

MODÈLES D'ABSTRACTION POUR LA RÉSOLUTION DE PROBLÈMES COMBINATOIRES

THÈSE D'HABILITATION À DIRIGER DES RECHERCHES
SPÉCIALITÉ : INFORMATIQUE

ADRIEN GOËFFON

Soutenance publique le 17 novembre 2014 devant un jury composé des personnalités suivantes :

Mme Clarisse DHAENENS, Professeure, École Polytechnique universitaire de Lille	Rapporteure
M. Jin-Kao HAO, Professeur, Université d'Angers	Examineur
M. Éric MONFROY, Professeur, Université de Nantes	Rapporteur
M. Frédéric SAUBION, Professeur, Université d'Angers	Garant
M. Marc SCHOENAUER, Directeur de recherche, INRIA Saclay	Rapporteur
Mme Christine SOLNON, Professeure, INSA Lyon	Examinatrice

Table des matières

Introduction générale	1
1 Principes de résolution de problèmes combinatoires par métaheuristiques	9
1.1 Optimisation combinatoire	9
1.1.1 Problème combinatoire	9
1.1.2 Approches de résolution	10
1.2 Métaheuristiques	12
1.2.1 Recherche locale	13
1.2.2 Algorithmes évolutionnaires	16
Première partie : Modèles en îles dynamiques, sélection adaptative des opérateurs	19
2 Modèles en îles dynamiques	21
2.1 Modèles en îles	21
2.2 Réguler dynamiquement les interactions entre individus	24
2.3 Modèle raffiné et application à la sélection d'opérateurs	27
3 Sélection adaptative d'opérateurs : étude des fonctions One-Max	31
3.1 Problème de la sélection adaptative des opérateurs	31
3.2 Le problème One-Max	33
3.2.1 Description du problème	33
3.2.2 Calcul théorique des opérateurs les plus appropriés	34
3.3 Modèle en îles dynamiques pour la sélection adaptative d'opérateurs	38
3.3.1 Optimisation d'une fonction One-Max	38
3.3.2 Influence des paramètres	41
4 Bandits comme modèle de sélection adaptative des opérateurs	45
4.1 Problème du bandit multi-bras	45
4.2 Stratégies de bandit	48
4.3 Bandits à bras interconnectés	53
4.4 Application de stratégies classiques aux bandits à bras interconnectés	59
4.5 Modèle en îles dynamique comme stratégie de bandit	61

Seconde partie : Escalader les paysages de fitness	65
Préambule	67
5 Définition et caractérisation des paysages de fitness	69
5.1 De la recherche par voisinage aux paysages de fitness	69
5.2 Définitions	72
5.3 Caractérisation des paysages de fitness	73
5.3.1 Dimension	73
5.3.2 Neutralité	73
5.3.3 Rugosité	75
5.4 Construire des paysages de fitness	78
5.4.1 Paysages NK	78
5.4.2 Incorporer de la neutralité aux paysages NK	81
5.4.3 Instances de problèmes combinatoires et paysages de fitness	83
5.5 Paysages de recherche : panel d'instances et propriétés	85
5.6 Stratégies d'escalade de paysages	87
6 Atteignabilité des optimums globaux	93
6.1 Visualisation des paysages de fitness	93
6.2 Bassin d'attraction et couverture	96
6.3 Atteignabilité des optimums globaux par un climber	97
6.4 Accessibilité des optimums	99
6.5 Influence de la solution initiale sur une recherche locale	102
6.6 Atteignabilité de l'optimum global par une recherche locale courte	104
7 Du choix de la règle pivot pour l'escalade de paysages de fitness	107
7.1 Comparaison des stratégies du premier et meilleur améliorant	107
7.2 Règles pivot alternatives et stratégie du moins bon améliorant	116
7.2.1 Du meilleur au moins bon améliorant	117
7.2.2 Approximer le moins bon améliorant	121
7.2.3 Application à des problèmes combinatoires académiques	126
7.2.4 Discussion	130
7.3 Vers d'autres règles pivot	131
8 Politiques de mouvements neutres et neutralité artificielle	135
8.1 Comparaison des politiques de mouvements neutres	135
8.2 Influence des climbers sur l'efficacité d'une recherche locale itérée.	147
8.3 Une neutralité artificielle pour des recherches locales plus efficaces	149
8.3.1 Neutralité artificielle et lissage de paysage	149
8.3.2 Paysages dynamiques à neutralité contrôlée	153
8.3.3 Recherche itérée simulée par l'escalade de paysages dynamiques	154
8.4 Climber stochastique et règle du moins bon améliorant	157
Conclusion générale	161
Références bibliographiques	163
 Annexe : Curriculum vitæ	 173

Introduction générale

Contexte scientifique et démarche personnelle

La résolution de problèmes combinatoires par algorithmes de recherche locale et évolutionnaires constitue le point central de mes travaux, depuis ma thèse de doctorat jusqu'à aujourd'hui. Prioritairement focalisé dans un premier temps sur les problèmes de bio-informatique¹, nécessitant davantage des outils de résolution *ad hoc* efficaces plutôt que des concepts algorithmiques plus génériques transposables à d'autres types de problèmes, j'ai cependant toujours orienté les procédés de résolution de manière à comprendre et anticiper le comportement des algorithmes de type évolutionnaire au sens large.

L'étude et la conception d'algorithmes approchés pour la résolution de problèmes difficiles peuvent s'aborder au moyen de deux démarches complémentaires. La première, plus fondamentale, consiste à rechercher des solutions algorithmiques indépendamment de toute application à un problème précis. Des instances éprouvées ou aléatoires de problèmes académiques, ou des cadres de travail applicatifs plus abstraits, permettent alors d'estimer leur efficacité relative selon un schéma de développement méthode \rightarrow problème. La seconde approche, plus appliquée, vise à aborder frontalement des problèmes particuliers aux propriétés structurelles spécifiques afin d'envisager des mécanismes de recherche originaux (problème \rightarrow méthode). Un tel processus d'élaboration de stratégies de résolution propres à un problème, voire à une instance de problème, permet de garder une vision précise de toutes les difficultés inhérentes aux problèmes difficiles ainsi que des verrous à lever.

À titre personnel, j'ai eu l'occasion d'observer lors de l'étude de certains problèmes de bio-informatique ou certains jeux combinatoires vus à plusieurs reprises lors de challenges de conférences en algorithmique évolutionnaire — résolution du puzzle *Eternity II* et détermination de chemins de résolution optimaux pour le *Rubik's Cube* — que des approches de résolution *ad hoc* pouvaient mener à des impasses. Malgré une expérience riche dans la résolution de ce type de problèmes, et surtout une volonté systématique de justifier chaque mécanisme de recherche, il m'est de plus en plus apparu manifeste qu'il est bien souvent extrêmement difficile voire impossible de déterminer le ou les éléments, cumulativement ou en combinaison, permettant de rendre la recherche plus efficace, plus rapide, plus robuste. Ce type de questionnement sans réponse satisfaisante m'est venu lors de ces études, mais également à de nombreuses reprises lors de congrès, de soutenances de thèses

¹Reconstruction phylogénétique [Goëffon, 2006; Goëffon *et al.*, 2008b], génomique comparative [Goëffon *et al.*, 2008a; Souciet *et al.*, 2009; Vyahhi *et al.*, 2009], caractérisation de données biologiques [Chhel *et al.*, 2012].

ou de lectures d'articles. Un exemple schématique est d'affirmer péremptoirement qu'un composant particulier original est efficace sur la seule observation empirique que l'algorithme devient bien plus performant avec que sans. Or il est aussi possible qu'il ne fasse que compenser la faiblesse d'un composant de base, ou d'une combinaison d'éléments particulière ; ou encore que son efficacité est conditionnée par des facteurs spécifiques comme les propriétés des instances ou, une nouvelle fois, d'autres choix de conception faits par ailleurs. Il ne serait cependant pas juste d'y voir systématiquement une faiblesse scientifique, puisque cette thématique de recherche ne dispose bien souvent d'aucun moyen théorique pour montrer la pertinence pratique d'une solution algorithmique, propriétés de convergence mises à part. Les instances de problèmes étudiées sont en effet si larges et variées que seules des justifications empiriques peuvent indiquer l'efficacité générale d'une méthode de recherche. Quoi qu'il en soit, il semble raisonnable de penser que le potentiel d'efficacité de ce type d'algorithmes dépend pour sa large part de ses briques de base plutôt que de mécanismes complexes régissant par exemple l'évolution d'un paramètre de diversification ou la détermination fine d'un ordre sur les variables à réinstancier.

Mes travaux de recherche sont guidés par une démarche globale qui vise à simplifier et unifier la représentation des algorithmes évolutionnaires afin de nous aider à en redécouvrir les principes de résolution généraux efficaces. L'élégance de l'idée de *métaheuristiques*, modèle d'abstraction proposant une carte restreinte de solutions algorithmiques pour la résolution approchée de divers problèmes d'optimisation, s'effrite en effet devant le nombre grandissant de techniques de résolution sophistiquées, permettant des calculs plus courts ou passant mieux à l'échelle, ou une prise en compte particulière des caractéristiques d'une certaine classe d'instances de problème. De classes d'algorithmes approchés génériques plutôt simples à utiliser pour un non spécialiste et se déclinant en quelques variantes seulement, ont émergé d'innombrables algorithmes en recherche d'originalité, complexes à appréhender, impossibles à répliquer et requérant parfois un nombre très important de paramètres difficiles à régler. À ce titre, beaucoup de nouvelles méthodes de résolution dites génériques développées ces dernières années n'apportent rien de pertinent à la communauté ou aux réels utilisateurs potentiels. Ce point critique a été soulevé par Sörensen [2013], dont l'article regorge de références illustrant notre propos. Un avis partagé avec l'auteur est que le champ disciplinaire des métaheuristiques doit évoluer non pas en élargissant la quantité de métaphores évolutionnaires et la complexité des méthodes, mais en se recentrant sur l'étude des composants de base des métaheuristiques, et le développement de paradigmes de plus haut niveau comme les algorithmes auto-adaptatifs.

Au sein de notre communauté, la *recherche autonome* [Hamadi *et al.*, 2011] est une thématique émergente qui étudie des concepts algorithmiques de haut niveau tels que les algorithmes auto-adaptatifs et les hyperheuristiques, dont le principe général est d'adapter leur mécanisme de résolution au moyen d'informations collectées au cours du processus. Cela concerne en particulier le réglage automatique de paramètres, ou la sélection adaptative d'opérateurs. Dans cet esprit, nous aborderons dans ce manuscrit le modèle d'algorithmes évolutionnaires dit *en îles* — où les individus sont partitionnés en sous-populations évoluant au moyen de migrations — que nous avons étendu en un modèle dynamique dont la topologie et les politiques de migration sont auto-adaptatives [Lardeux et Goëffon, 2010]. Davantage qu'un modèle en îles permettant d'optimiser les interactions entre individus, une telle représentation peut être utilisée comme un modèle d'abstraction pour

la sélection adaptative d’opérateurs [Candan *et al.*, 2012]. Par modèle d’abstraction, nous entendons que rien ne relie conceptuellement l’outil à sa finalité, soit ici la description et le fonctionnement algorithmique du modèle, à la problématique de la sélection d’opérateurs. Cela permet de s’affranchir de la complexité et de l’incertitude liées à la prise en compte d’informations propres aux problèmes et aux instances, afin de réduire en substance l’outil de résolution à sa marge d’action potentielle.

