  d’une solution suit une loi normale  $\mathcal{N}(f, \frac{\sigma(f)}{\sqrt{n}})$ , où  $\sigma(f)$  est l’écart-type d’un échantillon de moyenne  $f$  et de taille  $n$ . Pour un échantillon relatif à la performance standard,  $\sigma^2(f)$  est égal à  $f(1 - f)$ , la variance d’une loi de Bernouilli. Alors, deux solutions voisines  $s$  et  $s'$  sont des voisins neutres (*isNeutral*( $s, s'$ ) est vrai) si un t-test accepte l’égalité de  $f(s)$  et  $f(s')$  à 95% de confiance (cf figure 4.11). Le nombre maximum de valeurs de performances statistiquement différents est 113 pour  $n = 10^4$ , 36 pour  $n = 10^3$  et 12 pour  $n = 10^2$ .

#### Premières mesures statistiques

La figure 4.12-a montre la D.O.S. du paysage en utilisant un échantillon de taille  $n = 10^4$  uniforme de l’espace de recherche. Le nombre de points est  $4.10^3$  et sur ce nombre 3979 ont une performance nulle. Clairement, le paysage apparaît difficile puisque la queue de la distribution n’est pratiquement pas existante. La figure 4.12-b montre la D.O.S. en utilisant l’algorithme

---

<sup>23</sup>computational mechanics, en anglais

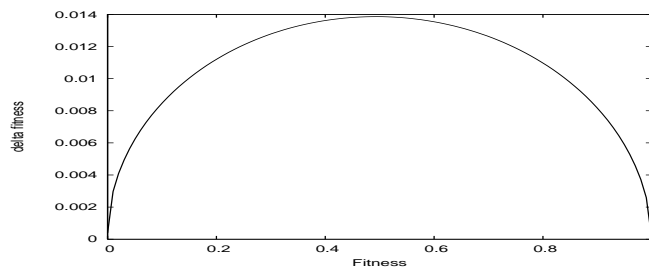


FIG. 4.11 – Erreur de l'évaluation de la performance standard donné par un t-test pour un échantillon de taille  $n = 10^4$ .  $isNeutral(s, s')$  est vrai si la différence de performance entre les deux solutions est en dessous la courbe.

de Métropolis-Hastings. Cette fois, sur les  $4 \cdot 10^3$  solutions échantillonnées, seules 176 ont une performance nulle, et la D.O.S. montre une distribution plus uniforme des performances.

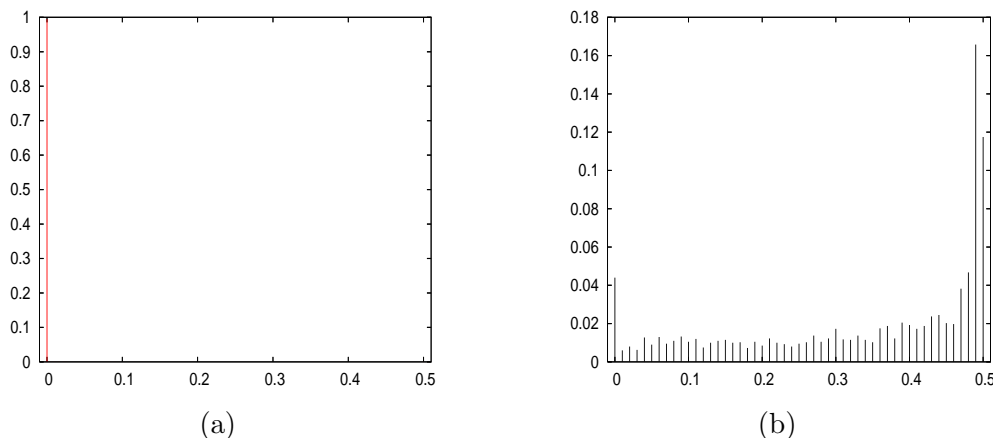


FIG. 4.12 – D.O.S. obtenus par échantillonnage équiprobable de l'espace de recherche (a) et en utilisant l'algorithme de Métropolis-Hastings (b).

Il est important de remarquer le nombre considérable de solutions échantillonnées avec une performance approximativement égale à 0.5. De plus, aucune solution de performance supérieure à 0.55 n'a été échantillonné.

L'autocorrélation au cours d'une marche aléatoire, afin de mesurer la rugosité du paysage, n'est pas significative en raison du nombre trop important de solutions de performance nulle. Ainsi, cette mesure n'est pas présentée ici.

Les corrélations performance-distance (CPD) aux optima locaux, calculée à partir d'un échantillon de  $4 \cdot 10^3$  solutions en utilisant l'algorithme de Métropolis-Hastings sont reportées dans la table 4.1. Chaque valeur a été obtenue en utilisant les meilleurs optima locaux connus jusqu'alors (cf. section 4.3.2). Le CPD est proche de zéro pour l'optimum DAS. Pour l'optimum ABK, le CPD est proche de  $-0.15$ , valeur identifiée par Jones [65] comme le seuil entre un problème difficile et trivial à optimiser. Pour les autres optima, le CPD est proche de  $-0.10$ . Il ne fournit donc pas d'information sur la difficulté du problème.

La figure 4.13 montre le nuage adaptatif, et l'ensemble des segments utilisés pour calcul le CPN. L'algorithme de Métropolis-Hastings permet d'échantillonner les solutions de performance

TAB. 4.1 – CPD pour les six meilleurs optima connus, calculés à partir d’un échantillon de taille  $4.10^3$  en utilisant l’algorithme de Métropolis-Hastings.

Règles	GLK [42]	Davis [4]	Das [28]	ABK [4]	Coe1 [67]	Coe2 [67]
FDC	-0.1072	-0.0809	-0.0112	-0.1448	-0.1076	-0.1105

supérieure à zéro. La valeur du CPN de ce paysage est  $-0.7133$ , ce qui semble indiquer qu’il est difficile pour une métaheuristique d’atteindre une valeur proche de 0.5, et encore plus l’obtenir une performance plus grande.

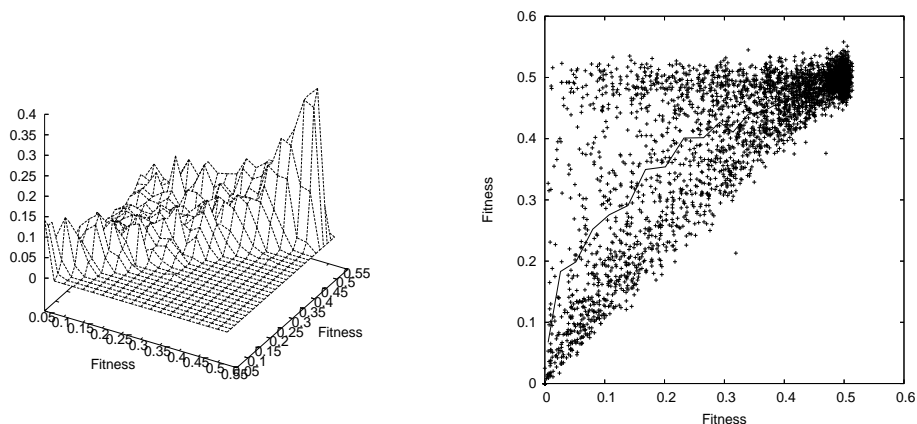


FIG. 4.13 – Nuage adaptatif et segments utilisés pour calculer le CPN. L’algorithme de Métropolis-Hastings a été utilisé pour créer l’échantillon de solutions.

## Neutralité

Le coût d’évaluation ne permet pas d’analyser beaucoup de réseaux de neutralité. Dans cette section, nous analysons deux réseaux de neutralité (RN) significatifs de grande taille. Un grand nombre d’AC résolvent le problème de majorité pour seulement la moitié des CI parce qu’ils convergent presque toujours vers l’une des configurations  $(0)^N$  ou  $(1)^N$  et ont une performance d’environ 0.5. Mitchell [87] les appelle les “stratégies par défaut” et remarque qu’il constitue la première étape dans l’évolution d’une population d’un AE avant de découvrir des stratégies plus évoluées associées à l’“expansion de blocs” (cf section 4.3.1). Nous étudions donc le RN d’une solution de performance proche de 0.5, noté  $RN_{0.5}$ , pour comprendre le lien entre les propriétés du RN et l’évolution d’un AE. L’autre RN, noté  $RN_{0.76}$ , a une performance d’environ 0.7645 et contient une solution voisine de celle découverte par Mitchell *et al.* La description de ce “haut” RN peut donner des indications sur la dynamique qui permet de découvrir des solutions de meilleure performance.

Dans nos expérimentations, nous réalisons 5 marches neutres sur  $RN_{0.5}$  et 19 sur  $RN_{0.76}$ . Chaque marche neutre commence à partir de la même solution sur chaque RN. Nous explorons le RN en augmentant strictement la distance de Hamming à la solution initiale à chaque pas de la marche. La marche neutre s’arrête lorsqu’il n’y a plus de mouvement neutre à effectuer qui augmente la distance. La longueur de marche est donc au plus 128. En moyenne, la longueur de la marche sur  $RN_{0.5}$  est 108.2 et 33.1 sur  $RN_{0.76}$ . Le diamètre (voir section 3.1.1) de  $RN_{0.5}$  est



donc plus grand que celui de  $RN_{0.76}$ .

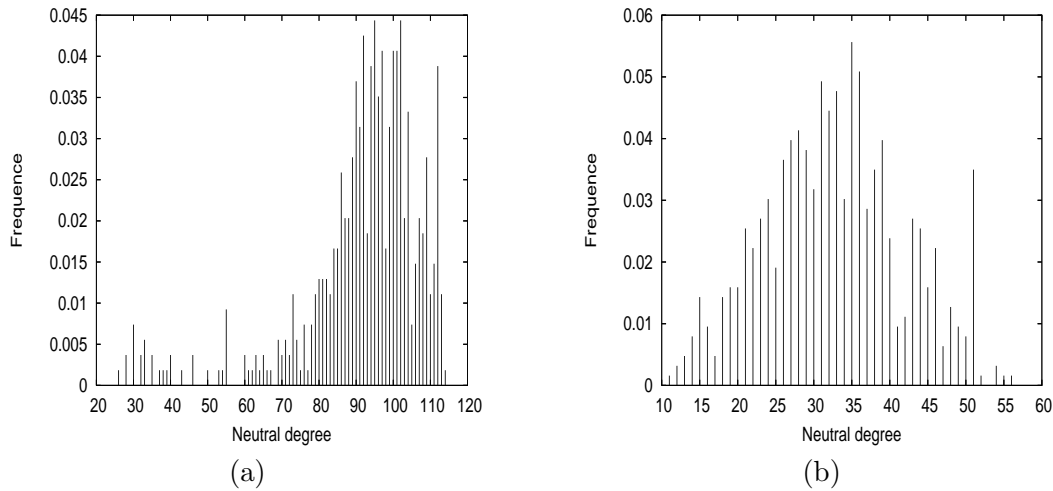


FIG. 4.14 – Distribution du degré de neutralité au cours des marches neutres sur  $RN_{0.5}$  (a) et  $RN_{0.76}$  (b).

La figure 4.14 montre la distribution des degrés de neutralité collectés au cours des marches neutres. Les distributions sont proches de distributions normales pour  $RN_{0.76}$ . Pour  $RN_{0.5}$  la distribution est biaisée et approximativement bimodale avec un pic important autour de 100 et un plus petit autour de 32. Le degré de neutralité moyen sur  $RN_{0.5}$  est de 91.6 et l'écart-type est 16.6 ; sur  $RN_{0.76}$ , la moyenne est 32.7 et l'écart-type est 9.2. Le degré de neutralité pour  $RN_{0.5}$  est très important : 71.6% des solutions voisines sont des voisins neutres. Pour  $RN_{0.76}$ , il y a 25.5% de solutions voisines neutres. Ce nombre peut être comparé au degré moyen du paysage  $NK_q$  avec  $N = 64$ ,  $K = 4$  and  $q = 2$  qui est de 26% (cf tableau 3.4).

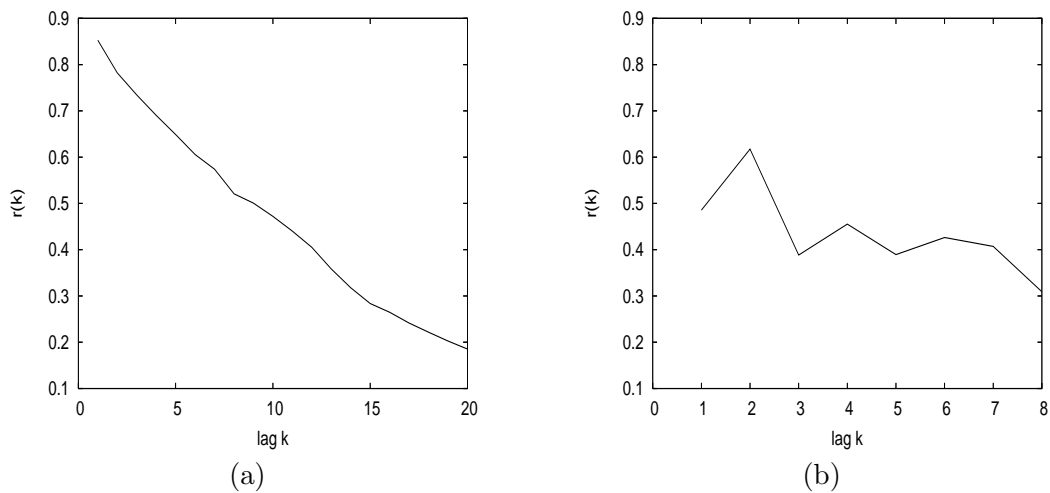


FIG. 4.15 – Estimation de la fonction d'autocorrélation des degrés de neutralité pour les marches neutres pour  $RN_{0.5}$  (a) et pour  $RN_{0.76}$  (b).

La figure 4.15 donne une estimation de la fonction d'autocorrélation des degrés de neutralité au cours des marches neutres. La fonction d'autocorrélation est calculée pour chaque marche neutre et nous représentons la moyenne des coefficients obtenus. Pour les RN, la corrélation est

non nulle. Celle-ci est plus importante pour  $RN_{0.5}$  ( $\rho(1) = 0.85$ ) que pour  $RN_{0.76}$  ( $\rho(1) = 0.49$ ). De cette autocorrélation des degrés de neutralité, on peut conclure que le graphe des RN n'est pas aléatoire. Les variations du degré de neutralité ne sont pas aléatoires, il existe des zones homogènes de degré de neutralité pour  $RN_{0.5}$  et  $RN_{0.76}$ .

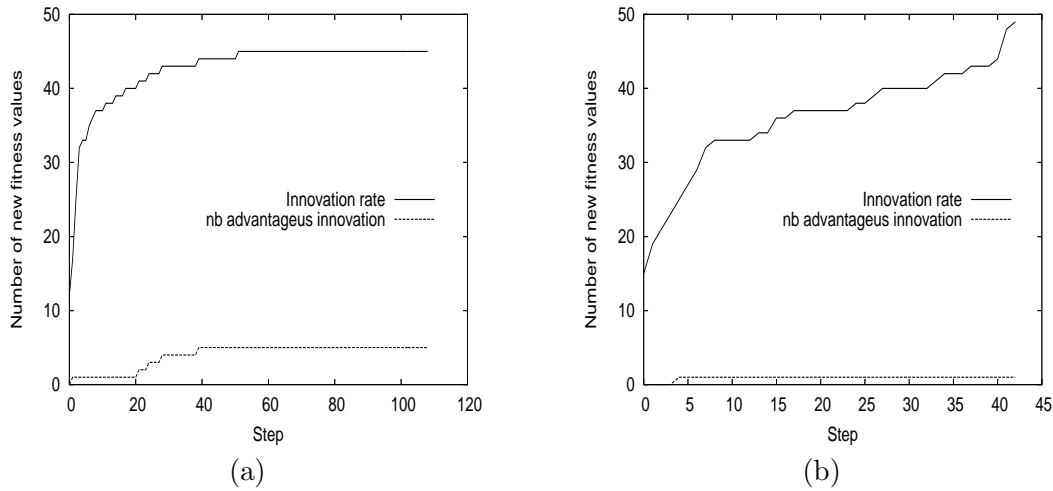


FIG. 4.16 – Innovation cumulative neutre au cours des marches neutres pour  $RN_{0.5}$  (a) et  $RN_{0.76}$  (b).

Les innovations cumulatives neutres pour la plus longue des marches neutres obtenues pour chaque RN sont représentée figure 4.16. La majorité des nouvelles valeurs de performance au cours des marches neutres sont de moins bonne performance et peu sont meilleures.

Cette étude donne une meilleure description de la neutralité du paysage du problème de majorité qui a une importance sur la conception de métaheuristique efficace. Le degré de neutralité est important. L'opérateur de sélection devrait donc prendre en compte le cas où les performance des solutions sont égales. De même le taux de mutation ainsi que la taille de la population devrait être ajustée afin de trouver de meilleure solution hors d'un RN.

### Étude des meilleurs optima locaux connus

Nous venons de montrer qu'il était difficile d'obtenir des informations pertinentes sur le paysage du problème de majorité du fait du nombre important de solution de performance nulle. Dans cette sous-section, nous étudions le paysage par le "haut". Plusieurs auteurs ont trouvé de très bonnes solutions, soit "à la main", soit à l'aide d'un AEs [42, 28, 4, 67]. Nous allons considérer ces optima locaux<sup>24</sup>, noté *moc* pour Meilleurs Optima local Connus, de performance standard supérieure à 0.81 (tab. 4.2). Dans la suite, nous allons analyser la partie du paysage où sont situés les *moc*.

### Répartition spatiale

Dans cette section, nous étudions la répartition spatiale des six *moc*. La table 4.3 donne la distance de Hamming entre ces optima locaux. Toutes les distances sont inférieures à 64 qui est la distance entre deux solutions uniformément choisies dans l'espace de recherche. Les optima locaux ne semblent pas répartis aléatoirement sur le paysage. Certains sont proches ;

<sup>24</sup>Dans la sous-section 4.3.2, nous allons montrer que ceux sont réellement des optima locaux

TAB. 4.2 – Description et performance standard des six meilleures règles connues (*moc*) calculées sur un échantillon de CI de  $10^4$ .