Un autre axe, qui s’est parallèlement imposé à la suite des différentes observations quant à la complexité grandissante des métaheuristiques et la difficulté d’en identifier les leviers de résolution efficaces, a consisté à revenir, sans *a priori*, sur les fondamentaux de la recherche locale. En premier lieu, nous nous focalisons sur les méthodes intensificatrices de base de type *amélioration itérative*, dont le rôle au sein des algorithmes de recherche locale est central mais dont les dynamiques restent, à en juger par la production scientifique, relativement peu étudiées au contraire des techniques perturbatives ; ces dernières ne représentant pourtant qu’une part minime de l’effort calculatoire fourni par ce type d’algorithmes. D’abord initiée comme démarche personnelle quitte à revenir, sur certains aspects, une vingtaine d’années en arrière — retour aux sources d’une époque dont je n’ai pas connu l’effervescence scientifique mais où le contexte était toutefois autre en termes de forces de calcul —, cette remise à plat de l’évaluation de ces méthodes intensificatrices en fonction des propriétés des instances et des politiques de mouvement [Basseur et Goëffon, 2013] a été si enrichissante que nous en avons prolongé activement l’analyse, jusqu’à identifier de nouvelles stratégies de mouvement efficaces [Basseur et Goëffon, 2014]. Pour mener à bien cette étude qui se focalise exclusivement sur l’aspect méthode, nous avons utilisé un cadre applicatif abstrait, celui des *paysages de fitness*, qui standardise l’évaluation du comportement des stratégies de recherche tout en permettant d’identifier des indicateurs de difficulté propres à certains problèmes et pouvant déterminer la dynamique des recherches.

Des travaux annexes et complémentaires en optimisation multiobjectif, qui ne seront pas détaillés ici, nous ont amené à redéfinir les problèmes d’optimisation multiobjectif comme des problèmes d’optimisation mono-objectif en considérant tout ensemble de solutions, dites solutions-éléments, comme une solution potentielle au problème, permettant ainsi la spécification de classes de voisinages et d’algorithmes de recherche locale basés sur les ensembles [Basseur *et al.*, 2013]. Cette étude est une abstraction supplémentaire des problèmes d’optimisation combinatoire, où la transformation d’un problème multiobjectif en problème mono-objectif permet une représentation alternative de la problématique et l’application de solutions algorithmiques nouvelles. De plus, cette représentation alternative des paysages multiobjectif en paysages mono-objectif est un élément d’appui supplémentaire à nos études sur la dynamique des recherches locales dans les paysages de fitness.

Présentation du manuscrit

Afin de mieux apprécier notre démarche scientifique, il nous a semblé plus approprié de développer de manière suffisamment détaillée ces différents travaux effectués relatifs aux outils génériques pour la résolution de problèmes combinatoires. C’est ainsi que nous avons fait le choix d’un manuscrit plus dense que si nous avions opté pour un plus tradi-

tionnel résumé des travaux accompagné d’une sélection d’articles. D’un point de vue plus personnel, le format choisi a constitué une motivation supplémentaire et fut l’occasion de reprendre ces différentes études, d’en extraire les points essentiels, d’y ajouter de nouveaux éléments, d’y annoter des réflexions plus générales qui n’auraient pas nécessairement eu leur place dans les articles originaux. En outre, incorporer quelques développements récents non encore publiés permet d’étayer notre propos général. Précisons que ce document ne constitue pas un rapport exhaustif de l’ensemble des travaux menés ces dernières années, mais se focalise sur les études les plus avancées, et s’inscrivant dans ce cadre des *modèles d’abstraction pour la résolution de problèmes combinatoires* ; c’est-à-dire que nous étudions ici la méthode plutôt que le problème, et identifions différentes manières de s’abstraire des problèmes à résoudre ou des problématiques de résolution — notons que ces deux expressions sont bien plus éloignées sémantiquement que syntaxiquement. Ces modèles d’abstraction mènent à la conceptualisation de stratégies de recherche simples, efficaces et génériques, concernant les politiques de mouvement et de choix d’opérateurs au sein d’algorithmes évolutionnaires.

Nous proposons à présent d’introduire plus précisément les deux parties développées dans ce manuscrit en présentant les principaux points et résultats qui y seront abordés.

I Modèles en îles dynamiques et sélection adaptative des opérateurs

Un algorithme évolutionnaire fait évoluer une population d’individus au moyen de transformations locales, et éventuellement de croisements. La représentation en îles, introduite pour pallier la convergence prématurée des algorithmes génétiques, est surtout utilisée pour faciliter la parallélisation des algorithmes à base de populations. Dans un tel modèle, les individus sont partitionnés en sous-populations, qui évoluent par le jeu des politiques migratoires. De nombreuses topologies de modèles ont été proposées (chaîne, anneau, hypercube, *etc.*), de même que des politiques de migration basées sur des mesures de diversité locale ou globale. Ces modèles restent difficiles à paramétrer, notamment concernant la fréquence et l’importance des migrations.

Le modèle que nous avons proposé, dit *dynamique* [Lardeux et Goëffon, 2010], n’est pas restrictif topologiquement — toutes les migrations sont envisageables *a priori* — et permet de gérer dynamiquement les flux migratoires en fonction de l’effet bénéfique ou non des plus récentes migrations effectuées. Plus précisément, une matrice de transition est mise à jour tout au long du processus de résolution selon un mécanisme d’apprentissage, ce qui permet de renforcer ou de réduire les migrations entre chaque couple d’îles. Une première application est de permettre la régulation auto-adaptative des migrations et par conséquent des croisements dans un algorithme évolutionnaire, de manière à combiner automatiquement intensifications locales et diversification globale. En outre, cette modélisation peut être vue comme une représentation alternative très générale du fonctionnement des algorithmes évolutionnaires, puisque les politiques migratoires classiques — non adaptatives — sont des cas particuliers, à paramétrage spécifique, du modèle dynamique.

Une évolution naturelle a été d’utiliser ce cadre de travail pour la sélection adaptative des opérateurs ou le réglage automatique de paramètres, en associant chaque île à des algorithmes ou des paramétrages d’algorithmes différents, c’est-à-dire dans les deux cas

et plus généralement, des *opérateurs*. Le mécanisme d'apprentissage fait que davantage d'individus sont envoyés sur les îles les plus pertinentes au regard d'un critère d'utilité. En associant des opérateurs différents à chaque île, ce processus permet d'allouer dynamiquement les individus aux opérateurs les plus utiles tout au long de la recherche. Nous montrons ainsi au moyen d'une fonction One-Max à maximiser que ce modèle parvient à suivre une politique d'application des opérateurs proche d'une politique montrée théoriquement comme optimale [Goëffon et Lardeux, 2011a].

La sélection adaptative des opérateurs pouvant s'abstraire en un problème de bandits, nous discutons dans ce manuscrit de quelques pistes pour rendre le cadre des bandits plus pertinent pour l'évaluation des stratégies de sélection d'opérateurs. À ce titre, nous étudions un modèle original de bandits dits à *bras interconnectés*, où l'efficacité d'un bras diminue à force d'être utilisé ; plus précisément, où le gain d'un bras est inversement proportionnel à son nombre d'activations sur une fenêtre de temps. Cette définition permet de prendre davantage en compte le comportement effectif des opérateurs dans les algorithmes évolutionnaires, où l'efficacité d'un opérateur d'intensification décroît progressivement lorsqu'il est appliqué, mais peut se recharger après application d'autres opérateurs. Nous discutons aussi de la possibilité d'utiliser le modèle en îles dynamique comme stratégie de bandit.

II Escalader les paysages de fitness

L'efficacité des algorithmes de recherche locale dépend fortement des caractéristiques des instances de problème à résoudre : nombre de variables, dépendances entre variables, corrélations entre solutions et fonction objectif, *etc.* Ces caractéristiques, ainsi que l'efficacité comparée de différentes stratégies de recherche, peuvent être évaluées par l'étude de paysages de fitness correspondants. Ces derniers abstraient l'instance à résoudre en surface multidimensionnelle dont les propriétés peuvent permettre d'anticiper la capacité de résolution de l'instance par un algorithme de recherche locale particulier. Cette abstraction en paysages est usuellement utilisée pour illustrer l'équilibre à respecter entre intensification et diversification de la recherche, et peut servir de justification *a priori* pour l'usage de certains mécanismes de recherche. Cependant, puisque la visualisation n'implique que des surfaces bi ou tridimensionnelles — dont les propriétés sont très différentes des paysages combinatoires habituels, fortement multidimensionnels —, les intuitions qu'elle amène se révèlent parfois erronées. Nous montrons en particulier que même une stratégie purement intensificatrice a un potentiel non négligeable pour atteindre des optimums globaux [Basseur *et al.*, 2014].

L'objectif principal de cette partie est de réexaminer, via l'étude de paysages de fitness, certains concepts basiques de la recherche locale dans le but de mieux comprendre l'effet des différentes stratégies d'intensification sur l'efficacité de la recherche. Une large étude empirique de l'efficacité des stratégies d'escalade réalisée sur des milliers de paysages variés nous a permis de déterminer les stratégies de recherche menant aux plus hauts optimums locaux, en fonction des principales propriétés caractérisant les paysages (dimension, rugosité, neutralité). Le point intéressant est que très souvent, les résultats que nous avons obtenus contredisent les choix de conception habituellement procédés lors de la résolution de problèmes combinatoires par métaheuristiques.

Un premier volet de cette étude est consacré à l’analyse comparée des deux règles pivot traditionnellement considérées (*meilleur* et *premier améliorant*). Nous y montrons notamment que la règle du premier améliorant est nettement moins pénalisée par la rugosité des paysages — qui détermine le nombre d’optimums locaux et la difficulté d’escalade — que la règle du meilleur améliorant, souvent privilégiée dans la conception de recherches locales [Basseur et Goëffon, 2013]. Ce résultat indique que sur ce type de paysages illustrant particulièrement la difficulté de résolution des problèmes d’optimisation combinatoire, sélectionner prioritairement les plus hauts voisins conduit le plus souvent à rencontrer prématurément des optimums locaux plus bas². Par la suite, nous avons montré que sur des paysages de structure relativement homogène, l’efficacité des recherches est directement corrélée à la qualité des voisins améliorants choisis à chaque itération, positivement ou négativement en fonction de la rugosité du paysage [Basseur et Goëffon, 2014]. Ainsi, sur les paysages plus rugueux, où le premier améliorant domine le meilleur améliorant, une stratégie plus exotique de type *moins bon améliorant* sera encore plus efficace dans le cas général.

Un second élément de comparaison des stratégies intensificatrices concerne les différentes politiques de sélection des mouvements neutres. L’objectif principal est de montrer que de réaliser des mouvements neutres au cours de la recherche, y compris lorsque des voisins strictement améliorants existent potentiellement, accélère la recherche et permet surtout d’atteindre de plus hauts optimums. Nous ne doutions pas que ce phénomène est connu de certains, mais l’absence de publication de référence abondant en ce sens, et le grand nombre d’algorithmes de recherche locale développés sans prendre en compte cette gestion sans doute contre-intuitive de la neutralité, nous a poussé à mener cette étude. Sur les paysages neutres, on constate que plus le taux de neutralité est important, plus une politique d’escalade qui ne distingue pas voisins neutres et voisins strictement améliorants, domine en moyenne les stratégies qui n’utilisent les mouvements neutres que pour sortir des optimums locaux. Cela se vérifie lors de la résolution de problèmes combinatoires académiques, en particulier sur toutes les instances de flowshop et la quasi-totalité des instances MAX-SAT étudiées, où appliquer des mouvements neutres tout au long de la recherche permet de converger plus rapidement vers de meilleures solutions. Suite à ces résultats, nous discutons de l’intérêt d’incorporer artificiellement de la neutralité aux paysages en discrétisant leur fonction d’évaluation afin de baisser en précision.

Organisation du document

Ce document est structuré en huit chapitres. Le chapitre 1 introduit et rappelle quelques fondements associés à l’optimisation combinatoire et aux métaheuristiques, en guise de prérequis pour chacune des deux parties, composées respectivement des chapitres 2 à 4, et 5 à 8.