GLK 0.815	00000000 01011111 00000000 01011111 00000000 01011111 00000000 01011111 00000000 01011111 11111111 01011111 00000000 01011111 11111111 01011111
Das 0.823	00000000 00101111 00000011 01011111 00000000 00011111 11001111 00011111 00000000 00101111 11111100 01011111 00000000 00011111 11111111 00011111
Davis 0.818	00000111 00000000 00000111 11111111 00001111 00000000 00001111 11111111 00001111 00000000 00000111 11111111 00001111 00110001 00001111 11111111
ABK 0.824	00000101 00000000 01010101 00000101 00000101 00000000 01010101 00000101 01010101 11111111 01010101 11111111 01010101 11111111 01010101 11111111
Coe1 0.851	00000001 00010100 00110000 11010111 00010001 00001111 00111001 01010111 00000101 10110100 11111111 00010111 11110001 00111101 11111001 01010111
Coe2 0.860	00010100 01010001 00110000 01011100 00000000 01010000 11001110 01011111 00010111 00010001 11111111 01011111 00001111 01010011 11001111 01011111

TAB. 4.3 – Distances entre les six meilleurs optima locaux connus

	GLK	Davis	Das	ABK	Coe1	Coe2	moyenne
GLK	0	20	62	56	39	34	28.6
Davis	20	0	58	56	45	42	33
Das	62	58	0	50	59	44	35.4
ABK	56	56	50	0	51	54	36.6
Coe1	39	45	59	51	0	51	43
Coe2	34	42	44	54	51	0	39

par exemple les règles GLK et Davis, ou GLK et Coe2. En revanche, les règles Das et GLK, ou Coe2 et Das sont très éloignées les unes des autres.

La figure 4.17 représente le centroïde ( $C$ ) des *moc*. L’ordonnée est la fréquence d’apparition du bit de valeur 1 pour chaque bit. La colonne de droite indique le nombre de bits qui ont la même fréquence. Pour six solutions aléatoires du paysage, en moyenne le centroïde est la chaîne  $(0.5)^{128}$  et le nombre de bits qui ont la même fréquence d’apparition de la valeur 1 suit une loi binomiale 2, 12, 30, 40, 30, 12, 2. Pour les six meilleurs optima connus, un grand nombre de bits ont la même valeur (29 au lieu de 4 dans un cas aléatoire) et un faible nombre de bits (22 au lieu de 40 dans le cas aléatoire) sont “indécis” avec une fréquence de 0.5.

Les *moc* ne sont pas réparties aléatoirement sur le paysage. Ils sont dans un sous-espace particulier de dimension 91 défini par le schéma  $S$  suivant :

```
000*0*** 0***** 0**0*** *****1** 000***** 0*0***** ***** *****1*1
0*0***** ***** *****1** ***1*111 ***** ***1***1 *****1 ***1*111
```

On peut donc supposer que les bits communs sont utiles pour obtenir de bonnes solutions. Ainsi, la recherche d’une bonne solution devrait être plus efficace dans le sous-espace défini par le schéma  $S$ . Avant de vérifier cette conjecture, nous allons continuer d’analyser le paysage par “le haut”.

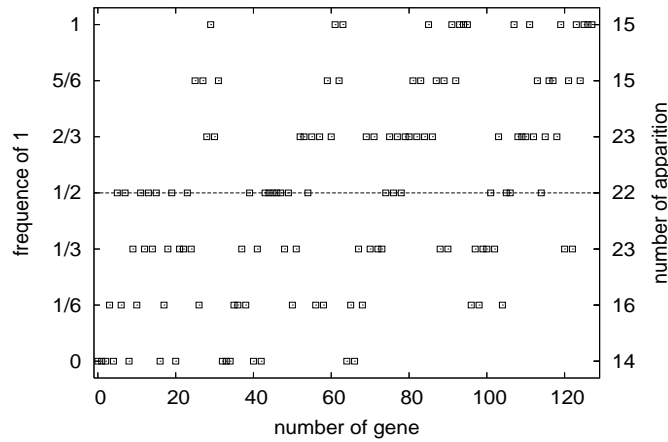


FIG. 4.17 – Centroïde  $C$  de six *moc*. Les carrés indiquent la fréquence de la valeur 1 pour les six *moc* en fonction de la position du bit. La colonne de droite indique le nombre de bit de  $C$  parmi les 128 qui ont la même fréquence de 1 indiquée par la colonne de gauche.

### Profil d'évolvabilité

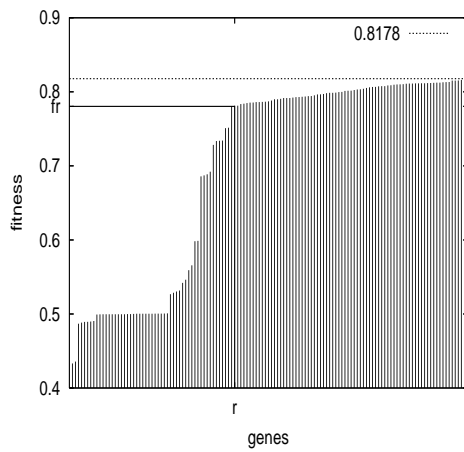
L'évolvabilité a été définie section 1.3.2. Le *profil d'évolvabilité* (PE) a pour but de proposer une représentation de l'évolvabilité d'une solution relativement aux opérateurs de recherche locale. Le PE d'une solution est la performance de toutes ses solutions voisines triées par ordre décroissant. Nous obtenons un profil rang-performance où l'ordonnée est la performance d'une solution voisine et l'abscisse correspond au rang de cette performance parmi toutes les solutions possibles (voir figure 4.18).

La figure 4.18 montre le profil d'évolvabilité des *moc*. Il n'y a pas de solution voisine de meilleure performance que la solution initiale ; toutes les meilleures solutions connues sont donc bien des optima locaux. Le paysage possède deux réseaux de neutralité de performance 0 ( $RN_0$ ) et de performance 0.5 ( $RN_{0.5}$ ) (voir section 4.3.2). Aucun optimum local n'est dans le voisinage de  $RN_0$  ; mais beaucoup de solution voisine des optima locaux (25% environs) appartiennent à  $RN_{0.5}$ . En conséquence, une recherche restreinte au réseau de neutralité  $RN_{0.5}$  peut potentiellement trouver une porte qui mène à l'un des *moc*.

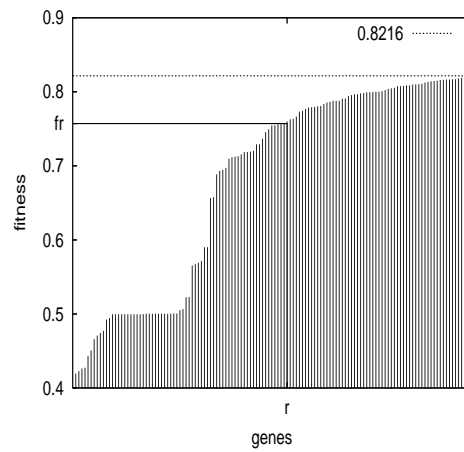
Pour chaque PE, il existe une abscisse  $r$  pour laquelle la performance devient quasi-linéaire en fonction du rang. Soient  $f_r$  cette performance ( $f_{128}$  est la valeur de la moins bonne performance) et  $m$  la pente de la droite entre les abscisse  $r$  et 128. Ainsi, les performances des solutions voisines sont d'autant meilleures que  $m$  et  $r$  sont petits. En revanche, une pente et une abscisse  $r$  grandes signifient que la performance décroît plus vite.

Par exemple, l'évolvabilité est légèrement négative pour la règle GLK, son PE a une faible pente  $m$  et une petite abscisse  $r$ . À l'opposé, pour la règle Coe2, l'PE possède une pente importante ; l'optimum est donc isolé et l'évolvabilité est fortement négative. On peut imaginer "la vue depuis GLK" est plus plate que celle depuis Coe2.

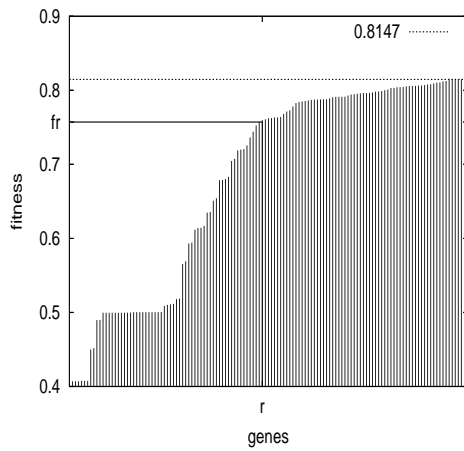
Bien que tous les profils se ressemblent (cf fig. 4.18), on peut se demander le changement de valeur d'un bit modifie la performance de la même manière. Par exemple, pour tous les optima, le changement du premier bit de '0' à '1' provoque une variation importante de la performance. Plus généralement, pour tous les optima, nous avons calculé la moyenne et l'écart-type de la différence de performance lors du changement de valeur d'un bit ; les résultats sont triés par valeur décroissante de la moyenne (voir figure 4.21-a). Les bits qui sont plus délétères (dont la différence provoque la plus grande différence), sont plus souvent ceux qui sont dans le schéma



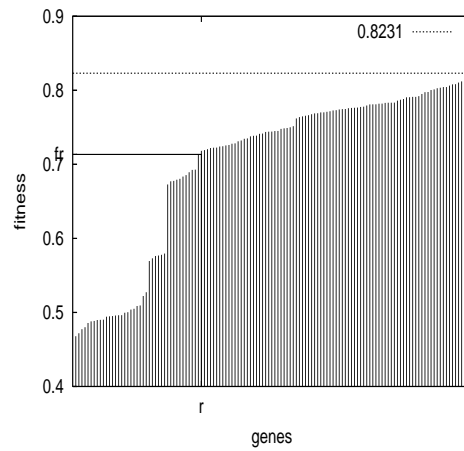
GLK :  $r = 53$ ,  $m = 0.000476$



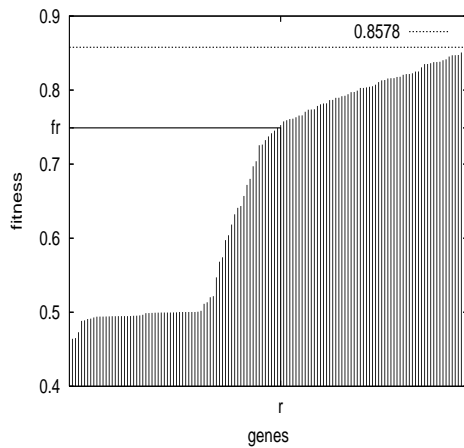
Das :  $r = 69$ ,  $m = 0.00106$



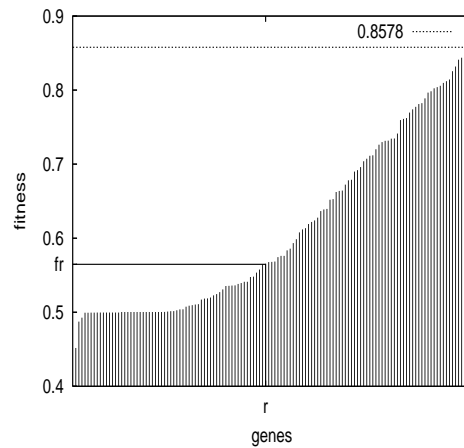
Davis :  $r = 62$ ,  $m = 0.000871$



ABK :  $r = 41$ ,  $m = 0.00114$



Coe1 :  $r = 68$ ,  $m = 0.00170$



Coe2 :  $r = 62$ ,  $m = 0.00424$

FIG. 4.18 – Profil d'évolvabilité pour les six meilleures optima locaux connus. Pour chaque optima, la ligne pointillé indique sa performance. La colonne  $r$  et la pente  $m$  (voir texte) sont reportées sous chaque figure.

*S.* Les bits communs à tous les *moc* semblent important pour trouver de bonnes solutions : pour une métaheuristique, il semble nécessaire de particulariser la recherche au sous-espace défini par

le schéma  $S$ .

### 4.3.3 Olympe

Nous avons mis en évidence les similarités qui existent entre les *moc*. Dans cette section, nous allons utiliser cette caractéristique pour définir l’*Olympe*, un sous-paysage particulier, et montrer et exploiter, les propriétés pertinentes de ce sous-paysage.

#### Définition

L’*Olympe* est un sous-paysage du paysage relatif au problème de majorité. Son nom vient du mont Olympe qui est considéré dans la mythologie grecque comme le lieu de vie des dieux. Avant de définir ce sous-paysage, nous étudions deux symétries naturelles du problème de majorité.

Les états 0 et 1 jouent le même rôle dans le calcul de la tâche ; changer la valeur des bits de toutes les entrées et sortie d’une règle n’a pas d’effet sur la performance. De même, un AC calcule la tâche de majorité de la même manière par la droite ou par la gauche sans changer les performances. On note  $S_{01}$  et  $S_{rl}$  respectivement les opérateurs de la symétrie 0/1 et de la symétrie *droite/gauche*. Soit  $x = (x_0, \dots, x_{N-1}) \in \{0, 1\}^N$  une solution avec  $N = 2^{2r+1}$ . Le symétrique 0/1 de  $x$  est  $S_{01}(x) = y$  où pour tout  $i$ ,  $y_i = 1 - x_{N-i}$ . Le symétrique *droite/gauche* de  $x$  est  $S_{rl}(x) = y$  où pour tout  $i$ ,  $y_i = x_{\sigma(i)}$  avec  $\sigma(\sum_{j=0}^{N-1} 2^{n_j}) = \sum_{j=0}^{N-1} 2^{N-1-n_j}$ . Ces deux opérateurs sont commutatifs :  $S_{rl}S_{01} = S_{01}S_{rl}$ . Parmi les 128 bits, 16 sont invariants par la symétrie  $S_{rl}$  et aucun par la symétrie  $S_{01}$ . La symétrie introduit de la diversité sans modifier la qualité des solutions ; un AE pourrait donc être amélioré en utilisant les opérateurs  $S_{01}$  et  $S_{rl}$ .

Nous avons montré que certaines valeurs de bit particulière pourraient être nécessaires pour trouver de bonnes solutions (cf sous-section 4.3.2), et ceux-ci sont parmi les 29 bits en commun aux *moc* (cf sous-section 4.3.2). Néanmoins, deux optima parmi les *moc* peuvent être distants alors que leurs symétriques peuvent être proches. L’idée pour définir l’*Olympe* est de choisir pour chaque optima l’un de ses symétrique afin de maximiser le nombre de bits commun. Les règles GLK, Das, Davis et ABK ont seulement deux symétriques puisque leurs symétriques par  $S_{01}$  et  $S_{rl}$  sont égaux. Les règles Coe1 et Coe2 ont quatre symétriques. Il y a donc  $2^4 \cdot 4^2 = 256$  ensembles possibles de symétriques. Parmi ces ensembles, nous avons établi celle qui maximise le nombre de bits en commun obtenu est 51. Cet ensemble “optimal” contient les six *symétriques des meilleurs optima connus (moc’)*, ils sont donnés dans la table 4.4. L’*Olympe* est défini à partir des *moc’* par le schéma  $S'$  dont 51 bits sont fixés :

```
000*0*0* 0****1** 0***00** **0**1** 000***** 0*0**1** ***** 0*0**1*1
0*0***** *****1** 111111** **0**111 ***** 0**1*1*1 11111**1 0*01*111
```

L’*Olympe* est un sous espace de dimension 77. Tous les bits fixés dans le schéma  $S$  (cf section 4.3.2) sont fixés dans le schéma  $S'$  avec la même valeur de bit excepté pour le bit numéro 92.

La table 4.5 donne la distance de Hamming entre les *moc’*. Toutes les distances sont inférieures à celles entre les *moc* (cf table 4.3). La distance moyenne entre les règles des *moc’* est de 29.93 et de 35.93 pour les *moc*.

Le centroïde  $C'$  des *moc’* a moins de bits libres (13) et plus de bits fixés (51) que le centroïde  $C$  (voir figure 4.19). Les distances entre  $C'$  et les *moc’* (voir figure 4.20) sont plus courtes que celles entre  $C$  et les *moc*. Les six *moc’* sont plus concentrés autour de  $C'$ . Notons que les optima locaux Coe1 et Coe2 sont ceux de meilleurs performances et ils sont aussi les plus éloignés de  $C'$  bien que leur distance soit en dessous de 38.5 qui est la distance moyenne entre  $C'$  et une solution aléatoire de l’*Olympe*. Ceci suggère peut-être que la recherche ne doit pas uniquement s’effectuer trop près du centroïde.

La figure 4.21-b montre la moyenne et l’écart-type pour les six *moc’* de la différence d’évolvabilité par bit. La courbe moyenne pour les *moc’* a la même forme que celle pour les *moc*, seuls les écart-types sont différents. La moyenne de l’écart-type est 0.08517 pour les *moc* et 0.08367 pour les *moc’*. Le profil d’évolvabilité est plus homogène pour les *moc’* que pour les *moc*.

TAB. 4.4 – Description des six symétriques des meilleurs optima locaux connus (*moc'*).

GLK' = <i>GLK</i>	00000000 01011111 00000000 01011111 00000000 01011111 00000000 01011111 00000000 01011111 11111111 01011111 00000000 01011111 11111111 01011111
Das' = <i>Das</i>	00000000 00101111 00000011 01011111 00000000 00011111 11001111 00011111 00000000 00101111 11111100 01011111 00000000 00011111 11111111 00011111
Davis' = $S_{01}(Davis)$	00000000 00001111 01110011 00001111 00000000 00011111 11111111 00001111 00000000 00001111 11111111 00001111 00000000 00011111 11111111 00011111
ABK' = $S_{01}(ABK)$	00000000 01010101 00000000 01010101 00000000 01010101 00000000 01010101 01011111 01010101 11111111 01011111 01011111 01010101 11111111 01011111
Coe1' = <i>Coe1</i>	00000001 00010100 00110000 11010111 00010001 00001111 00111001 01010111 00000101 10110100 11111111 00010111 11110001 00111101 11111001 01010111
Coe2' = $S_{ri}(Coe2)$	00010100 01010101 00000000 11001100 00001111 00010100 00000010 00011111 00010111 00010101 11111111 11001111 00001111 00010111 11111111 00011111

TAB. 4.5 – Distances entre les symétriques des meilleurs optima locaux connus (*moc'*)

	GLK'	Davis'	Das'	ABK'	Coe1'	Coe2'	moyenne
GLK'	0	20	26	24	39	34	23.8
Davis'	20	0	14	44	45	42	27.5
Das'	26	14	0	50	43	44	29.5
ABK'	24	44	50	0	39	26	30.5
Coe1'	39	45	43	39	0	49	35.8
Coe2'	34	42	44	26	49	0	32.5



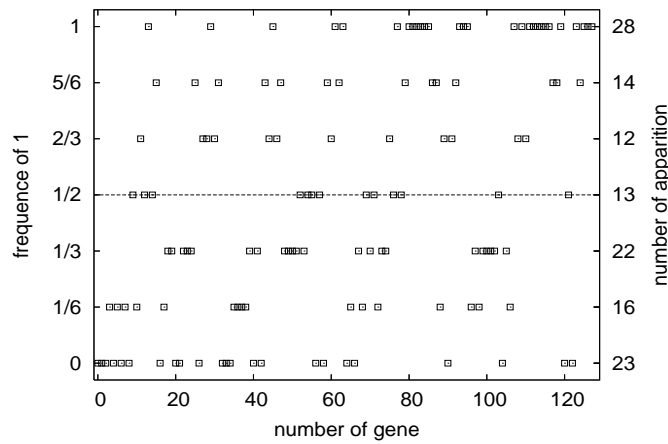


FIG. 4.19 – Centroïde des  $moc'$ . Les carrés indiquent la fréquence de la valeur 1 pour les six  $moc$  en fonction de la position du bit. La colonne de droite indique le nombre de bit de  $C$  parmi les 128 qui ont la même fréquence de 1 indiquée par la colonne de gauche.