Le chapitre 2 présente le concept de modèle en îles dynamique de manière relativement informelle. Nous illustrons cette présentation par quelques exemples applicatifs et discutons de différentes possibilités de régler les politiques de migration de manière autonome. Dans le chapitre 3, nous montrons que ce modèle peut être utilisé comme cadre de

²*Haut et bas* fait référence ici aux *points* des paysages de fitness dont le vocabulaire, abstraction oblige, est légèrement adapté par rapport à une échelle de qualité (*bon, mauvais*) relative à l’évaluation de solutions.

travail pour la sélection adaptative d'opérateurs de recherche locale. L'étude porte sur la maximisation d'une fonction One-Max au moyen d'opérateurs dont les espérances de gain évoluent au cours de la recherche et peuvent se déterminer analytiquement. Le chapitre 4 pousse plus loin l'abstraction de la problématique de sélection adaptative des opérateurs en l'associant à un problème de bandit. Nous y rappelons le fonctionnement de quelques stratégies de bandit usuelles, puis introduisons le modèle de bandit à bras interconnectés, qui permet de prendre en compte certains aspects de l'évolution des utilités des opérateurs.

Le chapitre 5 est consacré à la définition et la caractérisation des paysages de fitness et des stratégies d'escalade simples appelées *climbers*. On y détaille également le matériel expérimental — à savoir les instances de paysages — qui sera utilisé au cours de cette seconde partie. Dans le chapitre 6, nous discutons de biais que peuvent engendrer de mauvaises abstractions des espaces et paysages de recherche combinatoires, et indiquons que les stratégies d'intensification offrent un potentiel exploratoire plus important que certaines intuitions pourraient laisser penser. La large étude sur les *climbers*, qui constitue l'élément principal de cette partie, est divisée en deux chapitres. Les stratégies d'escalade strictes sont abordées dans le chapitre 7, et la prise en compte de la neutralité dans le chapitre 8.

Nous espérons que le lecteur prendra plaisir à parcourir ces pages, comme nous en avons eu à conduire ces recherches.

Collaborateurs

Les travaux présentés dans ce manuscrit sont le fruit de collaborations fructueuses avec quelques brillants collègues, dont un simple remerciement rendrait insuffisamment justice à leur apport. Frédéric Saubion, qui me fait l'honneur de m'encourager dans cette démarche de solliciter l'habilitation et m'a toujours fait part de remarques pertinentes sur l'ensemble de mes travaux, a plus particulièrement participé aux études sur les bandits et la sélection adaptative d'opérateurs (*cf.* chapitre 4). Tous les travaux préliminaires sur les modèles en îles, en particulier la définition, l'implémentation du modèle, et ses diverses applications ont été réalisés en collaboration avec Frédéric Lardeux (*cf.* chapitres 2 à 4). L'intégralité des recherches sur les paysages de fitness et les recherches locales (*cf.* chapitres 5 à 8) a été réalisée conjointement avec Matthieu Basseur. Deux études en particulier ont été menées grâce à l'excellent travail de deux étudiants : Vincent Vigneron (*cf.* chapitre 6) et Hugo Traverson (*cf.* section 8.3). Enfin, j'intègre à cette liste Arnaud Liefoghe et Sébastien Verel pour la partie sur l'optimisation multiobjectif que j'ai dû à regret renoncer de détailler dans ce mémoire déjà suffisamment long. Je tiens en particulier à souligner, d'un point de vue plus général, l'apport notable de Sébastien Verel, dans le sens où certains de ses conseils m'ont directement ou indirectement conduit à explorer des pistes et considérer des problématiques qui trouvent aujourd'hui un grand écho tout au long de ce manuscrit.

Chapitre 1

Principes de résolution de problèmes combinatoires par métaheuristiques

1.1 Optimisation combinatoire

Au sens large, l'optimisation consiste à déterminer l'élément ou les éléments d'un ensemble qui maximise(nt) un certain critère. Selon le type d'ensemble considéré (continu ou discret), l'optimisation sera dite continue ou discrète. Dans ce dernier cas, on parlera également d'optimisation combinatoire, selon une condition implicite discutée ci-après. Bien qu'il s'agisse de deux branches sœurs qui partagent naturellement un certain nombre de concepts, l'optimisation continue et l'optimisation combinatoire constituent deux disciplines bien distinctes, principalement car les méthodes de résolution efficaces sont faiblement transposables de l'une à l'autre. Ce document se focalise sur l'optimisation combinatoire et traitera d'outils et de concepts permettant la représentation et le traitement de problèmes combinatoires.

1.1.1 Problème combinatoire

Comme nous l'avons indiqué, un problème d'optimisation combinatoire consiste à déterminer un des *meilleurs* éléments d'un ensemble discret X (ou plus généralement l'ensemble des meilleurs éléments). Ces dits éléments sont appelés *solutions optimales* (ou *optimums globaux*) au sens d'un critère à maximiser ou minimiser donné par une *fonction objectif* affectant aux éléments de X une valeur numérique. La définition 1 s'applique aux problèmes de maximisation seulement, mais la définition d'un problème de minimisation s'en déduit aisément.

Définition 1 (Problème d'optimisation combinatoire) Soit X un ensemble discret de solutions (ou configurations), appelé espace de recherche, et $f : X \rightarrow \mathbb{K}$ une fonction

objectif³. Le problème d'optimisation combinatoire (X, f) revient à déterminer

$$\operatorname{argmax}_{x \in X} f(x),$$

c'est-à-dire l'ensemble des solutions x^* telles que $\forall x \in X, f(x) \leq f(x^*)$.

Il faut préciser que l'on admet plusieurs déclinaisons de définition d'un problème combinatoire. Tout d'abord, dans de nombreux cas pratiques, déterminer l'une des solutions optimales, même s'il en existe plusieurs, est une condition de résolution suffisante. Ensuite, dans le cas plus général de l'optimisation sous contraintes, on peut distinguer au sein de X un sous-ensemble S de solutions dites *réalisables*, parmi lequel les solutions optimales seront déterminées. Le problème consiste alors à résoudre $\operatorname{argmax}_{x \in S} f(x)$, mais cela ne change pas fondamentalement la manière d'aborder ce type de problèmes.

Enfin, il faut noter que le terme *problème* est ambigu car il peut aussi bien désigner un problème précis où les données sont connues et instanciées — c'est à dire un couple (X, f) unique —, ou bien la modélisation d'une problématique globale, à savoir une définition générale des couples (X, f) pouvant se décliner en fonction des données, si ce n'est à l'infini, du moins en très grand nombre. Dans ce dernier cas, un couple (X, f) sera une instance de problème et un problème sera une collection d'instances [Papadimitriou et Steiglitz, 1998]. La difficulté d'un problème sera jugée à la capacité de résoudre une instance en général ou tout du moins en fonction de sa taille, et non en particulier. Le contexte permet généralement de déterminer sans ambiguïté si l'on se réfère à un problème au sens de *classe de problèmes* — soit d'énoncé — ou au sens d'*instance de problème*, et nous espérons qu'il en sera de même dans ce manuscrit.

1.1.2 Approches de résolution

Même si déterminer parmi les 28 dominos celui possédant le plus grand nombre de points⁴ peut être considéré par définition comme un problème combinatoire, il ne l'est pas réellement par essence. L'adjectif *combinatoire* implique en effet une explosion du nombre de solutions à envisager en fonction des données du problème. Ainsi, dans les problèmes combinatoires communément étudiés, il existe au moins un rapport exponentiel entre les définitions en intention (regroupant les données nécessaires) et en extension de l'instance d'un problème et en particulier de l'ensemble X . Nombre de ces problèmes — dont ceux qui nous intéresseront ici — sont NP-difficiles, c'est-à-dire que leur résolution exacte dans le cas général passe également, dans l'état actuel des connaissances, par un nombre de calculs à effectuer exponentiellement plus grand que la taille des données.

Deux approches peuvent donc être employées pour aborder de tels problèmes combinatoires dans le cas général : les *algorithmes exacts*, non polynomiaux mais permettant de déterminer une solution optimale à coup sûr, et les *algorithmes approchés*, polynomiaux mais ne garantissant pas que la solution calculée soit optimale.

³ \mathbb{K} est un ensemble de valeurs scalaires, généralement \mathbb{N} , \mathbb{Z} , \mathbb{D} ou \mathbb{R} .

⁴Problème que l'on peut formaliser par $\operatorname{argmax}_{(x,y) \in \{0,\dots,6\}^2, x \leq y} (x + y)$.

Algorithmes exacts

L'algorithme de résolution exact le plus simple, l'*exploration exhaustive*, consiste à évaluer toutes les solutions, donc à calculer $f(x)$ pour tout $x \in X$ afin d'en déduire $\operatorname{argmax}_{x \in X} f(x)$; sa complexité dépendant directement de $\#X$. Suivant le même principe mais évitant d'énumérer l'intégralité de l'espace de recherche grâce à des mécanismes de coupe, les algorithmes de type *Branch & Bound* [Land et Doig, 1960], introduits il y a une cinquantaine d'années, sont toujours largement utilisés aujourd'hui. Néanmoins, l'explosion combinatoire des problèmes fait que les algorithmes exacts ne sont efficaces que sur des instances ne dépassant pas une certaine taille — très variable selon les problèmes.

Algorithmes approchés

Lorsqu'il semble peu probable de déterminer de manière certaine une solution optimale à un problème combinatoire dans des temps de calcul raisonnables — ce qui est typique des problèmes NP-difficiles —, l'alternative est d'utiliser ou de concevoir un algorithme approché ou *heuristique*. Le but est alors de calculer la meilleure solution possible au problème sans que la taille de l'instance ne soit un frein rédhibitoire à l'accomplissement de la méthode. Habituellement, les critères permettant de juger de l'efficacité d'un algorithme approché sont sa qualité (aptitude à produire de bonnes solutions, soit à déterminer un des meilleurs éléments de X au sens de f), sa robustesse (manière dont sa qualité est affectée par d'autres facteurs que la taille des données), sa rapidité (dépendant de sa complexité et donnée par son temps de calcul moyen ou maximal), et sa capacité à passer à l'échelle (manière dont les critères précédents évoluent lorsque les instances de problèmes deviennent plus grandes).

On trouve globalement trois types d'algorithmes approchés [Hao *et al.*, 1999] pour l'optimisation combinatoires :

- (i) les approches *constructives*⁵ comme les *heuristicques gloutonnes*, qui construisent progressivement une solution réalisable unique en visant le meilleur choix à faire à chaque étape (par exemple une instanciation de variable) ;
- (ii) la *recherche locale*, qui part d'une solution initiale et y effectue une série de modifications locales ;
- (iii) les approches *évolutionnaires*, qui considèrent simultanément un ensemble de solutions appelé population et sur lequel est appliqué un processus évolutionnaire généralement basé sur des concepts naturels.

Ces trois familles manipulent respectivement (i) une solution, (ii) une suite de solutions, et (iii) une suite d'ensembles de solutions. Les algorithmes hybrides, combinant plusieurs approches, sont fréquemment distingués comme une famille supplémentaire ; cependant, en définissant les types d'algorithmes par les objets manipulés, on peut tout aussi bien considérer que les algorithmes hybridant construction et recherche locale sont des algorithmes de recherche locale, et que les algorithmes hybridant construction et évolution, ou encore évolution et recherche locale, sont des algorithmes évolutionnaires.

⁵Précisons que tous les algorithmes constructifs ne sont pas nécessairement approchés.

Au contraire des approches purement constructives, les algorithmes de recherche locale et évolutionnaires reviennent au final à passer en revue et évaluer une partie très restreinte de l'espace de recherche et à retourner la meilleure solution rencontrée. La typicité de l'algorithme concerne alors la manière d'échantillonner X . Or un constat courant est que peu de techniques de résolution *ad hoc* sont réellement efficaces pour résoudre des problèmes NP-difficiles. Tous ces problèmes se réduisant polynomialement l'un à l'autre, le principal verrou à leur résolution exacte est commun, et en conséquence la plupart des concepts algorithmiques développés pour les aborder sont abstraits des problèmes. Ces méthodes de résolution génériques, indépendantes des connaissances particulières sur le problème à traiter, sont regroupées sous le terme *métaheuristiques* [Glover, 1986].

1.2 Métaheuristiques

Les algorithmes de recherche locale, évolutionnaires et hybrides, qui décrivent un schéma de résolution global pouvant être appliqué sans variation majeure à différents types de problèmes, sont des métaheuristiques. Cette caractéristique leur confère un niveau d'abstraction élevé, ce qui entraîne une séparation nette entre problème et méthode de résolution. Néanmoins, ces méthodes constituent également des cadres de travail, et peuvent tout aussi bien servir de base à un algorithme de résolution plus sophistiqué, ou encore d'élément au cœur d'une méthode de plus haut niveau.

Plusieurs dizaines de métaheuristiques ont été définies et appliquées à un large spectre de problèmes combinatoires [Bianchi *et al.*, 2009], mais leur hiérarchisation, à considérer qu'elle présente un intérêt, peut être problématique. En effet il n'existe pas véritablement de taxonomie exhaustive établie des métaheuristiques, tant les critères retenus pour leur classification varient en fonction des études [Byrne, 1997; Hao *et al.*, 1999; Talbi, 2002; Boussaïd *et al.*, 2013; Hao et Solnon, 2014].

Nous avons vu que les algorithmes approchés — et donc les métaheuristiques par extension — peuvent se distinguer en premier lieu par leur mécanisme général : *constructifs* pour ceux qui construisent une solution incrémentalement (algorithmes gloutons et gloutons aléatoires), *de recherche locale* — au sens large — pour ceux qui appliquent des modifications locales successives à une solution courante (descente, recuit simulé, recherche tabou, recherche locale itérée, voir section 1.2.1), et *évolutionnaires* pour ceux qui font évoluer un ensemble de solutions courantes (algorithmes génétiques, voir section 1.2.2). Il est cependant de plus en plus d'usage de regrouper les algorithmes de recherche locale et évolutionnaires au sein de la famille des *métaheuristiques perturbatives* (incluant les algorithmes génétiques hybrides et les recherches locales à population comme l'algorithme *Go with the winner* [Aldous et Vazirani, 1994]), et d'inclure dans la famille des *métaheuristiques constructives* des algorithmes plus sophistiqués comme l'*optimisation par colonies de fourmis* [Dorigo, 1992; Dorigo et Birattari, 2010] (qui repose sur une succession de solutions construites), voire des *algorithmes par estimation de distributions* [Larrañaga et Lozano, 2002] qui, s'ils sont basés sur des algorithmes génétiques, peuvent également être vus comme des algorithmes gloutons aléatoires.

Nous choisissons ici de nous focaliser sur les métaheuristiques perturbatives dont les concepts constituent l'un des points centraux de ce manuscrit. Ces méthodes de résolution

sont basées sur une exploration appropriée de l'espace de recherche en sélectionnant la ou les prochaines solutions à évaluer à partir des solutions préalablement rencontrées. Les algorithmes de recherche locale utilisent une relation de voisinage permettant de naviguer d'une solution à une autre, tandis que les algorithmes génétiques sont principalement basés sur un opérateur de variation permettant de générer de nouvelles solutions à partir de plusieurs solutions courantes.

1.2.1 Recherche locale

Un algorithme de recherche locale explore l'espace de recherche en parcourant un ensemble de solutions via une relation de voisinage — définie au moyen d'une fonction $\mathcal{N} : X \rightarrow 2^X$ — et une fonction d'évaluation $f : X \rightarrow \mathbb{K}$. La relation de voisinage associe à chaque solution $x \in X$ un sous-ensemble de solutions $\mathcal{N}(x) \subseteq X$. \mathcal{N} est usuellement décrite par les issues possibles d'une transformation particulière simple appliquée à une solution afin de pouvoir être définie en intension pour l'ensemble de l'espace de recherche, mais aussi d'associer des solutions proches structurellement. La fonction d'évaluation associe à chaque solution x un scalaire, par exemple un nombre décimal, induisant ainsi une relation d'ordre entre les solutions. Si $f(x) < f(x')$, alors x' est dite *meilleure*⁶ que x . Si $f(x) = f(x')$ ($x' \neq x$), alors x et x' sont dites *équivalentes*. Souvent, la fonction objectif f est utilisée comme fonction d'évaluation pour résoudre un problème (X, f) au moyen d'un algorithme de recherche locale.

Comme l'illustre l'algorithme 1, considérant un espace de recherche et une relation de voisinage (donnée par une fonction), un algorithme de recherche locale sera défini par :

- une manière de déterminer une solution de départ (initiale),
- une politique de mouvement (quel voisin sélectionner), et
- un critère d'arrêt.

Combinée au choix de la relation de voisinage, la politique de mouvement détermine grandement l'efficacité d'un algorithme de recherche locale. De nombreuses politiques ont été définies de manière générale, donnant lieu à des métaheuristiques bien connues comme le recuit simulé ou la recherche tabou.

Solution initiale

On dispose usuellement de deux alternatives pour démarrer une recherche locale. La première consiste à générer aléatoirement une solution, processus rapide mais qui n'offre pas la possibilité, dans le cas général, de partir d'une solution de qualité convenable. Dans ce cas de figure, c'est uniquement la capacité de la méthode à naviguer efficacement dans l'espace de recherche qui permettra de déterminer une solution de bonne qualité. Si l'instance traitée est difficile, l'algorithme ne bénéficie initialement d'aucun prérequis et la recherche peut partir dans une mauvaise direction.

⁶Ainsi, *un voisin meilleur* est toute solution voisine de la solution courante dont l'évaluation lui est strictement supérieure, tandis que *le meilleur voisin* de la solution courante est la solution voisine dont l'évaluation est la plus élevée.

Algorithme 1 : Schéma général d'un algorithme de recherche locale.

Données : (X, f) : une instance de problème ; \mathcal{N} : une fonction de voisinage ; f une fonction d'évaluation (éventuellement f).

Résultat : x^* , solution approchée de $\operatorname{argmax}_{x \in X} f(x)$.

```
1 sélectionner une solution initiale  $x_0 \in X$  ;
2  $x \leftarrow x_0$  ; //  $x$  est la solution courante
3  $x^* \leftarrow x$  ; //  $x^*$  est la meilleure solution rencontrée au sens de  $f$ 
4 tant que le critère d'arrêt n'est pas respecté faire
5   | sélectionner une solution voisine  $x' \in \mathcal{N}(x)$  ;
6   |  $x \leftarrow x'$  ;
7   | si  $f(x) > f(x^*)$  alors
8   |   |  $x^* \leftarrow x$ 
9   | fin
10 fin
11 retourner  $x^*$ 
```

La seconde alternative consiste à partir d'une solution calculée par une heuristique constructive comme un algorithme glouton. Cela permet d'initialiser la recherche à une solution pouvant déjà être de bonne qualité. Néanmoins, le risque est ici d'avoir été mal orienté par le processus constructif ; ce qui peut réduire les possibilités d'atteindre, malgré une politique de mouvement efficace, de nombreuses solutions potentiellement de meilleure qualité mais aux propriétés éloignées.

Le choix de la solution initiale dépend essentiellement du contexte. Si la recherche locale a pour vocation d'être lancée une unique fois, par exemple dans le cas de problèmes où l'évaluation des solutions est coûteuse, alors utiliser un algorithme constructif pour initialiser la recherche peut être envisagé si celui-ci est suffisamment efficace. Cependant et dans de nombreux contextes, les recherches locales, qui restent des algorithmes d'échantillonnage, sont exécutées plusieurs fois de manière indépendante sur une même instance (*multi-start local search*) afin de maximiser l'efficacité globale de l'algorithme et réduire les éventuels biais stochastiques. Il est alors préférable d'initialiser chacune de ces recherches à des solutions initiales suffisamment diversifiées pour permettre une exploration globale plus large de l'espace de recherche, ce que favorisent des solutions initiales aléatoires.

Politique de mouvement

Le point central d'un algorithme de recherche locale est sa politique de mouvement, identifiable dans l'algorithme 1 par le seul terme *sélectionner*. L'algorithme dit de descente, que dans un contexte de maximisation nous préférons nommer escalade, ou *hill-climbing*, ne sélectionne que des voisins *non détériorants*, c'est-à-dire meilleurs ou éventuellement équivalents. Plusieurs politiques différentes entrent dans le cadre des algorithmes de hill-climbing qui sont alors déterminés par une combinaison de sous-politiques dont la *règle pivot*. La seconde partie de ce manuscrit est intégralement consacrée à ce type d'algorithmes, dans un contexte particulier quoique très analogue des paysages de fitness. Nous

enrichirons alors le propos sur les différentes stratégies de hill-climbing, mais aussi sur d'autres aspects plus généraux des algorithmes de recherche locale.

Évoquons tout de même brièvement quelques politiques de mouvement plus sophistiquées qui peuvent sélectionner, dans le cas général, des voisins dégradants, c'est-à-dire de moins bonne évaluation. Ces politiques dites *perturbatives*⁷ permettent de contourner le traditionnel problème des *optimums locaux*, solutions n'ayant aucun voisin meilleur et qui peuvent bloquer certaines recherches locales utilisant des politiques de mouvement trop restrictives.

Le *recuit simulé* [Kirkpatrick *et al.*, 1983; Černý, 1985] modifie la solution courante x de manière à générer une solution voisine x' , puis valide cette modification selon une certaine probabilité dépendante de la variation d'évaluation obtenue $\Delta f = f(x') - f(x)$ ainsi que d'un paramètre de température τ qui décroît dans le temps. Cette probabilité, égale à $e^{-\frac{\Delta f}{\tau}}$, permet de sélectionner automatiquement tout voisin meilleur ou équivalent. Quant aux voisins détériorants, ceux-ci sont de moins en moins probablement acceptés au cours de la recherche. Le mécanisme de sélection proprement dit itère le processus jusqu'à ce qu'un voisin soit sélectionné, ou que le critère d'arrêt mette un terme à la recherche. Le recuit simulé est une adaptation à la recherche locale de l'algorithme de Metropolis–Hastings [Metropolis *et al.*, 1953] permettant de simuler des lois de probabilités à partir de chaînes de Markov.

La *recherche tabou* [Glover, 1986]⁸ utilise une mémoire afin de diriger la recherche. La version la plus simple consiste à garder en mémoire les dernières solutions rencontrées, tout en sélectionnant à chaque itération le meilleur voisin absent de cette mémoire (*non tabou*), indépendamment de l'évaluation de la solution courante. Ce dernier point le distingue d'une stratégie de hill-climbing puisqu'il permet la sélection d'un voisin détériorant lorsque la solution courante est un optimum local. La mémoire permet alors d'interdire de revenir sur une solution courante récente — donc de fortement diminuer la possibilité de revenir sur une ancienne solution courante — afin de ne pas cycler. La mémoire est appelée *liste tabou* et peut, plutôt que de mémoriser des solutions, conserver des attribus particuliers ayant été transformés, si l'on modélise la recherche locale non pas comme une marche dans l'espace de recherche, mais comme une transformation successive de solution. La recherche tabou comporte quelques paramètres critiques comme la longueur de la liste tabou, qui peut être statique ou dynamique [Battiti et Tecchioli, 1994], ou la définition du critère d'*aspiration*. Le fait de mémoriser des attribus plutôt que des solutions peut diversifier plus rapidement la recherche mais également rendre de nombreux voisins pertinents tabou. Le critère d'aspiration permet de lever le statut tabou de certains voisins, mais pour garder la cohérence de la stratégie, l'aspiration ne doit pas provoquer de cycle. Des recherches tabou dites *évoluées* utilisent plusieurs listes appelées mémoires à court, moyen et long terme, permettant d'alterner des phases d'intensification avec des phases de diversification.