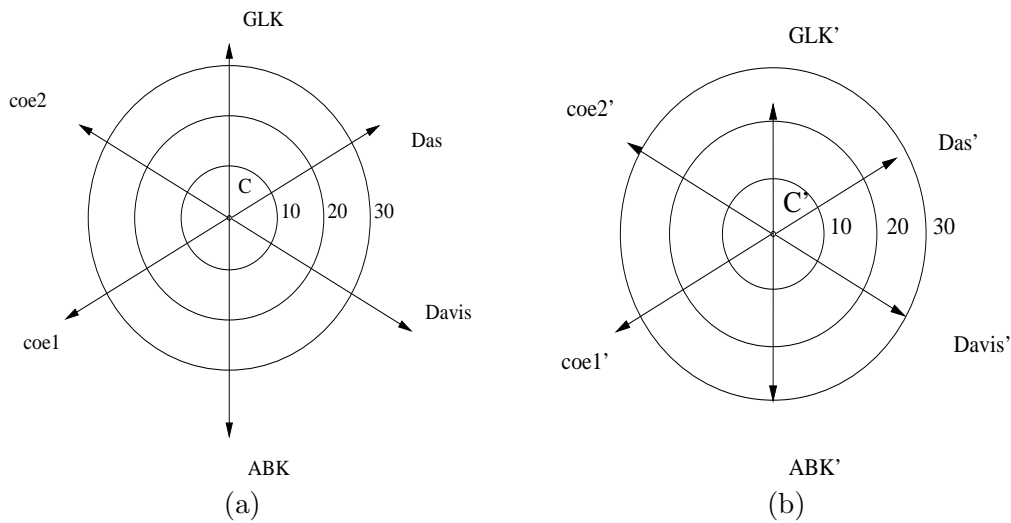


FIG. 4.20 – Distance entre les  $moc$  et le centroïde  $C$  (a) et entre le centroïde  $C'$  et les  $moc'$  (b).

#### 4.3.4 Analyse de l'Olympe

Dans cette section, nous présentons les principales mesures statistiques du paysage restreint à l'Olympe.

##### Densité des états et neutralité

La figure 4.22-a a été obtenue en échantillonnant de manière équiprobable l'Olympe. La D.O.S est plus favorable pour l'Olympe que pour le paysage dans sa globalité (cf figure 4.12-a) bien que la queue de la distribution décroît rapidement pour les performances au-dessus de 0.5.

La figure 4.22-b représente le degré de neutralité de  $10^3$  solutions uniformément choisie dans l'Olympe. Deux larges RN sont situés autour de la performance 0 et 0.5 où le degré de neutralité est supérieur à 80. Le degré de neutralité moyen des solutions est 51.7. En comparaison, cette

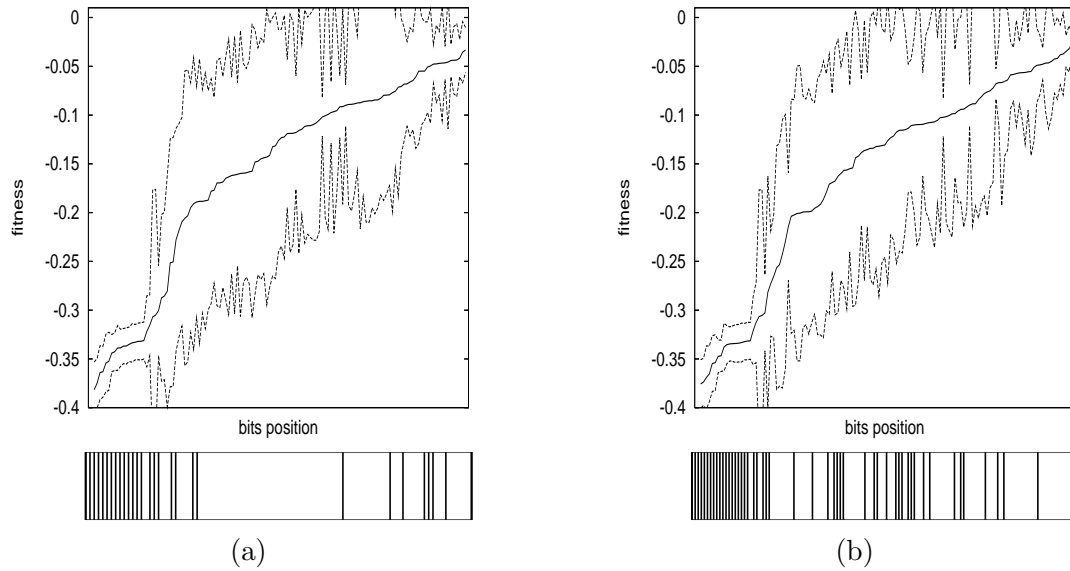


FIG. 4.21 – Moyenne et écart-type de l'évolvabilité par bit pour les *moc* (a) et pour les *moc'* (b). Les barres verticales en dessous des figures indiquent les bits fixés des schéma *S* (a) et *S'* (b).

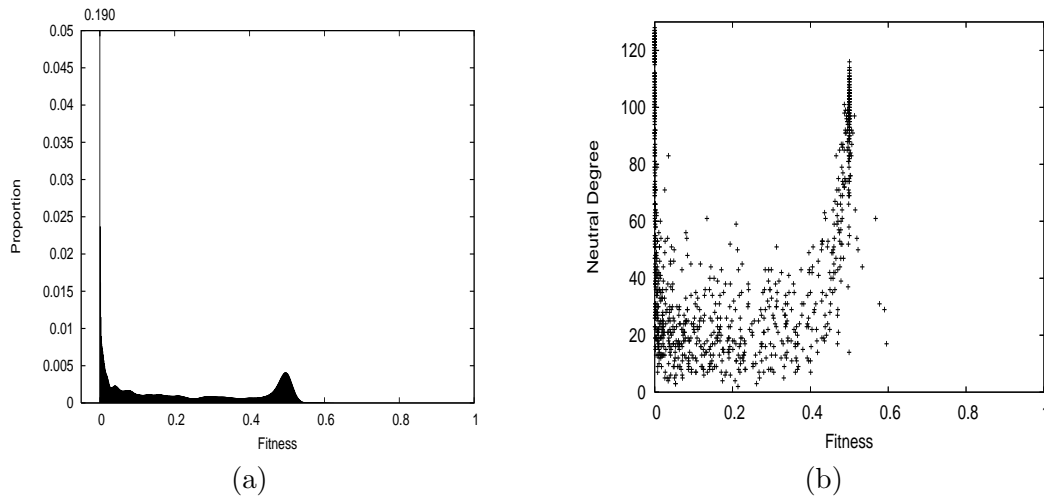


FIG. 4.22 – Densité des états (a) et degré de neutralité des solutions en fonction de leur performance (b) pour l'Olympe.  $10^3$  solutions ont été échantillonnées et ont été évaluées sur un échantillon de CI de taille  $10^4$ .

moyenne est supérieure à celle d'un paysage  $NK_q$  avec  $N = 64$ ,  $K = 2$  et  $q = 2$ , de l'ordre de grandeur d'un paysage  $NK_M$  avec par exemple<sup>25</sup>  $N = 64$ ,  $K = 8$  et  $M = 48$ , ou d'un paysage  $NK_p$   $N = 64$ ,  $K = 4$  et  $p = 0.9$ . Le degré de neutralité moyen de l'Olympe est donc important et devrait être pris en compte dans la conception d'une métaheuristique adaptée à ce problème.

<sup>25</sup>d'autres valeurs de paramètres sont possibles

## Corrélation performance distance

La corrélation performance distance (CPD) a été calculée à partir de  $4.10^3$  solutions appartenant à l'Olympe. Les résultats sont résumés dans la table 4.6. Les six premières lignes de la table donnent les CPD où la distance est calculée à partir de chacun des  $moc'$ . L'avant dernière ligne donne le CPD où la distance est calculée à partir de l'optimum local le plus proche et la dernière ligne donne le CPD où la distance est la distance euclidienne à partir du centroïde  $C'$ . Deux échantillons de solutions sont générés :  $Osample$  où les solutions sont choisies uniformément dans l'Olympe, et  $Csample$  où chaque bit d'une solution a une probabilité d'être 1 égale à la valeur de la coordonnée correspondante du centroïde.

TAB. 4.6 – CPD où la distance est calculée à partir de l'un des  $moc'$ , ou le plus proche des  $moc'$ , ou à partir du centroïde  $C'$ . Deux échantillons de solutions de taille  $10^4$  sont générées :  $Osample$  et  $Csample$ .

	<i>Osample</i>	<i>Csample</i>
GLK'	-0.15609	-0.19399
Davis'	-0.05301	-0.15103
Das'	-0.09202	-0.18476
ABK'	-0.23302	-0.23128
Coe1'	-0.01087	0.077606
Coe2'	-0.11849	-0.17320
“plus proche”	-0.16376	-0.20798
$C'$	-0.23446	-0.33612

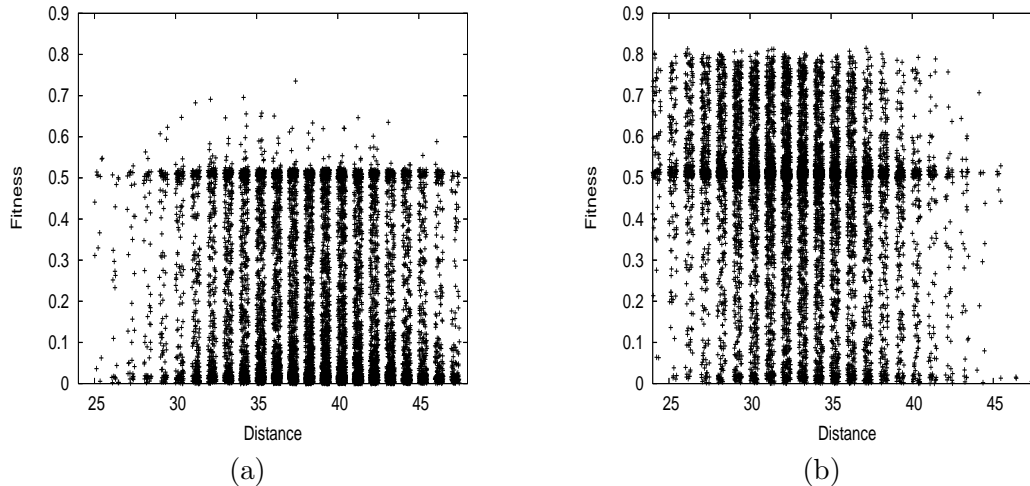


FIG. 4.23 – Nuage de point du CPD calculé avec la distance euclidienne du centroïde  $C'$ . Deux échantillons de solutions de taille  $10^4$  sont générés :  $Osample$  (a) et  $Csample$  (b).

Pour l'échantillon  $Osample$  basé sur l'Olympe, les CPD sont inférieurs à ceux calculés à partir de l'ensemble de l'espace (cf. section 4.3.2), excepté pour pour  $Coe1'$ , ce qui signifie que l'optimisation est plus facile dans l'Olympe. Pour les règles GLK', ABK', les CPD “plus proche” et  $C'$  sont au delà du seuil  $-0.15$ . Pour l'échantillon biaisé  $Csample$ , Tous les CPD sont inférieurs

que ceux pour *Osample*. De même, excepté pour la règle  $\text{Coe1}'$ , les CPD sont au delà de la limite  $-0.15$ . Cette corrélation montre que la performance fournit une information pertinente pour atteindre les meilleurs optima locaux connus. De plus, comme le CPD relatif au centroïde est important (voir aussi figure 4.23), la performance guide vers le centroïde  $C'$ . Nous pouvons en conclure que sur l'Olympe, la performance guide la recherche vers les  $\text{moc}'$  et leur centroïde  $C'$ .

## Rugosité de l'Olympe

Dans cette section, nous analysons la rugosité de l'Olympe en utilisant la méthode de Box-Jenkins décrite dans la section 1.3.2. Les solutions initiales de chaque marche aléatoire sont choisies uniformément dans l'Olympe. À chaque pas, une solution voisine appartenant à l'Olympe est choisie de manière uniforme et sa performance standard est calculée à partir d'un échantillon de taille  $10^4$ . Les marches aléatoires sont de longueur  $10^4$  et la borne de l'erreur utilisée dans l'approche Box-Jenkins est  $\pm 2/\sqrt{10^4} = 0.02$ .

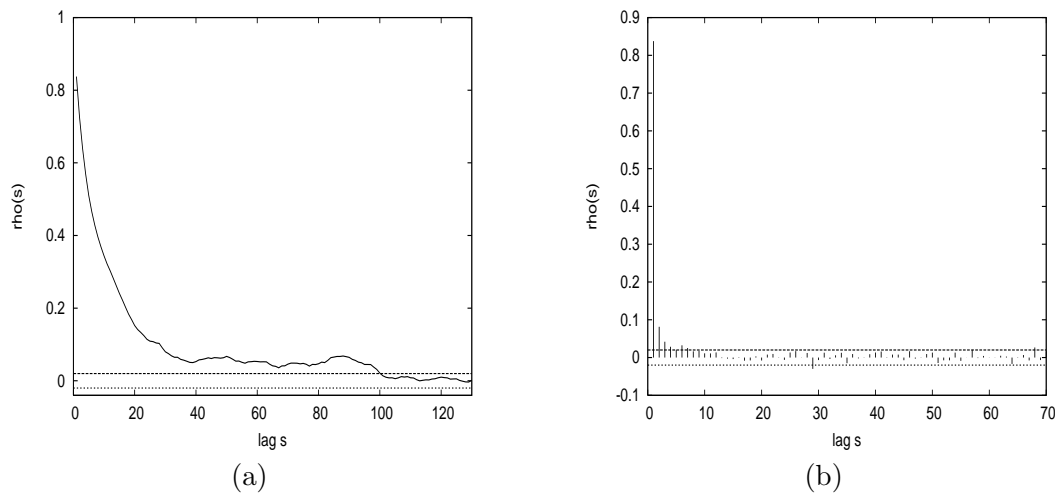


FIG. 4.24 – Fonction d'autocorrélation (a) et d'autocorrélation partielle (b) d'une marche aléatoire sur l'Olympe.

**Identification** La figure 4.24 montre les fonctions d'autocorrélation (acf) en (a) et d'autocorrélation partielle (pacf) en (b). L'acf décroît rapidement. le coefficient d'ordre 1 est égale à 0.838, il est du même ordre de grandeur que celui d'un paysage NK avec  $N = 100$  et  $K = 7$  [58]. L'acf est proche de la borne d'erreur à partir de l'ordre 40 et passe en-dessous à l'ordre 101 qui est la longueur de corrélation. Le coefficient du quatrième ordre de l'autocorrélation partielle est proche de la borne d'erreur. Après l'ordre 4, la pacf est quasiment nulle, ce qui suggère un modèle  $AR(3)$  ou  $AR(4)$ . Le t-test sur l'estimation des coefficients sont significatifs, mais la p-valeur du test Box-Jenkins montre que les résidus ne sont pas des bruits blancs. Nous avons donc essayer un modèle  $ARMA(3, 1)$ . Le dernier coefficient d'autorégression  $\alpha_3$  est quasiment non significatif. Afin de décider de la signifiante de ce coefficient, nous avons extrait la séquence des 980 premier pas et estimé de nouveau le modèle. Le t-test sur  $\alpha_3$  tombe à 0.0738.  $\alpha_3$  est donc non significatif et n'est pas nécessaire au modèle. Finalement, nous avons testé un modèle  $ARMA(2, 1)$ .

**Estimation** Le résultats pour le modèle  $ARMA(2, 1)$  est :

$$y_t = 0.00281 + 1.5384y_{t-1} - 0.5665y_{t-2} + \epsilon_t - 0.7671\epsilon_{t-1}$$

(20.4)                      (32.6)                      (13.7)                      (18.1)

où  $y_t = f(x_t)$ . Les tests statistiques, t-test, de la mesure de signifiacnce sont donnés entre parenthèses en dessous des coefficients : ils sont tous significatifs.

**Diagnostic** Pour l'estimation du modèle  $ARMA(2, 1)$ , le critère d'information d'Akaike (AIC) est  $-16763.63$  et la variance des résidus est  $Var(\epsilon_t) = 0.01094$ . La figure 4.25 montre l'autocorrélation des résidus et les p-valeur des tests Box-Jenkins. L'acf des résidus sont bien inférieure aux bornes d'erreur excepté pour  $h = 28$ . Les résidus ne sont donc pas corrélés. Les p-valeurs du test Box-Jenkins sont bien au dessus de 0.25. Les résidus peuvent donc être considérés comme des bruits blancs. La valeur du R2  $\bar{R}^2 = 0.7050$  est grande et plus grande en comparaison de celle obtenue dans le problème de synchronisation d'un CA [56] où le  $\bar{R}^2$  est égale à 0.38 et 0.35.

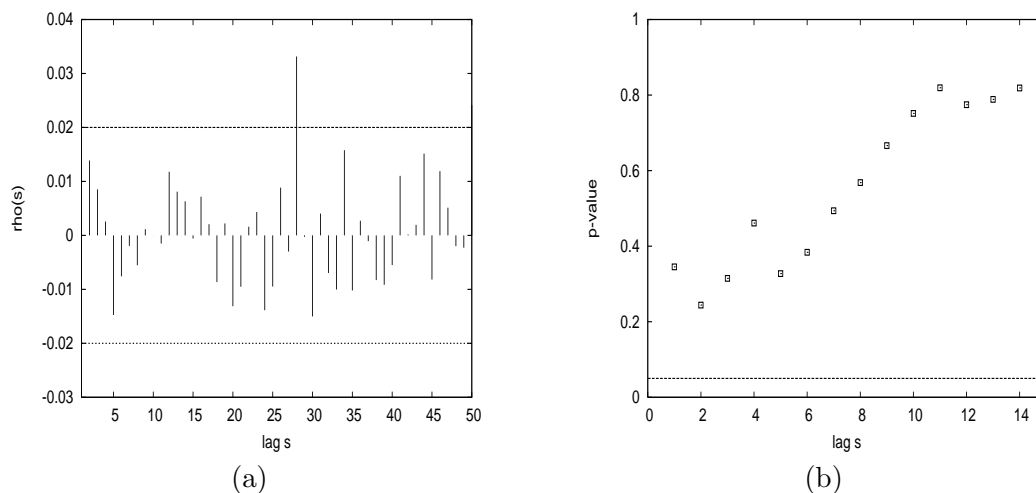


FIG. 4.25 – Fonction d'autocorrélation des résidus (a) et p-valeur de la statistique Ljung-Box (b) pour le modèle  $ARMA(2, 1)$ .