Cette alternance entre intensification et diversification est plus simplement décrite par la *recherche locale itérée* [Lourenço *et al.*, 2003], qui consiste à effectuer plusieurs recherches locales à la suite, séparées par des phases de perturbations. Ces dernières permettent de

⁷Le terme *perturbative* s'applique ici à l'évaluation de la solution courante, tandis que les métaheuristiques dites *perturbatives* le sont au regard des solutions elles-mêmes et regroupent davantage de méthodes.

⁸Hansen [1986] a développé simultanément une approche similaire.

généraliser les solutions initiales de chaque recherche locale, en partant d'un optimum local précédemment rencontré (typiquement le dernier ou le meilleur). La stratégie de recherche locale utilisée hors des perturbations peut dans ce cas être simplement un hill-climbing. Une perturbation se résume généralement à une simple marche aléatoire, c'est-à-dire une succession de sélections aléatoires de voisins, mais peut aussi être réalisée par une nouvelle recherche locale utilisant une fonction d'évaluation bruitée — ou plus concrètement la fonction d'évaluation originale appliquée sur un bruitage de l'instance de problème. Une recherche locale itérée ne se définit pas par sa politique de mouvement mais répète un processus de recherche locale défini par ailleurs. En ce sens, cette métaheuristique ne respecte pas le cadre algorithmique de la recherche locale tel que décrit par l'algorithme 1 et peut ainsi être qualifiée d'hybride. Il en est de même pour la métaheuristique de *descente à voisinage variable* [Mladenović et Hansen, 1997] qui emploie plusieurs voisinages utilisés alternativement.

Critère d'arrêt

Lors de la conception d'un algorithme de recherche locale, le critère d'arrêt peut s'avérer être l'élément occasionnant le plus de questionnement. Seules certaines politiques de mouvement intègrent un critère d'arrêt évident si plus aucun voisin n'est sélectionnable, comme c'est le cas du hill-climbing strict lorsque la solution courante est un optimum local⁹. Dans le cas contraire, le critère d'arrêt spécifie soit un certain temps de calcul alloué, soit un certain nombre d'itérations global à effectuer (généralement un nombre de voisins évalués), ou encore un nombre d'itérations sans amélioration de la meilleure solution x^* .

1.2.2 Algorithmes évolutionnaires

Les algorithmes évolutionnaires constituent une large famille d'algorithmes dont le mécanisme de base est inspiré du processus de sélection naturelle en théorie de l'évolution. Au plus près de cette métaphore, les *algorithmes génétiques* — introduits par Holland [1975] puis revisités par Goldberg [1989] — manipulent un ensemble de solutions appelé *population d'individus* pouvant se croiser, muter et disparaître au gré des générations. Les nouveaux individus sont créés à partir d'individus parents — généralement deux — sélectionnés dans la population, et héritent de certains de leurs caractères. Les individus les plus adaptés sont ceux ayant une meilleure évaluation — qualifiée ici de *fitness* — et ont de meilleures chances de survie, donc de perdurer dans la population. L'efficacité d'un algorithme génétique est déterminée par un grand nombre de paramètres (représentation ou *codage* des individus, taille de la population, sélection des parents, opérateurs de croisement et de mutation, fréquence des mutations, mise à jour de la population, *etc.*).

Appliqué à la résolution d'un problème combinatoire, un algorithme génétique peut ne pas être suffisamment efficace, dans le cas fréquent où la seule pression de sélection ne permet pas d'intensifier suffisamment la recherche. C'est pourquoi les algorithmes génétiques sont fréquemment hybridés avec des méthodes de recherche locale (on parle d'*algorithmes*

⁹À ce propos, l'algorithme 1 est volontairement simplifié et il est implicite que lorsqu'une sélection de voisin est impossible selon la politique de mouvement, alors l'algorithme s'arrête. Des algorithmes plus formels seront proposés dans la seconde partie qui se centre sur l'étude des stratégies de hill-climbing.

mémétiques [Moscato, 1999] ou de *recherche locale génétique* [Ulder *et al.*, 1991]). Les recherches locales transforment les individus-solutions selon un mécanisme tel que décrit précédemment, ce qui renforce la part intensificatrice de l'algorithme. Les croisements et mutations permettent de diversifier la recherche. Sur ce dernier point, une gestion efficace de la population doit éviter une convergence prématurée de ce type d'algorithme, qui s'observe lorsque les individus se ressemblent trop.

Présenter en détail la riche famille des algorithmes évolutionnaires, dérivés des algorithmes génétiques de Holland [1975] et des *stratégies d'évolution* [Rechenberg, 1965], ainsi que leurs multiples déclinaisons, nécessiterait une longue étude qui s'éloigne de notre propos. Nous préférons référer le lecteur aux articles fondateurs déjà cités, ainsi qu'aux revues et ouvrages très complets de Michalewicz [1996], Beyer et Schwefel [2002], Eiben et Smith [2003], Kicinger *et al.* [2005] ou Jansen [2013].

Première partie

**Modèles en iles dynamiques,
sélection adaptative des opérateurs**

Chapitre 2

Modèles en îles dynamiques

2.1 Modèles en îles

Relativement simples à concevoir, les algorithmes génétiques peuvent produire sur de nombreux problèmes d'optimisation de très bons résultats en termes de qualité des solutions et de robustesse de l'algorithme, particulièrement s'ils sont hybridés avec des mécanismes de recherche locale. Leur efficacité dépend néanmoins d'un certain nombre de facteurs, de la représentation des configurations [Michalewicz, 1996] à la fonction d'évaluation (mesure de *fitness*) utilisée [Wolpert et Macready, 1997], en passant par les opérateurs de mutation et de croisement définis [Richter, 2010] ou encore des paramètres plus généraux (taille de la population, fréquence des mutations, contrôle de la diversité, sélections des individus, élitisme, *etc.*) [Goldberg *et al.*, 1991; Goldberg et Deb, 1991; Spears, 2000; Eiben et Smit, 2012]. Même avec un effort particulier pour adapter un algorithme génétique efficace à un problème particulier, on observe rapidement sur des instances plus critiques ou difficiles un certain nombre de limitations en termes de performances générales ou de passage à l'échelle de l'algorithme.

Afin de rendre les algorithmes génétiques et mémétiques plus efficaces, une technique est d'utiliser une représentation de la population pouvant permettre de combiner plus efficacement l'intensification et la diversification de la recherche, mais aussi de rendre l'algorithme plus facilement parallélisable [Whitley *et al.*, 1998]. Depuis une vingtaine d'années et l'avènement des premiers algorithmes évolutionnaires distribués [Tanese, 1989], les modèles en îles sont de plus en plus étudiés dans la communauté évolutionnaire. Le principe est de partitionner la population d'individus en sous-populations (îles) évoluant indépendamment par périodes tout en pouvant interagir par moments par le biais de migrations d'individus. Inspirée par la nature, cette extension intuitive et logique des algorithmes évolutionnaires à base de population, et en premier lieu des algorithmes génétiques, permet d'intégrer deux niveaux de convergence (locale et globale) pour des algorithmes plus efficaces et robustes.

La principale problématique est de définir à la fois la topologie du modèle (*où peut migrer tel individu ?*) et les politiques de migration (*quel individu doit migrer ?*) de manière à ralentir la convergence générale de la population tout en favorisant le brassage global des bons individus. Araujo *et al.* [2009] dressent un état de l'art des modèles en îles

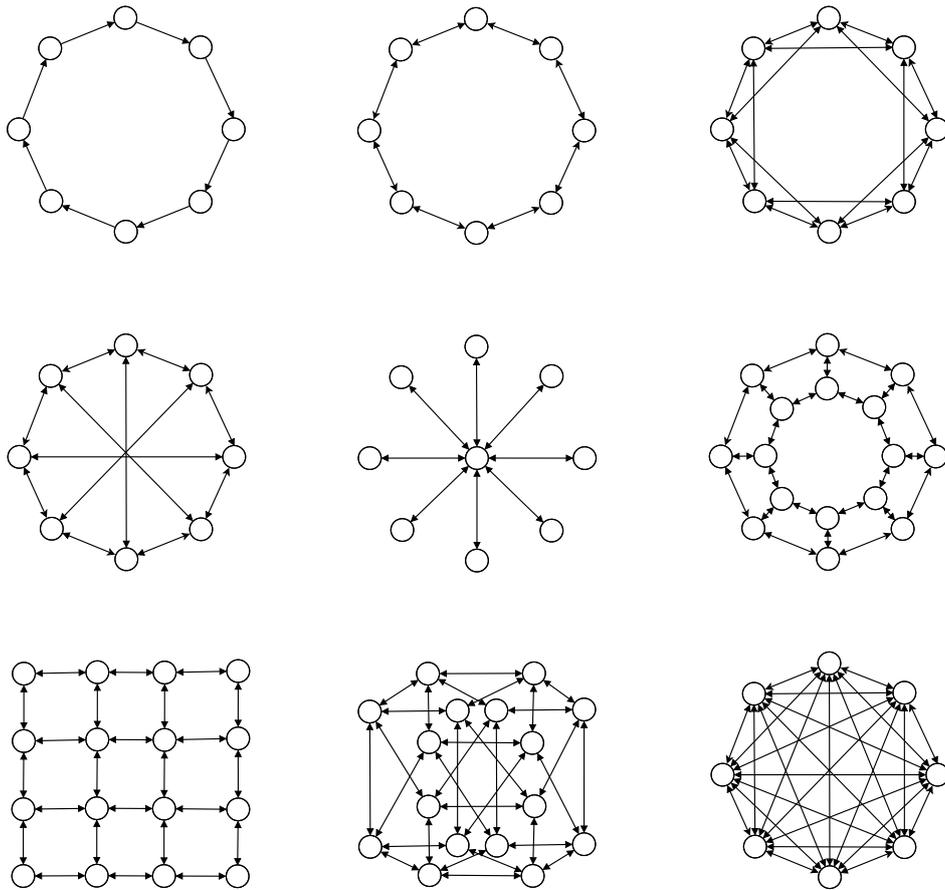


FIGURE 1 – Différentes topologies statiques envisageables : anneau unidirectionnel, anneau bidirectionnel d'ordre 1 puis 2, roue, étoile, tore, grille, hypercube, complet.

et discutent notamment des différentes politiques de migration. On constate alors qu'un nombre important de topologies [Gustafson et Burke, 2006; Ruciński *et al.*, 2010] (voir figure 1) et de politiques [Cantú-Paz, 2001; Noda *et al.*, 2002; Ursem, 2002; Denzinger et Kidney, 2003; Eldos, 2006; Araujo *et al.*, 2009] ont été définies, grandement basées sur des mesures de diversité locales (par population) et globales (sur l'ensemble des îles). La pertinence d'utiliser une topologie plutôt qu'une autre n'est pas clairement établie [Lopes *et al.*, 2014]. Et dans tous les cas, la fréquence et la quantité des migrations demeurent difficile à déterminer [Skolicki et De Jong, 2005], d'autant plus que les interactions entre les deux niveaux d'évolution (intra- et inter-îles) sont très importantes [Skolicki, 2008]. Idéalement, une stratégie d'évolution globale intelligente devrait tirer un avantage de ces interactions et maximiser le bénéfice des migrations. Mais à partir de simples indicateurs intra- et inter-îles — typiquement des mesures de diversité et de qualité —, il est difficile de prédire à quel moment des individus doivent migrer, lesquels, où et pour quel impact.

Bien souvent, les modèles en îles sont utilisés de manière relativement statique, où les individus migrent de population à population selon un schéma prédéfini, ou bien sont choisis spécifiquement pour renforcer la diversité locale des populations. L'algorithme 2

présente un exemple volontairement simple d’algorithme évolutionnaire utilisant le concept d’îles, dont la politique de migration consiste à établir une rotation des meilleurs individus d’île en île — donc en suivant une topologie en anneau unidirectionnel.

Algorithme 2 : Exemple de fonctionnement d’un algorithme évolutionnaire dont la population est partitionnée en îles.

```

1 pour chaque île  $i \in \{1 \dots m\}$  faire
2   |  $P_i \leftarrow \text{Initialiser\_Population}()$ 
3 fin
4 tant que le critère d’arrêt n’est pas respecté faire
5   | // Évolution
6   | pour chaque île  $i \in \{1, \dots, m\}$  faire
7     |  $P_i \leftarrow \text{Evoluer\_Population}(P_i)$ ;
8     |  $x_i \leftarrow \text{Meilleur\_Individu}(P_i)$ 
9   | fin
10  | // Migration
11  | pour chaque île  $i \in \{2 \dots m\}$  faire
12    |  $P_i \leftarrow P_i \cup \{x_{i-1}\} \setminus \{x_i\}$ 
13  | fin
14  |  $P_1 \leftarrow P_1 \cup \{x_m\} \setminus \{x_1\}$ 
15 fin

```

En 2010, nous avons introduit un modèle en îles dynamique dans un but d’auto-adapter topologie et politiques de migration au cours de la recherche, en fonction d’indicateurs définis en paramètres [Lardeux et Goëffon, 2010; Goëffon et Lardeux, 2012; Candan *et al.*, 2012]. Ces indicateurs peuvent être basés sur les propriétés des sous-populations, mais surtout, et dans le cadre d’une approche originale et plus efficace, les effets des précédentes migrations. Dans la section 2.2, nous évoquons notre première étude qui fut de définir dynamiquement une politique plus optimisée de croisements entre individus dans les algorithmes génétiques pour une meilleure efficacité. Ce modèle en îles dynamique constitue en outre un cadre de travail original pour la définition et la conception d’algorithmes évolutionnaires.

La section 2.3 présente un modèle dynamique ajusté, et une extension du champ d’application à la sélection automatique, dans le cadre d’un algorithme évolutionnaire, des opérateurs de recherche locale les plus pertinents au cours de la recherche. Dans cette extension, les îles ne symbolisent plus qu’un simple partitionnement entre les individus, mais disposent chacune de leurs propres règles et paramétrages, par exemple un opérateur de recherche locale propre. Cette présentation sera un prélude aux chapitres suivants qui seront davantage focalisés sur la sélection d’opérateurs, avec notamment le modèle en îles comme modèle d’abstraction.

2.2 Réguler dynamiquement les interactions entre individus

Comme le rappellent Araujo *et al.* [2009], plusieurs paramètres définissent un modèle en îles :

- le nombre d'individus pouvant migrer,
- la politique de sélection des individus migrants,
- la politique de remplacement des individus migrants,
- la topologie de la communication entre sous-populations, et
- la nature synchrone ou asynchrone de la connexion entre sous-populations.

Le modèle proposé dans [Lardeux et Goëffon, 2010] généralise ces paramètres, tout en permettant la régulation dynamique des migrations. Nous représentons ce modèle en îles dynamique par un digraphe où les nœuds représentent les îles (les sous-populations) et les arcs les possibilités de migrations. Chaque arc possède une valeur qui représente la probabilité pour un individu de migrer d'une île vers une autre à chaque déclenchement d'un processus migratoire. Le caractère auto-adaptatif de ce modèle est géré au moyen d'un mécanisme de récompenses et pénalités. Les probabilités sont mises à jour après chaque cycle migratoire en fonction des effets des dernières migrations. Si l'île qui accueille un individu voit sa population améliorée (respectivement détériorée) alors la probabilité de migration correspondante est augmentée (respectivement diminuée). Ici, la qualité des populations est évaluée en termes de fitness moyenne.

Le contrôle dynamique des paramètres, comme le taux de migration, rend possible la manipulation d'îles de différentes tailles. Ce mécanisme empêche les sous-populations de faible qualité de dépenser autant d'effort calculatoire que celles de bonne qualité, et permet de gérer la fusion de populations.

Modèle en îles dynamique : un cas simple

Nous illustrons figure 2 un exemple d'évolution dynamique des politiques migratoires d'un modèle en îles. L'état du modèle représenté en (a) affecte à chaque individu, une probabilité $p_{i \rightarrow j}$ de migration vers chacune des îles (j), selon l'île sur laquelle il se trouve (i). Lors du prochain processus migratoire (b), chaque individu est affecté à une île destination en fonction des probabilités $p_{i \rightarrow j}$. La plupart d'entre eux restent sur la même île — puisqu'ici les valeurs des boucles sont élevées — et quelques individus se déplacent. On fait alors évoluer les populations de chaque île au moyen d'opérateurs évolutionnaires de variation (croisement, mutation, recherche locale). La politique migratoire peut alors évoluer dynamiquement (c) en augmentant (resp. diminuant) les probabilités $p_{i \rightarrow j}$ selon l'itinéraire des parents dont les descendants (individus obtenus par croisement) améliorent (resp. détériorent) la population.

Dans notre définition originelle du modèle, nous avons proposé des récompenses et pénalités fixes, de type jetons. Par exemple, sur la figure 2.c, si nous prenons séparément l'île 2 et que la valeur de récompense (ou pénalité) est fixée à 5 points (± 0.05), on peut supposer que la politique a été mise à jour selon le scénario suivant :

- $p_{1 \rightarrow 2}$ diminue de 0.50 à 0.40 car deux individus venant de l'île 1 ont produit un individu fils détériorant l'évaluation de la population de l'île 2 ;

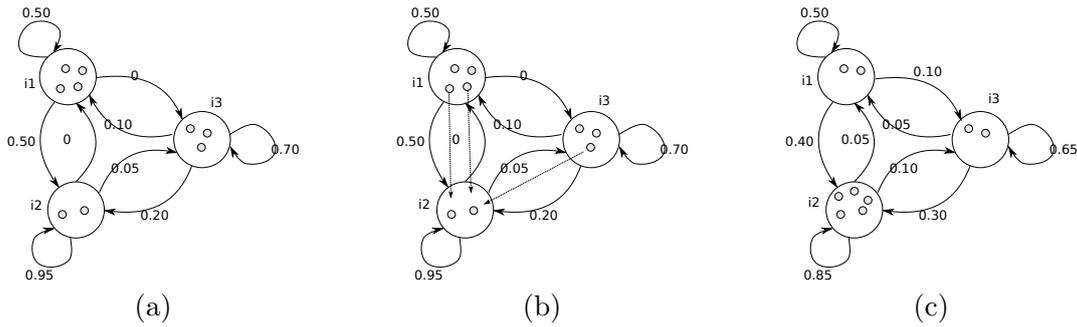


FIGURE 2 – Communications entre sous-populations : représentation d’un modèle en îles dynamiques (à trois îles).

- $p_{2 \rightarrow 2}$ diminue de 0.95 à 0.85 car les individus locaux n’améliorent pas la population ;
- $p_{3 \rightarrow 2}$ augmente de 0.20 à 0.30 car l’individu venant de l’île 3 a produit un descendant qui améliore l’évaluation de la population ;
- les autres $p_{i \rightarrow j}$ sont ajustées par normalisation.

Un mécanisme d’apprentissage plus fin, appliqué à la sélection automatique des opérateurs, sera présenté dans la section suivante.

Expérimentation

Afin de mesurer l’efficacité générale de ce premier schéma dynamique de modèle en îles (MID), nous avons comparé son comportement, dans le cadre d’un algorithme génétique, avec trois variantes statiques¹⁰ : un algorithme génétique classique (AG), un algorithme génétique parallèle (AGP), et un modèle en îles statique classique, avec une topologie en anneau unidirectionnel (MIS) produisant des migrations circulaires. AG peut être vu comme un modèle avec une seule île — tous les individus peuvent interagir à tout moment —, ou bien encore comme un modèle multi-îles où l’intervalle des migrations est minime et uniforme. De la même manière, AGP peut être assimilé à un modèle en îles ne réalisant aucune migration — les individus restent toujours sur leur île de départ, et ne peuvent interagir avec ceux des autres îles.

Les quatre variantes (AG, AGP, MIS, MID) ont été comparées avec un même jeu de paramètres par défaut. Tout d’abord, chaque île dispose du même algorithme pour l’évolution des populations. Celui-ci a été fixé en amont et est défini par les caractéristiques intra-îles du modèle. Le paramétrage du modèle en îles consiste alors à déterminer le nombre d’îles, d’individus et de croisements, ainsi que certains aspects concernant les migrations. Différentes expérimentations ont été effectuées à partir de paramétrages variés en termes de nombre d’îles et d’individus, donnant lieu à des résultats représentatifs et homogènes. Cette section s’attachant à observer l’effet d’une politique de migration dynamique, nous l’illustrons au moyen d’un paramétrage autorisant un nombre significatif d’individus et d’îles. Le nombre de croisements pour une île entre chaque migration est proportionnel à son nombre d’individus. Ce choix assure le même taux de croisements pour chaque île quelle que soit sa taille.

¹⁰Les variantes statiques sont simulables avec ce même modèle au moyen de paramétrages spécifiques.

L'expérimentation reportée ici respecte ainsi, pour toutes les variantes considérées, le paramétrage suivant :

1. Caractéristiques intra-iles :
 - type d'algorithme : steady state¹¹
 - élitisme : oui
 - sélection : tournoi
 - mutation : aléatoire sur les fils avec une probabilité d'application de 0.5
 - croisement : uniforme
2. Caractéristiques inter-iles :
 - nombre d'iles : 20 [AGP, MIS, MID]; 1 [AG]
 - nombre total d'individus : 600
 - répartition de départ : équilibrée (AGP, MIS, MID : 30 individus par ile)
 - nombre total de croisements : 216 000 (360×600 individus)
 - probabilités initiales de migration : $p_{i \rightarrow i} = 0.75, \forall i \in \{1, \dots, 20\}, p_{i \rightarrow j} = (1 - 0.75)/19, \forall j \neq i$
 - récompense : 5 points
 - pénalité : 5 points

Pour cette étude préliminaire, l'objectif restait de mesurer la pertinence d'un tel modèle et la capacité du modèle dynamique de gérer de manière autonome les migrations au cours de la recherche.

Nous reportons sur la table 1 les résultats comparatifs des quatre algorithmes génétiques sur 7 instances diverses de problèmes combinatoires, représentatives de résultats obtenus sur un plus large panel d'instances. Celles-ci proviennent de deux problèmes différents : sac à dos 0/1 et MAXSAT. Les quatre instances de sac à dos ont été générées en se basant sur la définition donnée par Chvátal [1980] et le générateur proposé par Martello *et al.* [1999] (avec respectivement 200, 500, 1000 et 2000 objets pour les instances reportées). Les deux premières instances MAXSAT sont des instances aléatoires comportant respectivement 600 et 1000 variables (2550 et 4250 clauses), la dernière étant un codage SAT d'une instance de carré latin (686 variables, 6816 clauses). Le critère d'efficacité retenu, qui reste celui sur lequel nous nous focalisons prioritairement tout au long de ce manuscrit, est celui de la qualité des solutions atteintes. Les valeurs reportées représentent la moyenne des valeurs objectif, sur 10 exécutions, des meilleurs individus obtenus (une valeur par exécution). Pour chaque couple (instance,algorithme), les mêmes 10 ensembles de 600 individus initiaux ont été utilisés. Les écarts-types associés, tous très faibles, ne sont pas reportés ici.