Nous pouvons en conclure que le modèle  $ARMA(2, 1)$  décrit correctement les performances collectées au cours d'une marche aléatoire sur l'Olympe. La forte corrélation montre qu'une métaheuristique de recherche locale peut trouver une bonne règle sur l'Olympe. Un modèle autorégressif d'ordre deux signifie qu'il est nécessaire de connaître les deux dernières performances pour prédire la valeur de la performance suivante. Ainsi, comme suggérer par Hordijk [56], il devrait être possible de construire une métaheuristique de recherche locale en prenant en compte cette information. La composante de moyenne mobile n'a jamais été rencontré dans l'étude d'un paysage adaptatif. Quelle information utile donne-t-elle? Peut-être sur la nature de la neutralité, des travaux futures devons étudier ce genre de modèle plus en détail.

### Nuage adaptatif et CPN

La figure 4.26 montre le nuage de points et les segments utilisés pour calculer le CPN sur l'Olympe (cf section 2.1 et 2.4.2). Aucun segment n'a de pente négative ; il semble facile pour une recherche locale d'atteindre les performances proches de 0.6. Une comparaison avec le nuage

adaptatif de la figure 4.13 relatif à l'ensemble du paysage est intéressante : si tout le paysage est considéré, il est “difficile” de trouver une solution de performance supérieure à 0.5 ; si seulement les solutions appartenant à l'Olympe sont considérés, le problème devient plus facile : il est maintenant plus “facile” de trouver des solutions supérieures à 0.5.

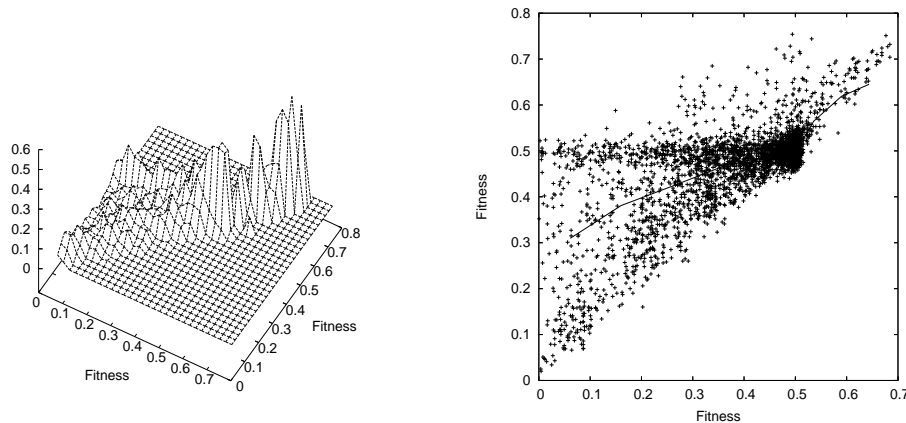


FIG. 4.26 – Nuage adaptatif et segments utilisés pour calculer le CPN sur l'Olympe.

### 4.3.5 Algorithmes évolutionnaires sur l'Olympe

Dans cette section, nous testons différents algorithmes évolutionnaires pour confirmer notre analyse de l'Olympe et trouver de bonnes règles qui résolvent le problème de majorité. Tous les AE sont basés sur l'algorithme génétique simple utilisé par Mitchell *et al.* [87] auquel nous avons ajouté quelques adaptations.

Une population de 200 solutions est utilisée et la performance est la performance standard, *i.e.* la taux de succès sur un échantillon de CI non biaisé. À chaque génération, un nouvel échantillon de taille  $10^3$  est généré. Si une solution reste dans la population pendant  $n$  générations, sa performance est calculée sur un échantillon de taille  $10^3 n$  qui correspond à l'ensemble cumulé de tous les échantillons pendant  $n$  générations. Pour tous les algorithmes, l'initialisation et la mutation sont restreintes à l'Olympe. Afin d'obtenir en moyenne, la mutation d'un bit par solution, la probabilité de mutation par bit est de  $1/77$ . Le croisement un point est utilisé appliqué avec une probabilité de 0.6. Nous utilisons trois versions d'AE :

- l'AE basé sur l'Olympe (AEo) permettant de tester la recherche sur l'Olympe,
- l'AE basé sur le centroïde (AEc) permettant de tester la recherche autour du centroïde,
- l'AE basé sur la neutralité (AEn) exploitant la neutralité considérable de l'Olympe.

**Population initiale** Pour les AE basés sur l'Olympe et la neutralité, la population initiale est uniformément choisie dans le sous-espace défini par l'Olympe. Pour l'AE basé le centroïde, la population initiale est généré à l'aide du centroïde : la probabilité que bit  $i$  soit de valeur 1 est égale à la  $i^{me}$  coordonnée du centroïde  $C'$ . De même, si un bit  $i$  est muté, la nouvelle valeur est générée selon la même probabilité qu'à l'initialisation.

**Opérateur de sélection et de remplacement** l'AEo et AEc utilisent les mêmes opérateurs de sélection et de remplacement que Mitchell *et al.* Les 20% meilleures solutions de la population sont appelées *population d'élites*. L'opérateur de sélection forme une population de

taille égale à 80% de la taille initiale en choisissant uniformément les solutions dans la population d'élites. L'opérateur de remplacement utilise l'élitisme, la population d'élites est intégrée sans modification dans la population de la génération suivante, s'ajoute à celle-ci les 80% de la population auquel a été appliqué l'opérateur de sélection et de variation.

L'AE<sub>n</sub> utilise l'opérateur de sélection par tournoi de taille 2. Il prend en compte la neutralité du paysage : si la performance de deux solutions n'est pas statistiquement différente par un t-test à 95% de confiance, on considère que les performances de deux solutions sont égales et on sélectionne la solution la plus distante du centroïde  $C'$  ; Ce choix permet de disperser la population sur un réseau de neutralité. Dans le cas où le test est positif, l'opérateur sélectionne la solution de meilleure performance. L'AE<sub>n</sub> utilise un remplacement par élitisme de 10% de la population. Ces 10% sont constitués de solutions strictement différentes et sont copiés à la génération suivante sans modification.

**Performances** Chaque AE est exécuté pendant  $10^3$  générations. 50 exécutions indépendantes sont réalisées. Pour chaque exécution, on effectue un post-traitement. À chaque génération, la meilleure solution est testée sur un nouvel échantillon de CI de taille  $10^4$  et la distance entre toutes les solutions est calculée. La meilleure, la moyenne et l'écart-type des performances sont donnés dans la table 4.7. Nous avons aussi calculé le pourcentage d'exécutions capables d'atteindre un niveau de performance donné (voir figure 4.27).

TAB. 4.7 – Performances de AE calculés sur un échantillon de CI de taille  $10^4$ .

AE	Moyenne	Ecart-type	Max
AE <sub>o</sub>	0.8315	0.01928	0.8450
AE <sub>c</sub>	0.8309	0.00575	0.8432
AE <sub>n</sub>	0.8323	0.00556	0.8472

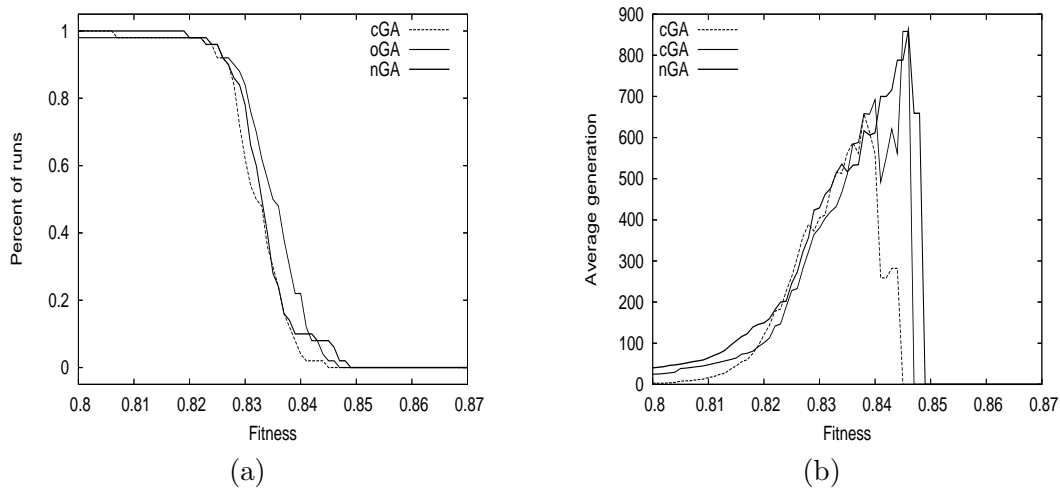


FIG. 4.27 – Pourcentage d'exécutions (a) et nombre de générations nécessaires (b) à l'émergence d'une solution dont la performance est supérieure ou égale au seuil de performance reporté en abscisse.

Tous les AE ont *en moyenne* une meilleure performance que les optima locaux conçus par un humain ou par programmation génétique. Comme attendu, la recherche sur l'Olympe est

très pertinente pour trouver de bonnes règles. Tous les AE ont presque les mêmes performances moyennes. Toutefois, l'écart-type pour l'AE basé sur l'Olympe est quatre fois plus important que celui basé sur le centroïde. Ce qui est confirmé la moyenne des distances entre solutions, l'AEc perd rapidement de la diversité (voir figure 4.28). La probabilité d'obtenir une solution supérieure à 0.835 avec l'AEo est donc plus grande qu'avec l'AEc.

L'AE basé sur la neutralité conserve encore la diversité durant la recherche. La figure 4.27 montre que pour le seuil le plus intéressant de 0.845, l'AEn a plus d'exécutions dépassant ce seuil (3/50) que l'AEo (1/50) ou l'AEc (0/50). Même si nous ne pouvons pas comparer statistiquement les performances maximales pour les différents AEn, l'AE basé sur la neutralité trouve la meilleure solution (0.8472) dont la performance est comparable à la deuxième meilleure performance de l'optimum Coel.

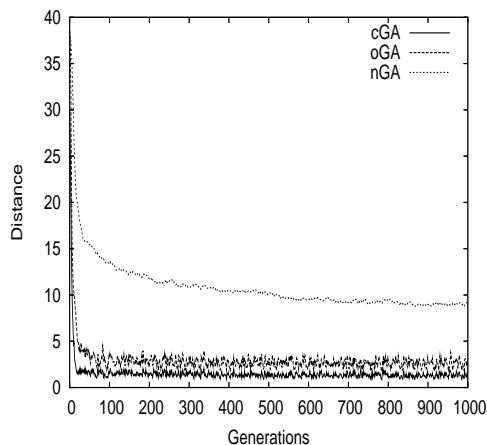


FIG. 4.28 – Distance de Hamming moyenne entre les solutions de la population en fonction des générations.

Ces résultats expérimentaux confirment qu'il est facile de trouver une bonne règle dans l'Olympe. Au cours des 50 exécutions, nous avons trouvé de nombreuses bonnes solutions (différentes) de performances supérieures à 0.82 : 3642 pour l'AEo, 1854 pour l'AEc et 11437 pour l'AEn. Un coût de calcul "moins" important est utilisé pour obtenir ces règles. Une exécution prend environ huit heures sur un PC à 2 GHz à comparer à plusieurs jours pour d'autres algorithmes sur des machines comparables.

Prendre la neutralité en compte permet de maintenir la diversité de la population et accroît la probabilité de trouver des règles de grande performance.

### 4.3.6 Synthèse

Les automates cellulaires sont des calculateurs universaux et leur dynamique peut être complexe et imprédictible. Nous avons étudié l'AC du problème de majorité. Ce problème est un bon exemple d'un phénomène d'émergence dans un système complexe. Nous avons analysé statistiquement le paysage adaptatif relatif à ce problème et mis en avant ses caractéristiques principales afin de discuter des conséquences sur la difficulté d'optimisation de ce problème à l'aide de métaheuristiques. Le degré de neutralité moyen du paysage est important et les graphes de deux grands réseaux de neutralité ne sont pas aléatoires. Beaucoup de solutions sont de performances nulles ou de performance 0.5 ce qui ne permet pas une analyse basée sur l'échantillonnage de l'ensemble du paysage.



Dans un second temps, nous avons étudié le paysage par le “haut” en considérant les meilleurs optima locaux connus. L’exploitation des similarités entre ces optima et des symétries du problème, nous a permis de définir un sous-espace, appelé *Olympe* où l’on peut trouver les six symétriques des meilleurs optima connus (*moc'*), les “dieux” de l’Olympe. Les mesures statiques sur l’Olympe ont montré que :

- les solutions de performance nulle sont moins nombreuses que dans la totalité du paysage,
- la corrélation performance-distance indique que la performance est une information qui guide une recherche locale vers les *moc'* et leur centroïde,
- un modèle *ARMA(2,1)* peut décrire la structure de corrélation de performance entre solutions voisines et montre qu’une recherche locale peut trouver de bonnes solutions.
- le nuage adaptatif et le CPN montrent qu’il est “facile” de trouver à moindre coût des solutions de performance supérieure à 0.5.

Toutes ces mesures indiquent qu’il est plus favorable de rechercher une bonne solution dans l’Olympe, ce que nous avons pu confirmer à l’aide d’algorithmes évolutionnaires qui ont permis de trouver de nombreuses bonnes solutions de performances équivalente aux meilleures solutions trouvées jusqu’à présent.

## 4.4 Synthèse du chapitre

Nous avons décrit la dynamique des équilibres ponctués qui fût initialement découverte en biologie de l’évolution moléculaire, puis adaptée au domaine de l’optimisation combinatoire. Dans le cadre de l’optimisation par algorithme évolutionnaire (AE), les équilibres ponctués sur les paysages adaptatifs neutres correspondent à l’alternance de deux phases : une longue période de dérive aléatoire sur un RN suivi de la découverte rapide d’une porte. Lors de la dérive, il a été montré que la population convergeait vers les régions du RN où les degrés de neutralité sont les plus importants. Nous avons généralisé ce résultat, démontré pour un AE utilisant la mutation et une sélection proportionnelle l’adaptation, au cas de la sélection par tournoi plus largement utilisé dans le domaine des AE.

Les mesures des paysages adaptatifs neutres ont mis en lumière de nouvelles caractéristiques des réseaux de neutralité : les RN sont structurés et leurs interactions peuvent être utilisées pour guider une recherche. Nous avons donc proposé une nouvelle métaheuristique, adaptée aux paysages neutres utilisant la notion d’évolvabilité pour guider la recherche sur les RN. Cette métaheuristique, appelé *recherche périscopique* (RP), est supporté par la métaphore de la nage avec périscope. Elle consiste en l’itération de deux étapes, la première optimise une mesure d’évolvabilité sur un réseau de neutralité, la suivante réalise un saut qualitatif de performance en sélectionnant une solution voisine adéquate.

Les résultats expérimentaux sur les variantes neutres des paysages NK confirment la qualité des performances de la RP et surtout la corrélation entre ses performances et les mesures réalisées au chapitre 3. Pour les paysages  $NK_M$ , la structure des RN est plus faible, et la recherche périscopique obtient de moins bonnes performances. En revanche, lorsque les RN sont plus structurés, comme dans les paysages  $NK_q$  et  $NK_p$ , la recherche périscopique est plus performante.

Le problème de majorité est un problème d’apprentissage difficile de la tâche de classification réalisée par un automate cellulaire. Le paysage adaptatif relatif à ce problème est massivement neutre : cela est dû en partie à l’erreur d’évaluation de la performance d’une règle. Nous avons étudié ce paysage dans son ensemble et montré que le nombre important de solutions de performance nulle ne permet pas une étude statistique. Ensuite, nous avons étudié ce paysage par le “haut” en considérant les meilleurs optima locaux à ce jour connus. L’exploitation des similarités entre ces différents optima et des symétries du problème, a permis de définir un sous-espace, ap-

pelé *Olympe*, à partir des six symétriques des meilleurs optima connus. L'étude de ce sous-espace et l'analyse de ses RN a permis de montrer que l'optimisation par l'algorithme évolutionnaire dans celui-ci et en considérant la neutralité du paysage est plus facile et de trouver des solutions à moindre coût de qualité équivalente aux meilleurs.



# Conclusion

La métaphore des paysages adaptatifs supporte un des concepts les plus pertinents pour modéliser des systèmes dynamiques complexes. Elle s'est imposée dans de nombreux domaines scientifiques tels que la biologie moléculaire, la physique statistique, ou l'optimisation combinatoire. Dans le domaine de l'optimisation combinatoire, l'intérêt même de la métaphore des paysages est de coupler description géométrique et dynamique de recherche. Nous avons présenté les géométries connues depuis les années 1930 relatives aux optima locaux ; elles permettent de décrire par exemple le paysage à l'aide de marches adaptatives et de l'autocorrélation des performances au cours d'une marche aléatoire.

Les paysages adaptatifs neutres, issus du domaine de l'évolution moléculaire, ajoutent aux paysages l'image des "plateaux". Dans cette thèse, nous avons utilisé les notions d'*ensemble de neutralité* et de *réseau de neutralité* pour modéliser ce type de paysage. Nous avons tout d'abord proposé un nouvel outil d'étude des ensembles de neutralité, le *nuage adaptatif* (NA), qui permet d'analyser l'effet d'un opérateur local sur ces ensembles. L'expression analytique du NA relativement à l'opérateur de recherche aléatoire a été donné pour une large famille de paysages, les paysages *embarqués uniformes*. Nous avons montré que le NA permet dans une certaine mesure de prédire l'évolution des performances lors de l'itération d'un opérateur local. Enfin, nous avons présenté une mesure de difficulté déduite du NA, le *Coefficient de Pente Négative*, particulièrement adaptée au domaine de la Programmation Génétique. Les "plateaux" d'un paysage adaptatif neutre sont l'image des réseaux de neutralité. Nous avons utilisé des outils issus de différentes disciplines, comme l'autocorrélation des degrés du graphe, les profils rang-taille des réseaux, le taux d'innovation,..., pour analyser ces réseaux sur différents paysages. Nous avons défini une nouvelle mesure, l'*autocorrélation de l'évolvabilité*, pour compléter la description des relations inter réseaux. Selon le point de vue de la neutralité, nous avons étudié une grande variété de paysages choisis de manière à couvrir un large panel de possibilités : le paysage MAX-SAT, différentes familles de paysages additifs, et le paysage des *Routes Épistatiques* que nous avons défini. L'ensemble des mesures effectuées pourra maintenant servir de référence lors de l'étude de nouveaux paysages. Cette étude montre également que la représentation classique d'un paysage à l'aide de "plateaux" regroupant des points de même performance est trop réductrice. La description par réseaux de neutralité et d'interconnexion entre réseaux est plus pertinente. À l'aide de cette représentation, nous avons pu concevoir une nouvelle métaheuristique, la *Recherche Périscopique*. Ses performances ont été "mesurées" sur plusieurs familles de paysages additifs. Nous avons observé que la recherche périscopique obtenait de meilleures performances lorsque les réseaux de neutralité étaient structurés. Ce résultat est en accord avec les caractéristiques des paysages étudiés. L'analyse d'un paysage massivement neutre relatif au *Problème de Majorité*, qui est un problème difficile d'apprentissage d'une tâche par un automate cellulaire, a permis de définir un sous-espace sur lequel l'optimisation est facilitée. Cela a été confirmé par l'optimisation de ce problème par un algorithme évolutionnaire prenant en compte la neutralité du paysage.

Nous pouvons envisager deux principales directions pour de futurs travaux, l'une portant

sur l'étude des paysages neutres et leurs rapports aux algorithmes d'optimisation, et l'autre sur la conception de nouvelles techniques d'analyse des paysages neutres.

Dans cette thèse, nous avons proposé et testé de nombreux outils d'analyse des paysages, ils permettront dans l'avenir d'analyser des paysages neutres issu de problèmes plus applicatifs. Ce type d'analyse permettra de déduire les caractéristiques pertinentes d'un paysage neutre afin de lui appliquer une métaheuristique adaptée. De nombreux travaux suggèrent que les paysages issus de la Programmation Génétique sont de nature neutre. Les études sur ces paysages pourront être complétées à l'aide des premiers résultats sur le paysage des Routes Épistatiques. En optimisation combinatoire un problème peut conduire à différents paysages selon par exemple le choix du codage des solutions ; se pose alors la question de savoir si, pour un problème donné, le degré de neutralité d'un paysage est corrélé à la difficulté de la recherche. Dans ce travail, nous n'avons pas établi de lien direct entre difficulté et neutralité, toutefois des éléments de réponses ont été apportés. Nous avons montré que la recherche périscopique pouvait être performante sur certains paysages neutres et non sur d'autres. Des travaux futurs pourront donc s'intéresser aux rapports entre la difficulté à optimiser relative à une métaheuristique donnée et un ensemble de mesures sur les paysages neutres.

Les outils que nous avons utilisées et développées mesurent un paysage de façon "statique" en échantillonnant un certain nombre de solutions suivant différentes méthodes. Nous avons principalement utilisé des méthodes statistiques pour décrire les paysages neutres ; des outils manquent pour "mesurer" les paysages neutres au cours de l'optimisation, cela permettrait d'adapter les métaheurstiques, soit *a posteriori*, soit pendant la recherche elle-même. L'essor des réseaux dans diverses disciplines scientifiques apporte de nouvelles méthodes de visualisation des grands graphes. Nous pourrions utiliser et développer ces méthodes afin d'améliorer notre perception des paysages neutres et peut-être ainsi d'élaborer de nouvelles techniques d'optimisation.

## Annexe A

# Résultats expérimentaux des métaheuristiques

TAB. A.1 – Moyenne et écart-type des performances des différents algorithmes sur les paysages  $NK_q$  pour  $N = 16$ .

$K$	$q$	RN	RP	HC	HC2	EA	SE
1	2	0.820 <sub>0.075</sub>	0.821 <sub>0.078</sub>	0.789 <sub>0.083</sub>	0.817 <sub>0.078</sub>	<b>0.830<sub>0.073</sub></b>	0.828 <sub>0.073</sub>
	3	0.771 <sub>0.065</sub>	0.778 <sub>0.064</sub>	0.758 <sub>0.069</sub>	0.775 <sub>0.067</sub>	<b>0.784<sub>0.062</sub></b>	0.781 <sub>0.061</sub>
	4	0.755 <sub>0.063</sub>	0.763 <sub>0.063</sub>	0.750 <sub>0.065</sub>	0.762 <sub>0.066</sub>	<b>0.769<sub>0.060</sub></b>	0.765 <sub>0.060</sub>
	10	0.719 <sub>0.052</sub>	0.728 <sub>0.053</sub>	0.723 <sub>0.051</sub>	0.729 <sub>0.054</sub>	<b>0.734<sub>0.049</sub></b>	0.731 <sub>0.050</sub>
2	2	0.863 <sub>0.065</sub>	0.861 <sub>0.069</sub>	0.814 <sub>0.082</sub>	0.853 <sub>0.077</sub>	<b>0.886<sub>0.054</sub></b>	0.882 <sub>0.054</sub>
	3	0.802 <sub>0.066</sub>	0.810 <sub>0.065</sub>	0.777 <sub>0.072</sub>	0.801 <sub>0.076</sub>	<b>0.826<sub>0.056</sub></b>	0.821 <sub>0.055</sub>
	4	0.774 <sub>0.064</sub>	0.786 <sub>0.061</sub>	0.763 <sub>0.066</sub>	0.783 <sub>0.069</sub>	<b>0.803<sub>0.054</sub></b>	0.796 <sub>0.053</sub>
	10	0.733 <sub>0.056</sub>	0.743 <sub>0.054</sub>	0.735 <sub>0.056</sub>	0.745 <sub>0.059</sub>	<b>0.761<sub>0.046</sub></b>	0.756 <sub>0.045</sub>
3	2	0.874 <sub>0.064</sub>	0.874 <sub>0.067</sub>	0.822 <sub>0.082</sub>	0.864 <sub>0.077</sub>	<b>0.900<sub>0.054</sub></b>	0.897 <sub>0.053</sub>
	3	0.807 <sub>0.057</sub>	0.815 <sub>0.057</sub>	0.781 <sub>0.066</sub>	0.806 <sub>0.069</sub>	<b>0.836<sub>0.048</sub></b>	0.831 <sub>0.048</sub>
	4	0.772 <sub>0.057</sub>	0.786 <sub>0.053</sub>	0.763 <sub>0.059</sub>	0.782 <sub>0.063</sub>	<b>0.806<sub>0.045</sub></b>	0.800 <sub>0.046</sub>
	10	0.733 <sub>0.050</sub>	0.745 <sub>0.048</sub>	0.734 <sub>0.051</sub>	0.746 <sub>0.056</sub>	<b>0.767<sub>0.039</sub></b>	0.763 <sub>0.039</sub>
5	2	0.864 <sub>0.061</sub>	0.870 <sub>0.063</sub>	0.822 <sub>0.075</sub>	0.866 <sub>0.073</sub>	<b>0.901<sub>0.052</sub></b>	0.896 <sub>0.051</sub>
	3	0.798 <sub>0.053</sub>	0.808 <sub>0.053</sub>	0.779 <sub>0.062</sub>	0.805 <sub>0.064</sub>	<b>0.834<sub>0.044</sub></b>	0.831 <sub>0.044</sub>
	4	0.767 <sub>0.053</sub>	0.779 <sub>0.049</sub>	0.757 <sub>0.056</sub>	0.784 <sub>0.059</sub>	<b>0.805<sub>0.040</sub></b>	0.798 <sub>0.040</sub>
	10	0.726 <sub>0.046</sub>	0.736 <sub>0.044</sub>	0.727 <sub>0.047</sub>	0.745 <sub>0.052</sub>	<b>0.763<sub>0.035</sub></b>	0.758 <sub>0.035</sub>
8	2	0.832 <sub>0.059</sub>	0.844 <sub>0.058</sub>	0.802 <sub>0.070</sub>	0.857 <sub>0.060</sub>	<b>0.873<sub>0.047</sub></b>	0.869 <sub>0.047</sub>
	3	0.770 <sub>0.052</sub>	0.783 <sub>0.051</sub>	0.758 <sub>0.057</sub>	0.798 <sub>0.055</sub>	<b>0.810<sub>0.042</sub></b>	0.806 <sub>0.041</sub>
	4	0.741 <sub>0.051</sub>	0.757 <sub>0.047</sub>	0.740 <sub>0.052</sub>	0.776 <sub>0.053</sub>	<b>0.785<sub>0.040</sub></b>	0.780 <sub>0.040</sub>
	10	0.704 <sub>0.045</sub>	0.717 <sub>0.042</sub>	0.709 <sub>0.044</sub>	0.737 <sub>0.047</sub>	<b>0.744<sub>0.034</sub></b>	0.740 <sub>0.034</sub>

TAB. A.2 – Moyenne et écart-type des performances des différents algorithmes sur les paysages  $NK_q$  pour  $N = 64$ .

$K$	$q$	RN	RP	HC	HC2	EA	SE
0	2	0.734 <sub>0.049</sub>	<b>0.736</b> <sub>0.049</sub>	<b>0.736</b> <sub>0.049</sub>	<b>0.736</b> <sub>0.049</sub>	0.735 <sub>0.048</sub>	0.732 <sub>0.044</sub>
	3	0.708 <sub>0.041</sub>	<b>0.709</b> <sub>0.040</sub>	<b>0.709</b> <sub>0.040</sub>	<b>0.709</b> <sub>0.040</sub>	0.702 <sub>0.039</sub>	0.694 <sub>0.040</sub>
	4	0.697 <sub>0.036</sub>	<b>0.698</b> <sub>0.036</sub>	<b>0.698</b> <sub>0.036</sub>	<b>0.698</b> <sub>0.036</sub>	0.685 <sub>0.035</sub>	0.676 <sub>0.036</sub>
	10	0.672 <sub>0.030</sub>	<b>0.673</b> <sub>0.030</sub>	<b>0.673</b> <sub>0.030</sub>	<b>0.673</b> <sub>0.030</sub>	0.659 <sub>0.030</sub>	0.649 <sub>0.030</sub>
2	2	0.862 <sub>0.041</sub>	<b>0.871</b> <sub>0.042</sub>	0.826 <sub>0.048</sub>	0.854 <sub>0.058</sub>	0.847 <sub>0.039</sub>	0.835 <sub>0.038</sub>
	3	0.802 <sub>0.037</sub>	<b>0.817</b> <sub>0.036</sub>	0.787 <sub>0.040</sub>	0.801 <sub>0.047</sub>	0.789 <sub>0.033</sub>	0.780 <sub>0.031</sub>
	4	0.769 <sub>0.030</sub>	<b>0.787</b> <sub>0.035</sub>	0.765 <sub>0.034</sub>	0.776 <sub>0.040</sub>	0.761 <sub>0.027</sub>	0.755 <sub>0.027</sub>
	10	0.729 <sub>0.027</sub>	<b>0.745</b> <sub>0.028</sub>	0.736 <sub>0.028</sub>	0.743 <sub>0.034</sub>	0.727 <sub>0.023</sub>	0.719 <sub>0.022</sub>
4	2	0.869 <sub>0.030</sub>	<b>0.880</b> <sub>0.031</sub>	0.832 <sub>0.040</sub>	0.855 <sub>0.052</sub>	0.857 <sub>0.027</sub>	0.847 <sub>0.027</sub>
	3	0.804 <sub>0.027</sub>	<b>0.822</b> <sub>0.027</sub>	0.788 <sub>0.031</sub>	0.797 <sub>0.044</sub>	0.798 <sub>0.023</sub>	0.791 <sub>0.022</sub>
	4	0.773 <sub>0.027</sub>	<b>0.793</b> <sub>0.026</sub>	0.768 <sub>0.029</sub>	0.775 <sub>0.039</sub>	0.770 <sub>0.022</sub>	0.766 <sub>0.020</sub>
	10	0.733 <sub>0.023</sub>	<b>0.749</b> <sub>0.023</sub>	0.739 <sub>0.024</sub>	0.742 <sub>0.031</sub>	0.733 <sub>0.019</sub>	0.729 <sub>0.018</sub>
8	2	0.849 <sub>0.030</sub>	<b>0.863</b> <sub>0.031</sub>	0.821 <sub>0.038</sub>	0.831 <sub>0.055</sub>	0.847 <sub>0.026</sub>	0.840 <sub>0.025</sub>
	3	0.787 <sub>0.026</sub>	<b>0.804</b> <sub>0.027</sub>	0.775 <sub>0.031</sub>	0.774 <sub>0.048</sub>	0.786 <sub>0.020</sub>	0.782 <sub>0.020</sub>
	4	0.760 <sub>0.025</sub>	<b>0.777</b> <sub>0.024</sub>	0.755 <sub>0.028</sub>	0.759 <sub>0.040</sub>	0.762 <sub>0.021</sub>	0.758 <sub>0.019</sub>
	10	0.721 <sub>0.023</sub>	<b>0.733</b> <sub>0.023</sub>	0.725 <sub>0.024</sub>	0.725 <sub>0.034</sub>	0.726 <sub>0.017</sub>	0.723 <sub>0.016</sub>
12	2	0.825 <sub>0.028</sub>	<b>0.838</b> <sub>0.029</sub>	0.803 <sub>0.036</sub>	0.809 <sub>0.057</sub>	0.826 <sub>0.026</sub>	0.824 <sub>0.024</sub>
	3	0.767 <sub>0.026</sub>	<b>0.781</b> <sub>0.025</sub>	0.758 <sub>0.031</sub>	0.759 <sub>0.045</sub>	0.771 <sub>0.022</sub>	0.769 <sub>0.020</sub>
	4	0.741 <sub>0.024</sub>	<b>0.756</b> <sub>0.025</sub>	0.738 <sub>0.027</sub>	0.742 <sub>0.042</sub>	0.748 <sub>0.020</sub>	0.748 <sub>0.019</sub>
	10	0.705 <sub>0.021</sub>	<b>0.717</b> <sub>0.022</sub>	0.709 <sub>0.024</sub>	0.711 <sub>0.033</sub>	0.715 <sub>0.017</sub>	0.713 <sub>0.016</sub>
16	2	0.803 <sub>0.028</sub>	<b>0.815</b> <sub>0.029</sub>	0.785 <sub>0.035</sub>	0.793 <sub>0.053</sub>	0.808 <sub>0.025</sub>	0.807 <sub>0.024</sub>
	3	0.748 <sub>0.025</sub>	<b>0.760</b> <sub>0.026</sub>	0.740 <sub>0.028</sub>	0.743 <sub>0.045</sub>	0.756 <sub>0.022</sub>	0.754 <sub>0.019</sub>
	4	0.724 <sub>0.024</sub>	<b>0.737</b> <sub>0.024</sub>	0.723 <sub>0.026</sub>	0.729 <sub>0.040</sub>	0.735 <sub>0.020</sub>	0.733 <sub>0.019</sub>
	10	0.691 <sub>0.021</sub>	0.700 <sub>0.021</sub>	0.694 <sub>0.022</sub>	0.698 <sub>0.033</sub>	0.701 <sub>0.017</sub>	<b>0.701</b> <sub>0.017</sub>



TAB. A.3 – Moyenne et écart-type des performances des différents algorithmes sur les paysages  $NK_M$  pour  $N = 16$ .

$K$	$M$	RN	RP	HC	HC2	EA	SE
1	16	0.668 <sub>0.050</sub>	0.675 <sub>0.050</sub>	0.675 <sub>0.050</sub>	<b>0.678</b> <sub>0.051</sub>	0.674 <sub>0.050</sub>	0.664 <sub>0.052</sub>
	32	0.683 <sub>0.048</sub>	0.691 <sub>0.047</sub>	0.691 <sub>0.047</sub>	0.694 <sub>0.049</sub>	<b>0.698</b> <sub>0.046</sub>	0.694 <sub>0.047</sub>
	48	0.689 <sub>0.047</sub>	0.696 <sub>0.047</sub>	0.696 <sub>0.047</sub>	0.699 <sub>0.049</sub>	<b>0.703</b> <sub>0.045</sub>	0.700 <sub>0.045</sub>
	160	0.696 <sub>0.047</sub>	0.703 <sub>0.047</sub>	0.703 <sub>0.047</sub>	0.706 <sub>0.048</sub>	<b>0.712</b> <sub>0.045</sub>	0.708 <sub>0.045</sub>
2	16	0.679 <sub>0.054</sub>	0.685 <sub>0.053</sub>	0.685 <sub>0.053</sub>	0.689 <sub>0.059</sub>	<b>0.702</b> <sub>0.045</sub>	0.694 <sub>0.046</sub>
	32	0.694 <sub>0.051</sub>	0.701 <sub>0.050</sub>	0.701 <sub>0.050</sub>	0.704 <sub>0.057</sub>	<b>0.722</b> <sub>0.042</sub>	0.717 <sub>0.041</sub>
	48	0.699 <sub>0.051</sub>	0.706 <sub>0.050</sub>	0.706 <sub>0.050</sub>	0.710 <sub>0.056</sub>	<b>0.726</b> <sub>0.042</sub>	0.722 <sub>0.041</sub>
	160	0.707 <sub>0.051</sub>	0.713 <sub>0.050</sub>	0.713 <sub>0.050</sub>	0.717 <sub>0.056</sub>	<b>0.734</b> <sub>0.042</sub>	0.730 <sub>0.041</sub>
3	16	0.678 <sub>0.049</sub>	0.684 <sub>0.048</sub>	0.684 <sub>0.048</sub>	0.690 <sub>0.055</sub>	<b>0.708</b> <sub>0.040</sub>	0.704 <sub>0.041</sub>
	32	0.693 <sub>0.047</sub>	0.699 <sub>0.046</sub>	0.699 <sub>0.046</sub>	0.706 <sub>0.053</sub>	<b>0.726</b> <sub>0.037</sub>	0.723 <sub>0.037</sub>
	48	0.698 <sub>0.046</sub>	0.705 <sub>0.045</sub>	0.705 <sub>0.045</sub>	0.711 <sub>0.053</sub>	<b>0.732</b> <sub>0.037</sub>	0.727 <sub>0.036</sub>
	160	0.706 <sub>0.046</sub>	0.712 <sub>0.045</sub>	0.712 <sub>0.045</sub>	0.719 <sub>0.052</sub>	<b>0.739</b> <sub>0.036</sub>	0.735 <sub>0.036</sub>
5	16	0.672 <sub>0.046</sub>	0.677 <sub>0.044</sub>	0.677 <sub>0.044</sub>	0.687 <sub>0.053</sub>	<b>0.708</b> <sub>0.035</sub>	0.705 <sub>0.035</sub>
	32	0.688 <sub>0.043</sub>	0.693 <sub>0.042</sub>	0.693 <sub>0.042</sub>	0.703 <sub>0.050</sub>	<b>0.724</b> <sub>0.032</sub>	0.720 <sub>0.032</sub>
	48	0.693 <sub>0.042</sub>	0.698 <sub>0.041</sub>	0.698 <sub>0.041</sub>	0.708 <sub>0.050</sub>	<b>0.729</b> <sub>0.032</sub>	0.725 <sub>0.032</sub>
	160	0.700 <sub>0.042</sub>	0.705 <sub>0.041</sub>	0.705 <sub>0.041</sub>	0.715 <sub>0.050</sub>	<b>0.735</b> <sub>0.032</sub>	0.732 <sub>0.031</sub>
8	16	0.653 <sub>0.044</sub>	0.662 <sub>0.043</sub>	0.661 <sub>0.043</sub>	0.681 <sub>0.048</sub>	<b>0.695</b> <sub>0.031</sub>	0.692 <sub>0.032</sub>
	32	0.668 <sub>0.041</sub>	0.677 <sub>0.040</sub>	0.677 <sub>0.040</sub>	0.697 <sub>0.045</sub>	<b>0.707</b> <sub>0.031</sub>	0.704 <sub>0.031</sub>
	48	0.674 <sub>0.041</sub>	0.682 <sub>0.040</sub>	0.682 <sub>0.040</sub>	0.702 <sub>0.045</sub>	<b>0.712</b> <sub>0.031</sub>	0.709 <sub>0.030</sub>
	160	0.681 <sub>0.040</sub>	0.690 <sub>0.039</sub>	0.689 <sub>0.039</sub>	0.709 <sub>0.045</sub>	<b>0.718</b> <sub>0.031</sub>	0.715 <sub>0.030</sub>

TAB. A.4 – Moyenne et écart-type des performances des différents algorithmes sur les paysages  $NK_M$  pour  $N = 64$ .