Cette courte comparaison expérimentale illustre l'intérêt de l'évolution dynamique des flux migratoires pour gérer les interactions entre les individus au sein d'un algorithme génétique, puisque son utilisation rend la convergence globale plus tardive et donc l'algorithme bien plus efficace. À l'inverse, le schéma de rotation classique (MIS) ne parvient pas toujours à obtenir des résultats compétitifs même par rapport à l'algorithme génétique classique AG. La raison principale provient probablement de la taille relativement petite des îles (20 individus) qui est adaptative au cours du processus de recherche pour MID, mais qui reste inchangée pour MIS. AGP est la version la moins performante; les sous-populations, plus petites, y convergent trop rapidement. Ces résultats sont cohérents avec

¹¹Les algorithmes génétiques dits *steady state* sont des algorithmes incrémentaux, par opposition aux algorithmes génétiques générationnels proposés initialement. Le premier algorithme génétique incrémental a été proposé par Whitley et Kauth [1988].

Instance	AG	AGP	MIS	MID
n200	301 675.8	301 421.5	302 503.1	302 528.7
n500	755 626.5	748 230.3	755 818.4	760 620.5
n1000	1 485 393.9	1 465 757.5	1 483 248.4	1 502 549.2
n2000	2 866 752.9	2 832 290.4	2 853 072.5	2 910 891.5
f600	2 513.8	2 357.2	2 518.7	2 533.4
f1000	4 174.2	3 890.5	4 174.1	4 208.9
qg1-07	6 756.3	6 211.7	6 776.5	6 787.0

TABLE 1 – Résultats comparatifs de quatre modèles d’algorithmes génétiques : standard (AG), parallèle (AGP), modèle en îles statique (MIS) et dynamique (MID).

les conclusions de Cantú-Paz et Goldberg [2003], qui ont montré qu’une simple exécution d’un algorithme génétique avec une large population était plus efficace que de multiples exécutions des mêmes algorithmes employant de plus petites populations.

Des analyses plus approfondies sont disponibles dans [Lardeux et Goëffon, 2010] et [Goëffon et Lardeux, 2011b]. Nous y mesurons en particulier l’effet des fréquences migratoires sur la convergence de la population.

2.3 Modèle raffiné et application à la sélection d’opérateurs

Concevoir un algorithme évolutionnaire en îles permet de considérer simultanément un ensemble de populations qui évoluent indépendamment durant certaines phases de la recherche tout en interagissant par moment. Ce modèle, qui constitue une abstraction supplémentaire par rapport aux algorithmes évolutionnaires classiques, permet non seulement de proposer plusieurs niveaux de diversification — il peut tolérer des convergences locales tout en préservant une diversité globale —, mais aussi de simplifier son traitement par une architecture parallèle.

Nous avons vu un moyen de réguler dynamiquement les migrations en considérant une matrice de transition entre les différentes îles, puis en renforçant ou diminuant les probabilités de migrations d’île en île durant le processus évolutionnaire en fonction de l’impact des précédentes migrations analogues. Le but d’une telle modélisation est d’auto-adapter les migrations sans schéma préétabli, de réguler dynamiquement le regroupement ou le cloisonnement d’individus en fonction de l’avancée de la recherche et par conséquent de moduler la taille des populations.

Dans un schéma classique de modèles en îles (modèle uniforme), les îles sont régies par les mêmes règles évolutionnistes, et sont ainsi uniquement distinguées et déterminées par les individus qui les composent. Le modèle permet alors de réguler l’interaction entre les individus ou les groupes d’individus. Nous proposons maintenant d’étendre ce modèle en attribuant à chaque île un mode opératoire particulier et indépendant, par exemple un algorithme ou un paramétrage d’algorithme spécifique. L’idée de base est qu’une régulation appropriée et autonome des flux migratoires affectera dynamiquement les ressources aux modes opératoires les plus pertinents en fonction de l’avancée de la recherche. Une telle modélisation peut alors servir de cadre de travail pour diverses applications comme la comparaison d’opérateurs, le paramétrage automatique ou la sélection adaptative des opérateurs.

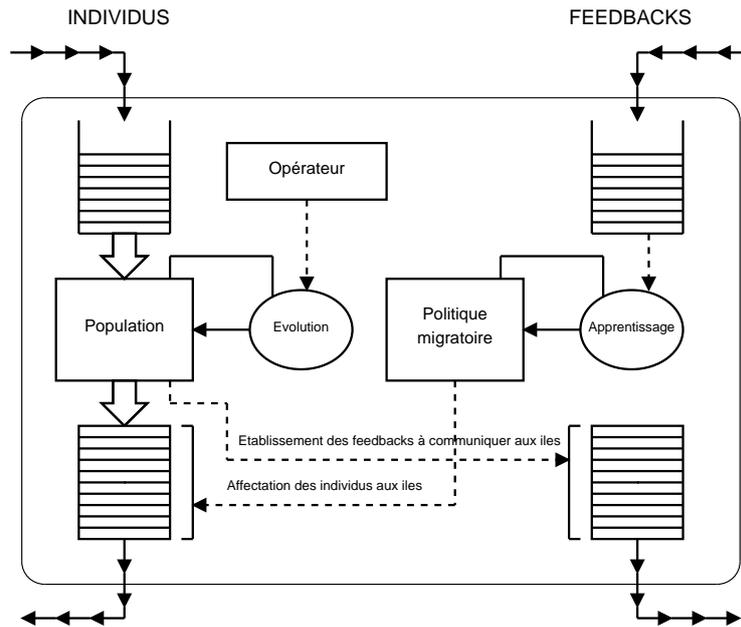


FIGURE 3 – Mécanisme opératoire autonome d'une île.

La section précédente présentait la version initiale de notre modèle en îles dynamique, qui constitue à notre connaissance la première contribution permettant de réguler dynamiquement les interactions entre les individus dans les algorithmes évolutionnaires à base de populations. Très rapidement, nous avons raffiné ce modèle sur deux points. Tout d'abord, une définition générique du modèle du point de vue de l'île, insistant sur le caractère autonome de chaque île. Une originalité est en effet de ne pas requérir de processus maître gérant l'ensemble des îles ; ceci dans l'optique d'un algorithme inspiré de la nature, où les politiques migratoires seraient laissées à l'appréciation des îles entre elles. Ensuite, une mise à jour des politiques migratoires utilisant un mécanisme de type apprentissage par renforcement, plus pertinent et dont le paramétrage s'avèrera simple et peu déterminant.

Précisons que plusieurs implémentations du modèle en îles dynamique ont été réalisées : fonctionnement séquentiel du modèle en intégralité (les îles sont considérées alternativement), fonctionnement séquentiel des îles (la réception des individus et des informations de feedback, l'évolution de la population, la mise à jour de la politique migratoire, la migration des individus et l'envoi des informations de feedback sont effectués séquentiellement, mais les îles évoluent parallèlement de manière synchrone ou asynchrone), ou parallèle (certaines tâches au sein de l'île sont effectuées simultanément). Le fonctionnement algorithmique général d'une île décrit sur la figure 3 illustre l'aspect parallèle et asynchrone du modèle. La partie gauche représente le traitement des individus.

Chaque île reçoit à tout moment des individus ou des ensembles d'individus, stockés dans une file. En fonction de ses conditions d'entrée, un processus évolutionnaire prend tout ou partie de la file d'individus, puis fait évoluer cette population au moyen d'un opérateur (ou algorithme). On affecte alors à chaque individu résultant une île destination en fonction de la politique migratoire. Les individus sont ensuite envoyés aux îles correspondantes. Ceux affectés à l'île courante retournent ainsi dans la file entrante en vue d'être traités

par l'opérateur évolutionnaire local.

Parallèlement, l'île reçoit des informations de feedback en provenance de l'ensemble des îles, qui rendent compte du devenir des individus ayant migré préalablement. Ces informations sont traitées par un mécanisme d'apprentissage qui met à jour la politique migratoire en renforçant les migrations fructueuses. La politique migratoire est un vecteur de transition M tel que M_j représente la probabilité pour un individu sortant de migrer vers l'île j . Des informations de feedback sont envoyées aux autres îles après chaque évolution de population. Il s'agit d'envoyer aux îles d'où proviennent les individus courants leur état actuel.

Ce fonctionnement général permet de nombreux choix conceptuels, au niveau du ou des opérateurs utilisés, mais également quant aux informations de feedback à transmettre et à la fonction d'apprentissage. Ces derniers éléments déterminent la stratégie d'évolution des politiques migratoires. La spécification du modèle est ainsi dépendante du type d'opérateur(s) en vigueur sur les différentes îles (mutation, croisement, recherche locale, *etc.*), des différents indicateurs à considérer au cours de la recherche (qualité des individus, entropie, effort computationnel), et de la stratégie générale à appliquer au modèle (gestion de la diversité, régulation des interactions entre individus, sélection d'opérateurs). Dans le reste de cette partie, nous nous intéressons à la sélection d'opérateurs, et particulièrement d'opérateurs unaires, s'appliquant à des individus uniques ou successivement et de manière indépendante à une population d'individus ; le seul critère retenu sera celui de qualité. La spécification d'une stratégie de sélection des opérateurs vise ainsi à déterminer les migrations favorisant l'amélioration des individus — et plus précisément à favoriser à plus long terme l'amélioration globale de la population — et se détermine par la nature des informations de feedback envoyées, et le traitement de celles reçues.

Individus → feedbacks. L'analyse de la population établit l'information à communiquer aux îles d'où proviennent les individus. Nous avons considéré l'amélioration moyenne (stratégie par défaut) ou maximale (stratégie alternative) de leurs individus après application de l'opérateur.

Feedbacks → mise à jour des politiques. Le regroupement des informations de feedback permet à chaque île de déterminer la pertinence des migrations, correspondant à l'utilité des opérateurs du modèle. Ces valeurs d'utilité ou de récompense permettent de mettre à jour la politique migratoire (vecteur de transition) au moyen d'un mécanisme d'apprentissage. On peut envisager plusieurs manières d'appliquer la prise en compte des feedbacks pour mettre à jour la politique migratoire. La première est de conserver des valeurs d'utilité mises à jour par apprentissage au moyen des informations de feedback, puis de déterminer une politique en fonction de ces valeurs d'utilité (par normalisation, ou selon diverses techniques de bandit comme celles détaillées dans le chapitre 4). Une seconde est de construire, à partir des feedbacks, un vecteur de récompense R à partir duquel le vecteur de transition sera mis à jour par apprentissage. Notre stratégie par défaut est basée sur cette méthode, et plus précisément ne récompense que la ou les îles les plus utiles :

$$R_j = \begin{cases} \frac{1}{|B|} & \text{si } j \in B, \\ 0 & \text{sinon,} \end{cases} \quad (1)$$

où B représente l'ensemble des îles ayant produit la meilleure amélioration d'après les informations de feedback compilées.

Les probabilités de migrations sont alors mises à jour au moyen d'un processus d'apprentissage par renforcement [Sutton et Barto, 1998] avec adjonction d'un paramètre de bruit :

$$M_j = (1 - \beta) \times (\alpha \times M_j + (1 - \alpha) \times R_j) + \beta \times N_i \quad (2)$$

N est un vecteur stochastique ($\sum_i N_i = 1$) aléatoire. α représente l'importance de la connaissance accumulée au cours des précédentes migrations (soit l'inertie du modèle), et β la quantité de bruit, nécessaire pour garder le modèle réactif (exploration). L'effet de ces paramètres sera mesuré dans le chapitre suivant. Nous pouvons simplement préciser que ce modèle permet de spécifier une stratégie de sélection d'opérateurs statique de type roulette prédéterminée, en initialisant sur toutes les îles un vecteur M identique (avec pour M_j la probabilité de sélection de l'opérateur j) et fixant les paramètres d'apprentissage aux valeurs qui rendent les vecteurs M inchangés, soit $\alpha = 1$ et $\beta = 0$.