$K$	$M$	RN	RP	HC	HC2	EA	SE
0	16	0.625 <sub>0.034</sub>	0.494 <sub>0.037</sub>	<b>0.627<sub>0.033</sub></b>	0.495 <sub>0.039</sub>	0.549 <sub>0.032</sub>	0.536 <sub>0.032</sub>
	32	0.641 <sub>0.029</sub>	0.514 <sub>0.035</sub>	<b>0.642<sub>0.029</sub></b>	0.513 <sub>0.036</sub>	0.593 <sub>0.029</sub>	0.580 <sub>0.033</sub>
	48	0.646 <sub>0.028</sub>	0.556 <sub>0.037</sub>	<b>0.648<sub>0.028</sub></b>	0.554 <sub>0.038</sub>	0.611 <sub>0.029</sub>	0.610 <sub>0.031</sub>
	160	0.654 <sub>0.028</sub>	0.649 <sub>0.029</sub>	<b>0.655<sub>0.028</sub></b>	0.646 <sub>0.029</sub>	0.638 <sub>0.028</sub>	0.629 <sub>0.025</sub>
2	16	0.675 <sub>0.030</sub>	0.527 <sub>0.038</sub>	<b>0.685<sub>0.029</sub></b>	0.528 <sub>0.038</sub>	0.612 <sub>0.033</sub>	0.598 <sub>0.034</sub>
	32	0.690 <sub>0.026</sub>	0.606 <sub>0.036</sub>	<b>0.701<sub>0.025</sub></b>	0.603 <sub>0.036</sub>	0.660 <sub>0.025</sub>	0.663 <sub>0.025</sub>
	48	0.695 <sub>0.026</sub>	0.657 <sub>0.031</sub>	<b>0.706<sub>0.024</sub></b>	0.648 <sub>0.040</sub>	0.678 <sub>0.023</sub>	0.681 <sub>0.023</sub>
	160	0.703 <sub>0.025</sub>	<b>0.716<sub>0.025</sub></b>	0.713 <sub>0.024</sub>	0.698 <sub>0.036</sub>	0.699 <sub>0.022</sub>	0.691 <sub>0.022</sub>
4	16	0.680 <sub>0.028</sub>	0.551 <sub>0.037</sub>	<b>0.688<sub>0.028</sub></b>	0.553 <sub>0.035</sub>	0.629 <sub>0.026</sub>	0.624 <sub>0.027</sub>
	32	0.695 <sub>0.023</sub>	0.645 <sub>0.031</sub>	<b>0.704<sub>0.023</sub></b>	0.638 <sub>0.036</sub>	0.673 <sub>0.022</sub>	0.677 <sub>0.021</sub>
	48	0.700 <sub>0.022</sub>	0.680 <sub>0.026</sub>	<b>0.709<sub>0.022</sub></b>	0.667 <sub>0.039</sub>	0.689 <sub>0.020</sub>	0.692 <sub>0.018</sub>
	160	0.708 <sub>0.021</sub>	<b>0.722<sub>0.021</sub></b>	0.717 <sub>0.022</sub>	0.703 <sub>0.033</sub>	0.706 <sub>0.017</sub>	0.702 <sub>0.017</sub>
8	16	0.669 <sub>0.026</sub>	0.588 <sub>0.034</sub>	<b>0.674<sub>0.028</sub></b>	0.590 <sub>0.034</sub>	0.633 <sub>0.022</sub>	0.630 <sub>0.022</sub>
	32	0.685 <sub>0.022</sub>	0.660 <sub>0.026</sub>	<b>0.690<sub>0.023</sub></b>	0.654 <sub>0.035</sub>	0.671 <sub>0.020</sub>	0.677 <sub>0.019</sub>
	48	0.689 <sub>0.021</sub>	0.684 <sub>0.023</sub>	<b>0.695<sub>0.022</sub></b>	0.670 <sub>0.037</sub>	0.684 <sub>0.018</sub>	0.687 <sub>0.016</sub>
	160	0.697 <sub>0.020</sub>	<b>0.712<sub>0.020</sub></b>	0.703 <sub>0.021</sub>	0.691 <sub>0.034</sub>	0.699 <sub>0.016</sub>	0.698 <sub>0.015</sub>
12	16	0.655 <sub>0.026</sub>	0.608 <sub>0.032</sub>	<b>0.661<sub>0.026</sub></b>	0.608 <sub>0.032</sub>	0.632 <sub>0.020</sub>	0.631 <sub>0.023</sub>
	32	0.670 <sub>0.021</sub>	0.662 <sub>0.024</sub>	<b>0.677<sub>0.022</sub></b>	0.654 <sub>0.031</sub>	0.666 <sub>0.019</sub>	0.672 <sub>0.019</sub>
	48	0.676 <sub>0.020</sub>	0.678 <sub>0.022</sub>	<b>0.682<sub>0.021</sub></b>	0.665 <sub>0.035</sub>	0.677 <sub>0.018</sub>	0.680 <sub>0.017</sub>
	160	0.683 <sub>0.019</sub>	<b>0.696<sub>0.019</sub></b>	0.689 <sub>0.020</sub>	0.681 <sub>0.033</sub>	0.690 <sub>0.016</sub>	0.690 <sub>0.015</sub>
16	16	0.642 <sub>0.026</sub>	0.614 <sub>0.030</sub>	<b>0.645<sub>0.027</sub></b>	0.615 <sub>0.028</sub>	0.627 <sub>0.017</sub>	0.628 <sub>0.017</sub>
	32	0.657 <sub>0.021</sub>	0.656 <sub>0.022</sub>	0.661 <sub>0.022</sub>	0.650 <sub>0.028</sub>	0.657 <sub>0.018</sub>	<b>0.663<sub>0.017</sub></b>
	48	0.663 <sub>0.020</sub>	0.668 <sub>0.020</sub>	0.666 <sub>0.021</sub>	0.658 <sub>0.032</sub>	0.666 <sub>0.016</sub>	<b>0.671<sub>0.017</sub></b>
	160	0.670 <sub>0.019</sub>	<b>0.682<sub>0.018</sub></b>	0.674 <sub>0.020</sub>	0.669 <sub>0.034</sub>	0.678 <sub>0.015</sub>	0.679 <sub>0.014</sub>

TAB. A.5 – Moyenne et écart-type des performances des différents algorithmes sur les paysages  $NK_p$  pour  $N = 16$ .

$K$	$p$	RN	RP	HC	HC2	EA	SE
1	0.5	0.482 <sub>0.066</sub>	0.493 <sub>0.064</sub>	0.488 <sub>0.066</sub>	0.496 <sub>0.066</sub>	<b>0.500</b> <sub>0.064</sub>	0.497 <sub>0.063</sub>
	0.8	0.267 <sub>0.066</sub>	0.275 <sub>0.065</sub>	0.256 <sub>0.072</sub>	0.276 <sub>0.066</sub>	<b>0.278</b> <sub>0.065</sub>	0.277 <sub>0.064</sub>
	0.9	0.166 <sub>0.060</sub>	0.169 <sub>0.060</sub>	0.151 <sub>0.062</sub>	0.169 <sub>0.060</sub>	<b>0.170</b> <sub>0.059</sub>	0.170 <sub>0.059</sub>
	0.95	0.097 <sub>0.046</sub>	0.100 <sub>0.046</sub>	0.083 <sub>0.046</sub>	0.100 <sub>0.046</sub>	<b>0.100</b> <sub>0.046</sub>	0.100 <sub>0.046</sub>
	0.99	0.020 <sub>0.028</sub>	<b>0.020</b> <sub>0.028</sub>	0.016 <sub>0.026</sub>	<b>0.020</b> <sub>0.028</sub>	<b>0.020</b> <sub>0.028</sub>	<b>0.020</b> <sub>0.028</sub>
2	0.5	0.492 <sub>0.058</sub>	0.503 <sub>0.054</sub>	0.499 <sub>0.055</sub>	0.508 <sub>0.060</sub>	<b>0.522</b> <sub>0.046</sub>	0.517 <sub>0.046</sub>
	0.8	0.297 <sub>0.064</sub>	0.309 <sub>0.061</sub>	0.289 <sub>0.069</sub>	0.304 <sub>0.068</sub>	<b>0.321</b> <sub>0.058</sub>	0.316 <sub>0.057</sub>
	0.9	0.205 <sub>0.054</sub>	0.214 <sub>0.053</sub>	0.180 <sub>0.064</sub>	0.210 <sub>0.057</sub>	<b>0.220</b> <sub>0.052</sub>	0.217 <sub>0.051</sub>
	0.95	0.123 <sub>0.057</sub>	0.129 <sub>0.057</sub>	0.094 <sub>0.057</sub>	0.126 <sub>0.057</sub>	<b>0.133</b> <sub>0.057</sub>	0.132 <sub>0.056</sub>
	0.99	0.040 <sub>0.040</sub>	0.039 <sub>0.040</sub>	0.023 <sub>0.032</sub>	0.039 <sub>0.039</sub>	0.042 <sub>0.041</sub>	<b>0.042</b> <sub>0.041</sub>
3	0.5	0.509 <sub>0.062</sub>	0.519 <sub>0.061</sub>	0.518 <sub>0.061</sub>	0.524 <sub>0.070</sub>	<b>0.552</b> <sub>0.052</sub>	0.544 <sub>0.052</sub>
	0.8	0.306 <sub>0.057</sub>	0.318 <sub>0.055</sub>	0.303 <sub>0.061</sub>	0.316 <sub>0.066</sub>	<b>0.338</b> <sub>0.050</sub>	0.330 <sub>0.049</sub>
	0.9	0.220 <sub>0.055</sub>	0.234 <sub>0.055</sub>	0.203 <sub>0.062</sub>	0.226 <sub>0.062</sub>	<b>0.246</b> <sub>0.052</sub>	0.241 <sub>0.050</sub>
	0.95	0.149 <sub>0.051</sub>	0.159 <sub>0.051</sub>	0.116 <sub>0.059</sub>	0.151 <sub>0.053</sub>	<b>0.167</b> <sub>0.049</sub>	0.164 <sub>0.047</sub>
	0.99	0.049 <sub>0.037</sub>	0.050 <sub>0.038</sub>	0.025 <sub>0.032</sub>	0.049 <sub>0.037</sub>	<b>0.059</b> <sub>0.036</sub>	0.058 <sub>0.036</sub>
5	0.5	0.503 <sub>0.060</sub>	0.512 <sub>0.059</sub>	0.511 <sub>0.059</sub>	0.524 <sub>0.070</sub>	<b>0.553</b> <sub>0.048</sub>	0.545 <sub>0.048</sub>
	0.8	0.317 <sub>0.056</sub>	0.328 <sub>0.054</sub>	0.321 <sub>0.057</sub>	0.335 <sub>0.067</sub>	<b>0.364</b> <sub>0.045</sub>	0.355 <sub>0.044</sub>
	0.9	0.224 <sub>0.047</sub>	0.237 <sub>0.046</sub>	0.218 <sub>0.053</sub>	0.234 <sub>0.057</sub>	<b>0.261</b> <sub>0.042</sub>	0.253 <sub>0.040</sub>
	0.95	0.166 <sub>0.044</sub>	0.182 <sub>0.042</sub>	0.143 <sub>0.053</sub>	0.172 <sub>0.052</sub>	<b>0.196</b> <sub>0.038</sub>	0.190 <sub>0.038</sub>
	0.99	0.067 <sub>0.036</sub>	0.076 <sub>0.036</sub>	0.033 <sub>0.035</sub>	0.073 <sub>0.037</sub>	<b>0.092</b> <sub>0.032</sub>	0.090 <sub>0.032</sub>
8	0.5	0.477 <sub>0.055</sub>	0.485 <sub>0.053</sub>	0.485 <sub>0.054</sub>	0.512 <sub>0.059</sub>	<b>0.524</b> <sub>0.043</sub>	0.519 <sub>0.043</sub>
	0.8	0.290 <sub>0.052</sub>	0.301 <sub>0.050</sub>	0.297 <sub>0.050</sub>	0.322 <sub>0.057</sub>	<b>0.337</b> <sub>0.043</sub>	0.332 <sub>0.044</sub>
	0.9	0.207 <sub>0.044</sub>	0.219 <sub>0.042</sub>	0.212 <sub>0.046</sub>	0.230 <sub>0.052</sub>	<b>0.245</b> <sub>0.037</sub>	0.240 <sub>0.037</sub>
	0.95	0.157 <sub>0.040</sub>	0.170 <sub>0.038</sub>	0.150 <sub>0.044</sub>	0.171 <sub>0.048</sub>	<b>0.188</b> <sub>0.036</sub>	0.184 <sub>0.035</sub>
	0.99	0.080 <sub>0.030</sub>	0.095 <sub>0.029</sub>	0.049 <sub>0.037</sub>	0.089 <sub>0.033</sub>	<b>0.102</b> <sub>0.026</sub>	0.099 <sub>0.025</sub>

TAB. A.6 – Moyenne et écart-type des performances des différents algorithmes sur les paysages  $NK_p$  pour  $N = 64$ .

$K$	$p$	RN	RP	HC	HC2	EA	SE
0	0.5	0.412 <sub>0.031</sub>	<b>0.414</b> <sub>0.030</sub>	<b>0.414</b> <sub>0.030</sub>	<b>0.414</b> <sub>0.030</sub>	0.408 <sub>0.029</sub>	0.403 <sub>0.029</sub>
	0.8	0.176 <sub>0.027</sub>	<b>0.177</b> <sub>0.027</sub>	<b>0.177</b> <sub>0.027</sub>	<b>0.177</b> <sub>0.027</sub>	0.177 <sub>0.027</sub>	0.176 <sub>0.027</sub>
	0.9	0.098 <sub>0.029</sub>	<b>0.098</b> <sub>0.029</sub>	<b>0.098</b> <sub>0.029</sub>	<b>0.098</b> <sub>0.029</sub>	<b>0.098</b> <sub>0.029</sub>	<b>0.098</b> <sub>0.029</sub>
	0.95	0.041 <sub>0.016</sub>	<b>0.041</b> <sub>0.016</sub>	<b>0.041</b> <sub>0.016</sub>	<b>0.041</b> <sub>0.016</sub>	<b>0.041</b> <sub>0.016</sub>	<b>0.041</b> <sub>0.016</sub>
	0.99	0.011 <sub>0.009</sub>	<b>0.011</b> <sub>0.009</sub>	<b>0.011</b> <sub>0.009</sub>	<b>0.011</b> <sub>0.009</sub>	<b>0.011</b> <sub>0.009</sub>	<b>0.011</b> <sub>0.009</sub>
2	0.5	0.500 <sub>0.030</sub>	0.516 <sub>0.029</sub>	0.512 <sub>0.031</sub>	<b>0.518</b> <sub>0.035</sub>	0.495 <sub>0.025</sub>	0.486 <sub>0.025</sub>
	0.8	0.284 <sub>0.029</sub>	<b>0.301</b> <sub>0.030</sub>	0.281 <sub>0.032</sub>	0.294 <sub>0.037</sub>	0.281 <sub>0.025</sub>	0.274 <sub>0.024</sub>
	0.9	0.184 <sub>0.022</sub>	<b>0.198</b> <sub>0.020</sub>	0.165 <sub>0.026</sub>	0.188 <sub>0.027</sub>	0.184 <sub>0.020</sub>	0.179 <sub>0.019</sub>
	0.95	0.117 <sub>0.029</sub>	<b>0.129</b> <sub>0.030</sub>	0.095 <sub>0.029</sub>	0.121 <sub>0.030</sub>	0.122 <sub>0.027</sub>	0.122 <sub>0.025</sub>
	0.99	0.027 <sub>0.011</sub>	0.028 <sub>0.011</sub>	0.017 <sub>0.011</sub>	0.028 <sub>0.011</sub>	0.031 <sub>0.010</sub>	<b>0.031</b> <sub>0.011</sub>
4	0.5	0.514 <sub>0.030</sub>	0.527 <sub>0.030</sub>	0.527 <sub>0.031</sub>	<b>0.528</b> <sub>0.039</sub>	0.512 <sub>0.023</sub>	0.502 <sub>0.022</sub>
	0.8	0.317 <sub>0.028</sub>	<b>0.337</b> <sub>0.028</sub>	0.325 <sub>0.031</sub>	0.328 <sub>0.037</sub>	0.317 <sub>0.023</sub>	0.305 <sub>0.022</sub>
	0.9	0.221 <sub>0.025</sub>	<b>0.242</b> <sub>0.023</sub>	0.215 <sub>0.028</sub>	0.229 <sub>0.036</sub>	0.221 <sub>0.020</sub>	0.212 <sub>0.019</sub>
	0.95	0.155 <sub>0.022</sub>	<b>0.177</b> <sub>0.021</sub>	0.134 <sub>0.027</sub>	0.158 <sub>0.033</sub>	0.157 <sub>0.018</sub>	0.151 <sub>0.017</sub>
	0.99	0.058 <sub>0.017</sub>	0.069 <sub>0.017</sub>	0.030 <sub>0.016</sub>	0.060 <sub>0.020</sub>	0.069 <sub>0.012</sub>	<b>0.070</b> <sub>0.012</sub>
8	0.5	0.500 <sub>0.028</sub>	0.509 <sub>0.029</sub>	<b>0.510</b> <sub>0.029</sub>	0.507 <sub>0.041</sub>	0.506 <sub>0.022</sub>	0.498 <sub>0.021</sub>
	0.8	0.312 <sub>0.026</sub>	<b>0.324</b> <sub>0.027</sub>	0.322 <sub>0.029</sub>	0.320 <sub>0.038</sub>	0.316 <sub>0.022</sub>	0.307 <sub>0.020</sub>
	0.9	0.224 <sub>0.023</sub>	<b>0.240</b> <sub>0.024</sub>	0.230 <sub>0.027</sub>	0.231 <sub>0.037</sub>	0.227 <sub>0.018</sub>	0.217 <sub>0.018</sub>
	0.95	0.165 <sub>0.021</sub>	<b>0.184</b> <sub>0.020</sub>	0.163 <sub>0.024</sub>	0.168 <sub>0.034</sub>	0.168 <sub>0.016</sub>	0.161 <sub>0.015</sub>
	0.99	0.079 <sub>0.017</sub>	<b>0.102</b> <sub>0.016</sub>	0.050 <sub>0.020</sub>	0.079 <sub>0.028</sub>	0.086 <sub>0.014</sub>	0.084 <sub>0.013</sub>
12	0.5	0.481 <sub>0.026</sub>	0.489 <sub>0.029</sub>	0.489 <sub>0.028</sub>	0.488 <sub>0.039</sub>	<b>0.490</b> <sub>0.022</sub>	0.486 <sub>0.021</sub>
	0.8	0.297 <sub>0.025</sub>	<b>0.306</b> <sub>0.027</sub>	0.306 <sub>0.027</sub>	0.304 <sub>0.038</sub>	0.305 <sub>0.020</sub>	0.298 <sub>0.019</sub>
	0.9	0.213 <sub>0.022</sub>	<b>0.223</b> <sub>0.024</sub>	0.219 <sub>0.024</sub>	0.219 <sub>0.033</sub>	0.219 <sub>0.017</sub>	0.213 <sub>0.017</sub>
	0.95	0.158 <sub>0.018</sub>	<b>0.171</b> <sub>0.019</sub>	0.162 <sub>0.022</sub>	0.162 <sub>0.031</sub>	0.161 <sub>0.015</sub>	0.156 <sub>0.015</sub>
	0.99	0.082 <sub>0.015</sub>	<b>0.104</b> <sub>0.015</sub>	0.065 <sub>0.020</sub>	0.083 <sub>0.027</sub>	0.087 <sub>0.012</sub>	0.085 <sub>0.011</sub>
16	0.5	0.464 <sub>0.026</sub>	0.471 <sub>0.027</sub>	0.471 <sub>0.027</sub>	0.472 <sub>0.039</sub>	<b>0.476</b> <sub>0.021</sub>	0.472 <sub>0.020</sub>
	0.8	0.282 <sub>0.024</sub>	0.288 <sub>0.025</sub>	0.289 <sub>0.024</sub>	0.288 <sub>0.038</sub>	<b>0.293</b> <sub>0.019</sub>	0.287 <sub>0.018</sub>
	0.9	0.200 <sub>0.021</sub>	0.207 <sub>0.022</sub>	0.207 <sub>0.022</sub>	<b>0.208</b> <sub>0.032</sub>	0.208 <sub>0.017</sub>	0.204 <sub>0.017</sub>
	0.95	0.147 <sub>0.018</sub>	<b>0.156</b> <sub>0.019</sub>	0.152 <sub>0.019</sub>	0.152 <sub>0.029</sub>	0.154 <sub>0.015</sub>	0.150 <sub>0.014</sub>
	0.99	0.081 <sub>0.014</sub>	<b>0.097</b> <sub>0.013</sub>	0.070 <sub>0.018</sub>	0.080 <sub>0.025</sub>	0.085 <sub>0.011</sub>	0.083 <sub>0.010</sub>



# Bibliographie

- [1] E.H.L. Aarts and P.J.M. van Laarhoven. Statistical cooling : a general approach to combinatorial optimization problems. *Philips Journal of Research*, 40(4) :193–226, 1985.
- [2] L. Altenberg. The evolution of evolvability in genetic programming. In Kenneth E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 3, pages 47–74. MIT Press, 1994.
- [3] Lee Altenberg. The Schema Theorem and Price’s Theorem. In L. Darrell Whitley and Michael D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 23–49, Estes Park, Colorado, USA, 1995. Morgan Kaufmann.
- [4] D. Andre, F. H. Bennett, and J. R. Koza. Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem. In J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, editors, *Genetic Programming 1996 : Proceedings of the First Annual Conference*, pages 3–11, Cambridge, MA, 1996. The MIT Press.
- [5] T. Back, F. Hoffmeister, and H. Schwefel. A survey of evolution strategies. In Lashon B. Belew and Richard K. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 2–9, San Diego, CA, 1991. Morgan Kaufmann.
- [6] Philippe Barbe and Michel Ledoux. *Probabilités. De la licence à l’agrégation*. Belin, 1998.
- [7] L. Barnett. Ruggedness and neutrality - the NKp family of fitness landscapes. In C. Adami, R. K. Belew, H. Kitano, and C. Taylor, editors, *ALIFE VI, Proceedings of the Sixth International Conference on Artificial Life*, pages 18–27. ALIFE, The MIT Press, 1998.
- [8] L. Barnett. *Evolutionary Search on Fitness Landscapes with Neutral Networks*. PhD thesis, University of Sussex, 2003.
- [9] Lionel Barnett. Netcrawling - optimal evolutionary search with neutral networks. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 30–37, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 2001. IEEE Press.
- [10] U. Bastolla, M. Porto, H. E. Roman, and M. Vendruscolo. Stastical properties of neutral evolution. *Journal Molecular Evolution*, 57(S) :103–119, August 2003.
- [11] Meriema Belaidouni and Jin-Kao Hao. An analysis of the configuration space of the maximal constraint satisfaction problem. In *PPSN VI : Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pages 49–58, London, UK, 2000. Springer-Verlag.
- [12] Meriema Belaidouni and Jin-Kao Hao. Sat, local search dynamics and density of states. In *Selected Papers from the 5th European Conference on Artificial Evolution*, pages 192–204, London, UK, 2002. Springer-Verlag.
- [13] E. Bornberg-Bauer and H.S. Chan. Modeling evolutionary landscapes : Mutational stability, topology and superfunnels in sequence space. *Proc. Natl. Acad. Sci. USA, track 2*, 96(19) :10689–10694, 1999.

- [14] Stefan Bornholdt. Probing genetic algorithm performance of fitness landscapes. In *FOGA*, pages 141–154, 1996.
- [15] G.E.P. Box and G. M. Jenkins. *Time series analysis, forecasting and control*. Holden Day, San Francisco, 1970.
- [16] John L. Bresina, Mark Drummond, and Keith Swanson. Expected solution quality. In *IJCAI*, pages 1583–1591, 1995.
- [17] M. S. Capcarrère, M. Sipper, and M. Tomassini. Two-state,  $r=1$  cellular automaton that classifies density. *Physical Review Letters*, 77(24) :4969–4971, December 1996.
- [18] B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press, Cambridge, UK, 1998.
- [19] Manuel Clergue, Philippe Collard, Marco Tomassini, and Leonardo Vanneschi. Fitness distance correlation and problem difficulty for genetic programming. In *GECCO '02 : Proceedings of the Genetic and Evolutionary Computation Conference*, pages 724–732, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- [20] P. Collard, M. Clergue, and M. Defoin Platel. Synthetic neutrality for artificial evolution. In *Artificial Evolution : Fourth European Conference AE'99*, pages 254–265. Springer-Verlag, 2000. Selected papers in Lecture Notes in Computer Sciences 1829.
- [21] Philippe Collard, Sébastien Verel, and Manuel Clergue. How to use the scuba diving metaphor to solve problem with neutrality? In R. L. de Mántaras and L. Saitta, editors, *Proceedings of the 2004 European Conference on Artificial Intelligence (ECAI04)*, pages 166–170, Valence, Spain, August 2004. IOS Press.
- [22] Philippe Collard, Sébastien Verel, and Manuel Clergue. Local search heuristics : Fitness cloud versus fitness landscape. In R. L. de Mántaras and L. Saitta, editors, *Poster at the 2004 European Conference on Artificial Intelligence (ECAI04)*, pages 973–974, Valence, Spain, August 2004. IOS Press.
- [23] M. Collins. Counting solutions in reduced boolean parity. In R. Poli, S. Cagnoni, M. Keijzer, and E. Costa, editors, *GECCO 2004 Workshop Proceedings*, Seattle, Washington, USA, 26-30 June 2004.
- [24] Christine Crisan and Heinz Mehlenbein. The frequency assignment problem : A look at the performance of evolutionary search. In *AE '97 : Selected Papers from the Third European Conference on Artificial Evolution*, pages 263–274, London, UK, 1998. Springer-Verlag.
- [25] J. P. Crutchfield, M. Mitchell, and R. Das. Evolutionary design of collective computation in cellular automata. In J. P. Crutchfield and P. Schuster, editors, *Evolutionary Dynamics : Exploring the Interplay of Selection, Accident, Neutrality, and Function*, pages 361–411. Oxford University Press, Oxford, UK, 2003.
- [26] Jason M. Daida, Robert R. Bertram, Stephen A. Stanhope, Jonathan C. Khoo, Shahbaz A. Chaudhary, Omer A. Chaudhri, and John Polito. What makes a problem gp-hard? analysis of a tunably difficult problem in genetic programming. *Genetic Programming and Evolvable Machines*, 2(2) :165–191, 2001.
- [27] R. Das, J. P. Crutchfield, M. Mitchell, and J. E. Hanson. Evolving globally synchronized cellular automata. In L. J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 336–343, San Francisco, CA, 1995. Morgan Kaufmann.
- [28] R. Das, M. Mitchell, and J. P. Crutchfield. A genetic algorithm discovers particle-based computation in cellular automata. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature- PPSN III*, volume 866 of *Lecture Notes in Computer Science*, pages 344–353, Heidelberg, 1994. Springer-Verlag.

- [29] Kenneth A. De Jong, William M. Spears, and Diana F. Gordon. Using markov chains to analyze GAFOs. In L. Darrell Whitley and Michael D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 115–137. Morgan Kaufmann, San Francisco, CA, 1995.
- [30] Michael Defoin Platel, Sébastien Verel, Manuel Clergue, and Philippe Collard. From royal road to epistatic road for variable length evolution algorithm. In Pierre Liardet, Pierre Collet, Cyril Fonlupt, Evelyne Lutton, and Marc Schoenauer, editors, *Evolution Artificielle, 6th International Conference*, volume 2936 of *Lecture Notes in Computer Science*, pages 3–14, Marseilles, France, 27-30 October 2003. Springer. Revised Selected Papers.
- [31] B. Derrida and L. Peliti. Evolution in a flat fitness landscape. *Bull. Math. Biol.*, 53 :355–382, 1991.
- [32] Marc Ebner. On the search space of genetic programming and its relation to nature’s search space. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 1357–1361, Mayflower Hotel, Washington D.C., USA, 6-9 July 1999. IEEE Press.
- [33] Marc Ebner, Mark Shackleton, and Rob Shipman. How neutral networks influence evolvability. *Complexity*, 7(2) :19–33, 2001.
- [34] N. Eldredge and S. J. Gould. Punctuated equilibria : An alternative to phyletic gradualism. In T. J. M. Schopf, editor, *Models in Paleobiology*, pages 82–115. Freeman Cooper, San Francisco, 1972.
- [35] R.A. Fisher. *The genetical theory of natural selection*, page 96. Dover Publication, 1930.
- [36] R.A. Fisher. The measurement of selective intensity. In *Proceedings of the Royal Society B*, volume 121, pages 58–62, 1936.
- [37] David B. Fogel and Adam Ghozeil. Using fitness distributions to design more efficient evolutionary computations. In *International Conference on Evolutionary Computation*, pages 11–19, 1996.
- [38] W. Fontana, P. F. Stadler, E. Bornberg-Bauer, T. Griesmacher, I. L. Hofacker, M. Tacker, P. Tarazona, E.D. Weinberger, and P. Schuster. RNA folding and combinatorial landscapes. *Phys. Rev E*, 47 :2083–2099, 1993.
- [39] H. Frauenfelder, A.R. Bishop, A. Garcia, A. Perelson, P. Schuster, D. Sherrington, and P.J. Swart. Landscape paradigms in physics and biology : Concepts, structures, and dynamics. *Physica D*, 107, 1997.
- [40] H. Fukús. Solution of the density classification problem with two cellular automata rules. *Physical Review E*, 55(3) :2081–2084, 1997.
- [41] L. Altenberg G. P. Wagner. Complex adaptations and the evolution of evolvability. In *Evolution*, pages 967–976, 1996.
- [42] P. Gacs, G. L. Kurdyumov, and L. A. Levin. One-dimensional uniform arrays that wash out finite islands. *Problemy Peredachi Informatsii*, 14 :92–98, 1978.
- [43] Francis Galton. Typical laws of heredity. In *Proceedings of the Royal Institution*, volume 8, pages 282–301, 1877.
- [44] Francis Galton. Regression toward mediocrity in hereditary stature. *Journal of the Anthropological Institute*, 15 :246–263, 1886.
- [45] Ricardo Garcia-Pelayo and Peter F. Stadler. Correlation length, isotropy, and meta-stable states. *Physica D*, 107 :240–254, 1997. Santa Fe Institute Preprint 96-05-034.
- [46] M. Garey and D. Johnson. *Computers and Intractability*. Freeman and Co., New York, 1979.



- [47] J. Garnier and L. Kallel. Statistical distribution of the convergence time of evolutionary algorithms for long-path problems. *IEEE-EC*, 4(1) :16, April 2000.
- [48] Josselin Garnier and Leila Kallel. Efficiency of local search with multiple local optima. *SIAM Journal on Discrete Mathematics*, 15(1) :122–141, 2002.
- [49] A. Gaspar and P. Collard. From gas to artificial immune systems : Improving adaptation in time dependent optimization. In *CEC'99 : IEEE International Congress on Evolutionary Computation*, volume 2, pages 1867–1874, 1999.
- [50] David E. Goldberg and Philip Segrest. Finite markov chain analysis of genetic algorithms. In *ICGA*, pages 1–8, 1987.
- [51] J. J. Grefenstette. Predictive models using fitness distributions of genetic operators. In D. Whitley, editor, *Foundations of Genetic Algorithms*, San Mateo, CA, 1995. Morgan Kaufmann Publishers.
- [52] Bruce Hajek. Cooling schedules for optimal annealing. *Math. Oper. Res.*, 13(2) :311–329, 1988.
- [53] J. E. Hanson and J. P. Crutchfield. Computational mechanics of cellular automata : An example. Technical Report 95-10-95, Santa Fe Institute Working Paper, 1995.
- [54] Inman Harvey and Adrian Thompson. Through the labyrinth evolution finds a way : A silicon ridge. In *ICES*, pages 406–422, 1996.
- [55] Robert B. Heckendorn. Embedded landscapes. *Evol. Comput.*, 10(4) :345–369, 2002.
- [56] W. Hordijk. Correlation analysis of the synchronizing-ca landscape. *Physica D*, 107 :225–264, 1997.
- [57] W. Hordijk, J. P. Crutchfield, and M. Mitchell. Mechanisms of emergent computation in cellular automata. In A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature- PPSN V*, volume 1498 of *Lecture Notes in Computer Science*, pages 613–622, Heidelberg, 1998. Springer-Verlag.
- [58] Wim Hordijk. A measure of landscapes. *Evolutionary Computation*, 4(4) :335–360, 1996.
- [59] Phil Husbands. Evolving robot behaviours with diffusing gas networks. In *Proceedings of the First European Workshop on Evolutionary Robotics*, pages 71–86. Springer-Verlag, 1998.
- [60] M. Huynen. Exploring phenotype space through neutral evolution. *Journal Molecular Evolution*, 43 :165–169, 1996.
- [61] Christian Igel and Kumar Chellapilla. Fitness distributions : Tools for designing efficient evolutionary computations. In Lee Spector, William B. Langdon, Una-May O'Reilly, and Peter J. Angeline, editors, *Advances in Genetic Programming 3*, chapter 9, pages 191–216. MIT Press, Cambridge, MA, USA, June 1999.
- [62] Christian Igel and Martin Kreutz. Using fitness distributions to improve the evolution of learning structures. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 3, pages 1902–1909, Mayflower Hotel, Washington D.C., USA, 1999. IEEE Press.
- [63] Christian Igel and Marc Toussaint. On classes of functions for which no free lunch results hold. *Inf. Process. Lett.*, 86(6) :317–321, 2003.
- [64] E. Izquierdo-Torres. The role of nearly neutral mutations in the evolution of dynamical neural networks. In J. Pollack and al, editors, *Ninth International Conference of the Simulation and Synthesis of Living Systems (Alife 9)*, pages 322–327. MIT Press, 2004.

- [65] T. Jones. *Evolutionary Algorithms, Fitness Landscapes and Search*. PhD thesis, University of New Mexico, Albuquerque, 1995.
- [66] H. Juillé and J. B. Pollack. Coevolutionary learning : a case study. In *ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning*, pages 251–259, San Francisco, CA, 1998. Morgan Kaufmann.
- [67] Hugues Juillé and Jordan B. Pollack. Coevolving the ideal trainer : Application to the discovery of cellular automata rules. In John R. Koza et al., editor, *Genetic Programming 1998 : Proceedings of the Third Annual Conference*, pages 519–527, University of Wisconsin, Madison, Wisconsin, USA, 22-25 1998. Morgan Kaufmann.
- [68] L. Kallel, B. Naudts, and C. R. Reeves. Properties of fitness functions and search landscapes. In Leila Kallel, Bart Naudts, and Alex Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, pages 175–206. Springer, Berlin, 2001.
- [69] S. A. Kauffman. *The Origins of Order*. Oxford University Press, New York, 1993.
- [70] S. A. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal Theoretical biology*, 128 :11–45, 1987.
- [71] Sami Khuri. Walsh and haar functions in genetic algorithms. In *SAC '94 : Proceedings of the 1994 ACM symposium on Applied computing*, pages 201–205, New York, NY, USA, 1994. ACM Press.
- [72] M. Kimura. Evolutionary rate at the molecular level. *Nature*, 217 :624–26, 1968.
- [73] M. Kimura. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, Cambridge, UK, 1983.
- [74] M. Kimura. Limitations of darwinian selection in a finite population. In *Proc. Natl. Acad. Sci. U.S.A.*, volume 92, pages 2343–2344, 1995.
- [75] Joshua D. Knowles and Richard A. Watson. On the utility of redundant encodings in mutation-based evolutionary search. In *PPSN*, pages 88–98, 2002.
- [76] J. R. Koza. *Genetic Programming*. The MIT Press, Cambridge, Massachusetts, 1992.
- [77] M. Land and R. K. Belew. No perfect two-state cellular automata for density classification exists. *Physical Review Letters*, 74(25) :5148–5150, June 1995.
- [78] R. Lande. Expected time for random genetic drift of a population between stable phenotypic states. In *Proceedings of the National Academy of Sciences of the USA*, volume 82, pages 7641–7645, 1985.
- [79] W. B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer, Berlin, 2002.
- [80] J. Levenick. Swappers : Introns promote flexibility, diversity and invention. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M.Jakiela, and R.E. Smith, editors, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference, GECCO'99*, volume 1, pages 361–368, San Francisco, California, 1999. Morgan Kaufmann Publisher.
- [81] J. Lobo, J. H. Miller, and W. Fontana. Neutrality in technology landscape, 2004.
- [82] N. Madras. *Lectures on Monte Carlo Methods*. American Mathematical Society, Providence, Rhode Island, 2002.
- [83] B. Manderick, M. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150, 1991.
- [84] J. Maynard-Smith. Natural selection and the concept of a protein space. *Nature*, 225 :563–564, 1970.

- [85] M. Mezard and R. Zecchina. The random k-satisfiability problem : from an analytic solution to an efficient algorithm. *Phys. Rev. E*, 66(056126), 2002.
- [86] Julian F. Miller and Peter Thomson. Cartesian genetic programming. In Riccardo Poli, Wolfgang Banzhaf, William B. Langdon, Julian F. Miller, Peter Nordin, and Terence C. Fogarty, editors, *Genetic Programming, Proceedings of EuroGP'2000*, volume 1802 of *LNCS*, pages 121–132, Edinburgh, 15-16 April 2000. Springer-Verlag.
- [87] M. Mitchell, J. P. Crutchfield, and P. T. Hraber. Evolving cellular automata to perform computations : Mechanisms and impediments. *Physica D*, 75 :361–391, 1994.
- [88] M. Mitchell, S. Forrest, and J. H. Holland. The royal road for genetic algorithms : Fitness landscape and GA performance. In F.J Varela and P. Bourguine, editors, *Proceedings of the First European Conference on Artificial Life*, pages 245–254, Cambridge, MA, 1992. MIT Press/Bradford Books.
- [89] M. Mitchell, P. T. Hraber, and J. P. Crutchfield. Revisiting the edge of chaos : Evolving cellular automata to perform computations. *Complex Systems*, 7 :89–130, 1993.
- [90] C.M. Newman, J.E. Cohen, and C. Kipnis. Neo-darwinian evolution implies punctuated equilibria. *Nature*, 315 :400–401, 1985.
- [91] M. Newman and R. Engelhardt. Effect of neutral selection on the evolution of molecular species. In *Proc. R. Soc. London B.*, volume 256, pages 1333–1338, 1998.
- [92] Erik Van Nimwegen and James P. Crutchfield. Optimizing epochal evolutionary search : Population-size dependent theory. *Mach. Learn.*, 45(1) :77–114, 2001.
- [93] Erik Van Nimwegen, James P. Crutchfield, and Martijn Huynen. Metastable evolutionary dynamics : Crossing fitness barriers or escaping via neutral paths? Technical Report 99-07-041, SanteFe institute, 1999.
- [94] Erik Van Nimwegen, James P. Crutchfield, and Martijn Huynen. Neutral evolution of mutational robustness. In *Proc. Nat. Acad. Sci. USA 96*, pages 9716–9720, 1999.
- [95] A. Nix and M. D. Vose. Modeling genetic algorithms with markov chains. *Annals of Mathematics and Artificial Intelligence*, 5 :79–88, 1992.
- [96] N. H. Packard. Adaptation toward the edge of chaos. In J. A. S. Kelso, A. J. Mandell, and M. F. Shlesinger, editors, *Dynamic Patterns in Complex Systems*, pages 293–301. World Scientific, Singapore, 1988.
- [97] R. Palmer. Optimization on rugged landscapes. *Molecular Evolution on Rugged Landscapes : Proteins, RNA and the Immune System*, pages 3–25, 1991.
- [98] K. Pearson. Mathematical contributions to the theory of evolution. iii. regression, heredity and panmixia. *Philosophical Transactions of the Royal Society of London*, 187 :253–318, 1896.
- [99] A.S. Perelson and S.A. Kauffman. *Molecular Evolution on Rugged Landscapes : Protein, RNA, and the Immune System*. Addison-Wesley, Reading, MA, 1991.
- [100] M. Defoin Platel. *Homologie en Programmation Génétique - Application à la résolution d'un problème inverse*. PhD thesis, Université de Nice Sophia Antipolis, France, 2004.
- [101] M. Defoin Platel, M. Clergue, and P. Collard. Maximum homologous crossover for linear genetic programming. In *Genetic programming EuroGP'2003*, volume 2610, pages 194–203. LNCS Springer-Verlag, 2003.
- [102] Mike Preuss, Lutz Schonemann, and Michael Emmerich. Counteracting genetic drift and disruptive recombination in  $(\lambda + \mu)$ -EA on multimodal fitness landscapes. In *GECCO '05 : Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 865–872, New York, NY, USA, 2005. ACM Press.

- [103] W. F. Punch. How effective are multiple populations in genetic programming. In J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. Riolo, editors, *Genetic Programming : Proceedings of the third annual conference*, pages 308–313, San Francisco, CA, 1998.
- [104] N. J. Radcliffe and P. D. Surry. Fundamental limitations on search algorithms : Evolutionary computing in perspective. In Jan van Leeuwen, editor, *Computer Science Today*, volume 1000 of *LNCS*. Springer-Verlag, 1995.
- [105] C. Reidys, P. Schuster, and P. Stadler. Generic properties of combinatorial maps — neutral networks of RNA secondary structures. *Bull. Math. Biol.*, 59(2) :339–397, 1997.
- [106] Christian M. Reidys and Peter F. Stadler. Combinatorial landscapes. Technical Report 01-03-014, Santa Fe Institute, March 2001.
- [107] Christian M. Reidys and Peter F. Stadler. Neutrality in fitness landscapes. *Applied Mathematics and Computation*, 117(2–3) :321–350, 2001.
- [108] Helge Rosé, Werner Ebeling, and Torsten Asselmeyer. The density of states - a measure of the difficulty of optimisation problems. In *Parallel Problem Solving from Nature*, pages 208–217, 1996.
- [109] Franz Rothlauf and David E. Goldberg. Redundant representations in evolutionary computation. *Evol. Comput.*, 11(4) :381–415, 2003.
- [110] C. Schumacher, M. D. Vose, and L. D. Whitley. The no free lunch and problem description length. In Lee Spector, Erik D. Goodman, Annie Wu, W. B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 565–570, San Francisco, California, USA, 7-11 2001. Morgan Kaufmann.
- [111] P. Schuster. Landscapes and molecular evolution. *Physica D*, 107 :351–365, 1997.
- [112] P. Schuster, W. Fontana, P. F. Stadler, and I. L. Hofacker. From sequences to shapes and back : a case study in RNA secondary structures. In *Proc. R. Soc. London B.*, volume 255, pages 279–284, 1994.
- [113] Peter Schuster and Peter F. Stadler. Landscapes : Complex optimization problems and biopolymer structures. *Computers & Chemistry*, 18(3) :295–324, 1994.
- [114] Peter Schuster and Peter F. Stadler. Networks in molecular evolutions. *Complexity*, 8 :34–42, 2002.
- [115] Chad W. Seys and Randall D. Beer. Evolving walking : The anatomy of an evolutionary search. In S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam, and J.-A. Meyer, editors, *Animals to Animats, Proceedings of the Eight international Conference on Simulation of Adaptive Behavior*, pages 357–363. The MIT Press, 2004.
- [116] M. Shackleton, R. Shipman, and M. Ebner. An investigation of redundant genotype-phenotype mapping and their role in evolutionary search. In *Proceeding of the 2000 Congress on Evolutionary Computation*, volume 1, pages 493–500, San Diego Marriott Hotel La Jolla, New York, 2000. IEEE Press.
- [117] P. Siarry and G. Dreyfus. *La méthode du recuit simulé : théorie et applications*. ESPCI IDSET, 1989.
- [118] M. Sipper. The evolution of parallel cellular machines : Toward evolware. *BioSystems*, 42 :29–43, 1997.
- [119] Smith, Husbands, Layzell, and O’Shea. Fitness landscapes and evolvability. *Evolutionary Computation*, 1(10) :1–34, 2001.

- [120] T.M.C. Smith, P. Husbands, and M. O’Shea. Neutral networks in an evolutionary robotics search space. In *Proceedings of the 2001 Congress on Evolutionary Computation : CEC2001*, pages 136–145. IEEE, Korea, 2001.
- [121] T.M.C. Smith, P. Husbands, and M. O’Shea. Not measuring evolvability : Initial exploration of an evolutionary robotics search space. In *Proceedings of the 2001 Congress on Evolutionary Computation : CEC2001*, pages 9–16. IEEE, Korea, 2001.
- [122] T.M.C. Smith, A. Philippides, P. Husbands, and M. O’Shea. Neutrality and ruggedness in robot landscapes. In *Congress on Evolutionary Computation : CEC2002*, pages 1348–1353. IEEE Press, 2002.
- [123] J.M. Hammersley S.R. Broadbent. Percolation processes i. crystals and mazes. In *Proceedings of the Cambridge Philosophical Society*, volume 53, pages 629–641, 1957.
- [124] P. F. Stadler. Fitness landscapes. In M. Lässig and Valleriani, editors, *Biological Evolution and Statistical Physics*, volume 585 of *Lecture Notes Physics*, pages 187–207, Heidelberg, 2002. Springer-Verlag.
- [125] Peter F. Stadler. Towards a theory of landscapes. In R. Lopez-Pena, R. Capovilla, R. Garcia-Pelayo, H. Waelbroeck, and F. Zertuche, editors, *Complex Systems and Binary Networks*, volume 461, pages 77–163, Berlin, New York, 1995. Springer Verlag.
- [126] Peter F. Stadler. Landscapes and their correlation functions. *J. Math. Chem.*, 20 :1–45, 1996.
- [127] Peter F. Stadler and W. Schnabl. The landscape of the traveling salesmen problem. *Phys. Letters, A*(161) :337–344, 1992.
- [128] Terry Stewart. Extrema selection : Accelerated evolution on neutral networks. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 25–29, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001. IEEE Press.
- [129] A. Thompson. An evolved circuit, intrinsic in silicon, entwined with physics. In T. Higuchi, M. Iwata, and W. Liu, editors, *Proceedings of The First International Conference on Evolvable Systems : From Biology to Hardware (ICES96)*, volume 1259 of *Lecture Notes in Computer Science*, pages 390–405. Springer-Verlag, Heidelberg, 1997.
- [130] Marc Toussaint and Christian Igel. Neutrality : A necessity for self-adaptation. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, pages 1354–1359, 2002.
- [131] Peter D. Turney. Increasing evolvability considered as a large scale trend in evolution. In Paul Marrow, Mark Shackleton, Jose-Luis Fernandez-Villacanas, and Tom Ray, editors, *GECCO’99 : Proceedings of the 1999 Genetic and Evolutionary Computation Conference, Workshop Program on evolvability*, pages 43–46, 1999.
- [132] Leonardo Vanneschi. *Theory and Practice for Efficient Genetic Programming*. PhD thesis, Faculty of Sciences, University of Lausanne, Switzerland, 2004.
- [133] Leonardo Vanneschi, Manuel Clergue, Philippe Collard, Marco Tomassini, and Sébastien Verel. Fitness clouds and problem hardness in genetic programming. In *Genetic and Evolutionary Computation – GECCO-2004*, volume 3103 of *LNCS*, pages 690–701, Seattle, 12-16 July 2004. Springer-Verlag.
- [134] Leonardo Vanneschi, Marco Tomassini, Manuel Clergue, and Philippe Collard. Difficulty of unimodal and multimodal landscapes in genetic programming. In *Genetic and evolutionary computation (GECCO2003)*, volume 2724 of *Lecture Notes in Computer Science*, pages 1788–1799. Springer-Verlag, Berlin, 2003.

- [135] Leonardo Vanneschi, Marco Tomassini, Philippe Collard, and Manuel Clergue. Fitness distance correlation in structural mutation genetic programming. In *Genetic Programming 6th European Conference (EuroGP2003)*, Lecture Notes in Computer Science, pages 455–464. Springer-Verlag, Heidelberg, 2003.
- [136] Vesselin K. Vassilev and Julian F. Miller. The advantages of landscape neutrality in digital circuit evolution. In *ICES*, pages 252–263, 2000.
- [137] Sébastien Verel, Philippe Collard, and Manuel Clergue. Where are bottleneck in NK fitness landscapes? In Ruhul Sarker, Robert Reynolds, Hussein Abbass, Kay Chen Tan, Bob McKay, Daryl Essam, and Tom Gedeon, editors, *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*, pages 273–280, Canberra, 8-12 December 2003. IEEE Press.
- [138] Sébastien Verel, Philippe Collard, and Manuel Clergue. Scuba search : when selection meets innovation. In *Proceedings of the 2004 Congress on Evolutionary Computation CEC2004*, pages 924–931, Portland, Oregon, 19-23 June 2004. IEEE Press.
- [139] B. Weinberg and E-G. Talbi. NFL theorem is unusable on structured classes of problems. In *Proceedings of the 2004 Congress on Evolutionary Computation CEC2004*, pages 220–226, Portland, Oregon, 19-23 June 2004. IEEE Press.
- [140] E. D. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. In *Biological Cybernetics*, pages 63 :325–336, 1990.
- [141] E. D. Weinberger. Local properties of kauffman’s NK model, a tuneably rugged energy landscape. *Physical Review A*, 44(10) :6399–6413, 1991.
- [142] C. O. Wilke. Adaptive evolution on neutral networks. *Bull. Math. Biol*, 63 :715–730, 2001.
- [143] S. Wolfram. *A New Kind of Science*. Wolfram Media, 2002.
- [144] David H. Wolpert and William G. Macready. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe, 1995.
- [145] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1) :67–82, April 1997.
- [146] S. Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proceedings of the Sixth International Congress of Genetics 1*, pages 356–366, 1932.
- [147] S. Wright. Character change, speciation, and the higher taxa. *Evolution*, 36(3) :427–443, 1982.
- [148] Tina Yu and Julian F. Miller. Neutrality and the evolvability of boolean function landscape. In *EuroGP ’01 : Proceedings of the 4th European Conference on Genetic Programming*, pages 204–217. Springer-Verlag, 2001.
- [149] Tina Yu and Julian F. Miller. Finding needles in haystacks is not hard with neutrality. In *EuroGP*, pages 13–25, 2002.







# Résumé

Le concept de paysage adaptatif a été introduit par S. Wright dans le domaine de la biologie de l'évolution dans les années 1930. Il est l'un des concepts pertinents pour modéliser l'évolution d'une population d'organismes. Dans le domaine de l'optimisation combinatoire par métaheuristiques, il est également utilisé et permet de lier une description géométrique d'un problème d'optimisation avec la dynamique des algorithmes de recherche. Deux géométries de paysage correspondant à deux dynamiques d'algorithme ont été principalement étudiées. La géométrie de paysage multimodale est liée à la présence d'optima locaux, où la dynamique est une succession de marches adaptatives vers de meilleures solutions et de dégradations de performance. La géométrie des paysages adaptatifs neutres, mise en avant par la théorie de la neutralité en évolution moléculaire de Motoo Kimura, est liée à la présence de plateaux ; la dynamique se caractérise alors par une dérive aléatoire entrecoupée de rares découvertes de solutions plus performantes. Cette thèse se propose d'approfondir l'étude des paysages neutres dans le contexte de l'optimisation et de proposer de nouvelles métaheuristiques adaptées à ce type de paysages.

La thèse se compose de quatre chapitres. Dans un premier chapitre, nous présentons les principaux résultats concernant les paysages adaptatifs et plus particulièrement les paysages adaptatifs neutres. Dans un deuxième chapitre, nous développons le concept d'ensemble de neutralité en introduisant la notion de 'nuage adaptatif' qui permet d'étudier la corrélation de performance entre solutions voisines et nous l'appliquons à la classe des paysages 'embarqués' qui regroupe les paysages NK et Max-SAT. Dans un troisième chapitre, nous résumons l'ensemble des mesures relatives aux réseaux de neutralité et nous proposons une nouvelle mesure. Une étude expérimentale est réalisée sur trois familles de paysages pour lesquelles la neutralité est ajustable et deux problèmes classiques de la littérature. Enfin, un nouvel algorithme de recherche adapté aux paysages neutres lié à la nouvelle mesure est proposé et évalué sur différents paysages neutres. Nous réalisons l'étude du paysage adaptatif massivement neutre issu du problème d'apprentissage de la règle d'un automate cellulaire réalisant la tâche de classification par la densité, afin d'en améliorer les métaheuristiques connues existantes.

**Mots-clés:** Paysage Adaptatif, Neutralité, Métaheuristique, Optimisation Difficile, Algorithme Évolutionnaire, Réseau de Neutralité

# Abstract

The concept of fitness landscape (or adaptive landscape) was introduced by S. Wright in the field of evolutionary biology in 1930's. It is one of the most relevant to explain the evolution of individuals. In the field of combinatorial optimization by metaheuristic, it is also used and allows to study the link between geometrical description of optimization problem and the dynamic of search algorithms. Two geometries of landscape which correspond to two dynamics of search have been studied. The multimodal geometry of landscape is related to the presence of local optima, where the search dynamic is a succession of adaptive walk toward better solutions and degradation of performance. The geometry of neutral fitness landscape, point out in molecular evolution by neutral theory of Motoo Kimura, is related to presence of plateaus ; the dynamic of search is characterized by random drift interrupted by the discover of rare better solution. This thesis propose to deeper study neutral fitness landscapes in the context of optimization and to design new metaheuristics according to those landscapes.

This thesis is composed by four parts. In the first one, we present the main results about fitness landscapes and more particularly about neutral fitness landscapes. In the second part, we develop the concept of neutral set by introducing the notion of 'fitness cloud' which allows to study the correlation of performance between two neighbor solutions and we measure this correlation on 'embedded fitness landscapes' as an extension of NK landscapes and Max-SAT problems. In the third part, we summarize the set of measures on neutral networks and we propose the new measure. Experimental study is performed on three family of landscapes for which the neutrality is and two classical problems. Then, a new metaheuristic adapted of neutral fitness landscapes inspired by the new measure is proposed and evaluated on different landscapes. We studied the massively neutral fitness landscapes from the learning problem of a rule of cellular automata which perform the density task, in order to improve the best metaheuristics known.

**Keywords:** Fitness Landscape, Neutrality, Metaheuristic, Hard Optimization, Evolutionary Algorithm, Neutral Network