Si l'architecture du modèle utilise des politiques migratoires indépendantes, celui-ci dans son intégralité est muni d'une politique générale simplement définie comme la combinaison de l'ensemble des politiques M . On note ainsi \mathcal{M} la matrice de transition telle que \mathcal{M}_{ij} décrit la probabilité pour un individu de l'île i , sur un intervalle de temps défini, de migrer vers l'île j . À l'unité temporelle près — les valeurs sont à normaliser si le modèle est asynchrone —, chaque ligne \mathcal{M}_i de la matrice de transition correspond au vecteur M décrivant la politique de l'île i .

Le chapitre 3 se recentre sur la sélection adaptative d'opérateurs. Nous y montrons au moyen d'un exemple d'application simple (maximisation d'une fonction One-Max) que le cadre du modèle en îles dynamique se prête particulièrement à la problématique de la sélection d'opérateurs, ouvrant de nouvelles pistes pour la conception d'algorithmes autonomes.

Chapitre 3

Sélection adaptative d'opérateurs : étude des fonctions One-Max

3.1 Problème de la sélection adaptative des opérateurs

La capacité d'un algorithme — en particulier évolutionnaire — à résoudre un problème ou une classe de problèmes d'optimisation est déterminée par sa suite d'actions (instructions), basée sur des informations que l'on peut distinguer en deux classes : connaissance *a priori* et connaissance apprise.

La connaissance *a priori* est l'information communiquée à l'algorithme et n'étant pas déterminée par l'effet de son application. Il s'agit des éléments conceptuels de l'algorithme, ce qui inclut son mécanisme opératoire, donc la représentation des solutions, l'ensemble des opérateurs applicables, et tout autre élément d'intervention non automatique. Plus généralement, il s'agit de l'algorithme lui-même, que l'on peut réduire à ses *possibilités* de résolution. Tout comme réduire le paramétrage, multiplier les opérateurs utilisables par l'algorithme, ainsi que les conditions d'application *a priori*, permet de moins restreindre son champ opératoire, et donc ses possibilités d'atteindre de bonnes solutions sur un plus grand nombre de problèmes ou d'instances de problèmes.

Le caractère autonome de l'algorithme regroupe les actions qui sont partiellement guidées par sa connaissance apprise. Ce second vecteur d'efficacité oriente la recherche parmi tous les chemins possibles dans le but de maximiser l'espérance d'atteindre les meilleures solutions. Si les théorèmes du *No Free Lunch* [Wolpert et Macready, 1997] confortent l'intuition qu'une solution algorithmique particulière — et notamment une stratégie de paramétrage particulière — ne pourra contourner les pièges de classes d'instances spécifiques, on peut imaginer qu'un mécanisme de construction algorithmique autonome guidé par le contexte pourra, de par ses plus larges possibilités initiales, mener à une efficacité et une robustesse générales accrues. Toute réduction du champ d'action de l'algorithme — comme l'instanciation de paramètres ou le choix d'opérateurs —, même justifiée, ne peut que diminuer son potentiel de résolution. Cette réduction est néanmoins indispensable ; d'ailleurs la spécification de l'algorithme lui-même en est une. La problématique sous-jacente à la conception d'un algorithme de recherche autonome est alors d'identifier l'information permettant de déterminer la politique générale de la recherche, et de spécifier

comment la connaissance générée par le processus lui-même peut être utilisée pour adapter et réorienter cette dernière.

L'accroissement des exigences computationnelles pour résoudre des instances de problèmes de plus en plus larges et difficiles fait qu'il devient illusoire de concevoir des algorithmes de recherche performants sur plusieurs types de problèmes et ne requérant pas de paramétrage spécifique. Le paramétrage d'algorithme est ainsi devenu un aspect extrêmement déterminant, en particulier concernant les métaheuristiques dont la généralité nécessite de déplacer les facteurs d'efficacité spécifiques au paramétrage. Deux types de paramétrages, éventuellement complémentaires, sont alors envisageables : le *réglage hors-ligne* et le *contrôle de paramètres*. Le premier peut être automatisé et entrer ainsi dans le cadre des composants d'algorithmes autonomes, par exemple au moyen d'hyperheuristiques en recherchant la meilleure configuration dans l'espace des paramètres, selon l'évaluation successive de différents paramétrages. Au contraire, le contrôle de paramètres consiste à régler ceux-ci dynamiquement au cours du processus en apprenant de la connaissance acquise en cours de recherche. Pour une vue plus précise des enjeux et des stratégies développées relatives à l'automatisation du réglage de paramètres, nous invitons le lecteur à se référer aux travaux de De Jong [2007], Eiben *et al.* [2007], Burke *et al.* [2010] et Hoos [2012], parmi d'autres.

Plus spécifiquement, nous nous intéressons dans la suite de cette partie à la problématique de la *sélection adaptative d'opérateurs*, qui consiste à déterminer à tout moment de la recherche l'action à effectuer la plus pertinente au regard des informations acquises au cours de la recherche. Bien sûr, une partie de ces actions devra être effectuée en guise d'évaluation permettant d'accroître l'espérance d'efficacité à moyen terme. D'autres actions seront effectuées car elles apparaissent comme les plus rentables sur le court terme. Ce problème de définir une stratégie de sélection efficace revient à répondre de manière adaptative au dilemme entre exploration et exploitation, étudié en particulier dans le cadre de problèmes de bandits et qui feront l'objet du chapitre suivant.

Dans ce chapitre, nous introduisons la sélection adaptative des opérateurs au moyen d'un exemple simple, le problème One-Max, et entrons plus directement dans le cadre scientifique de ce manuscrit, c'est-à-dire la recherche locale. One-Max est un problème combinatoire — la taille de l'espace de recherche est exponentielle — mais non NP-difficile et même trivial puisque la solution optimale est directement calculable. Son intérêt réside ici dans le fait qu'il est possible de déterminer théoriquement les opérateurs de recherche locale les plus appropriés à tout moment de la recherche. Cela nous permettra d'évaluer la pertinence du modèle en espace dynamique comme modèle d'abstraction approprié à la sélection adaptative d'opérateurs.

3.2 Le problème One-Max

3.2.1 Description du problème

Le problème One-Max¹² permet d'étudier certains aspects du comportement des algorithmes de recherche locale et génétiques au moyen d'une fonction objectif très simple. À toute solution d'un espace de recherche composé des chaînes de bits de longueur n fixée, la fonction objectif One-Max y associe son nombre de *uns*. Le problème d'optimisation se définit alors comme suit :

$$\operatorname{argmax}_{x \in \{0,1\}^n} \sum_{i=1}^n x_i \quad (3)$$

Le *score* d'un individu $x \in \{0,1\}^n$ sera aussi noté $|x|_1$. La solution optimale est connue, puisqu'il s'agit naturellement de la chaîne composée uniquement de *uns* ($x_i = 1, \forall i \in \{1, \dots, n\}$). Cette fonction, introduite nous semble-t-il par Ackley [1985], et d'autres variantes — fonction *Royal Road* [Mitchell *et al.*, 1992], fonction trompeuse *Trap* [Goldberg *et al.*, 1992] ou fonctions NK ajoutant l'interaction entre les bits [Kauffman et Weinberger, 1989] — sont régulièrement employées dans des travaux fondamentaux en algorithmique évolutionnaire [Mühlenbein, 1992; Mitchell *et al.*, 1993; Harik *et al.*, 1999; Droste *et al.*, 2002; Yang, 2003; Cheng et Kosorukoff, 2004; Lu *et al.*, 2011]. Ces fonctions de score permettent de disposer de plusieurs cadres pour simuler l'effet des croisements, mutations et mécanismes sélectifs sur l'évolution des individus. Dans la seconde partie de ce manuscrit, traitant des paysages de fitness, nous emploierons très largement les fonctions NK.

Dans notre présent problème, nous souhaitons utiliser différents types d'opérateurs de mutation affectant plus ou moins la solution courante, de manière à atteindre la solution optimale 1^n à partir de la solution opposée 0^n — chaîne composée uniquement de n zéros — en minimisant les applications d'opérateurs dont le nombre symbolise ici l'effort computationnel. Nous considérons des opérateurs de recherche locale stochastiques, effectuant des mutations aléatoires conservées seulement lorsqu'elles améliorent les individus (mécanisme de l'amélioration itérative de type *premier améliorant*). One-Max a été préalablement utilisé dans ce cadre de sélection adaptative des opérateurs afin d'évaluer les mécanismes de sélection [Fialho *et al.*, 2008; Derbel et Verel, 2011]. Nous avons repris les opérateurs utilisés dans ces études :

- *bit-flip*, qui flippe¹³ chaque bit avec une probabilité $1/n$;
- *k-flip* (avec $k = \{1, 3, 5\}$), qui flippe exactement k bits.

Par la suite et en fonction du contexte, *bit-flip*¹⁴ et *k-flip* pourront désigner l'opérateur de mutation ou bien la fonction de voisinage correspondante. *k-flip* se définit en effet par une fonction de voisinage $\mathcal{N}_k : \{0,1\}^n \rightarrow 2^{\{0,1\}^n}$, avec $x' \in \mathcal{N}_k(x) \leftrightarrow h(x, x') = k$

¹²Le terme *problème* s'applique ici dans le sens où l'on considère un espace de recherche combinatoire ($\{0,1\}^n$) associé à une fonction à optimiser (One-Max, équation 3). Il s'agit ainsi d'un problème combinatoire défini au moyen d'une fonction One-Max. Cependant, comme nous l'avons mentionné, ce dernier n'est un problème qu'en théorie puisque sa résolution est immédiate par raisonnement.

¹³*Flipper* un bit consiste à changer sa valeur via l'opérateur \neg .

¹⁴Dans la littérature, *bit-flip* désigne parfois l'opérateur 1-flip, et 1/ n -flip est alors utilisé pour désigner ce que nous appelons ici *bit-flip*.

et $h(x, x') = \sum_{i=1}^n |x_i - x'_i|$ représentant la distance de Hamming entre x et x' . Il est moins évident de décrire l'opérateur bit-flip à l'aide d'une fonction de voisinage : puisque toute transition est théoriquement possible, cela correspondrait à un voisinage complet avec une probabilité de mouvement non uniforme. Comme l'indique la proposition 1, un mouvement bit-flip peut être réduit à un mouvement k -flip avec une certaine probabilité de sélectionner chaque valeur de k .

Proposition 1 Appliquer l'opérateur bit-flip à un individu $x \in \{0, 1\}^n$ consiste à lui appliquer l'opérateur k -flip avec la probabilité $\binom{n}{k} \times \frac{1}{n^k} \times \left(1 - \frac{1}{n}\right)^{n-k}$.

En effet, puisque l'opérateur bit-flip flippe chaque bit avec une probabilité de $1/n$, la probabilité de flipper exactement k bits parmi n est $\binom{n}{k} \times \frac{1}{n^k} \times \left(1 - \frac{1}{n}\right)^{n-k}$ (distribution binomiale).

3.2.2 Calcul théorique des opérateurs les plus appropriés

Intuitivement, l'opérateur 5-flip devrait être plus pertinent lorsqu'il est appliqué à des individus moins bons en termes de score (comportant une grande majorité de zéros), alors que 1-flip pourra au contraire améliorer avec une plus grande probabilité que 5-flip les individus comportant une plus grande proportion de uns. Ce point est corroboré par la comparaison théorique des quatre opérateurs que nous avons proposée dans [Goëffon et Lardeux, 2011a] — puis détaillée davantage dans [Goëffon et Lardeux, 2011b] et [Candan et al., 2012] —, et que nous reprenons dans cette section.

En premier lieu, nous déterminons la probabilité d'améliorer un individu x de taille n et de score s avec un opérateur de mutation k -flip :

Proposition 2 Soit $x \in \{0, 1\}^n$ et $x' \in \mathcal{N}_k(x)$ deux chaînes de bits de scores $s = |x|_1$ et $s' = |x'|_1$. On a :

$$\mathbb{P}(s' > s) = \sum_{i=\max(\lfloor k/2 \rfloor + 1, k-s)}^{\min(k, n-s)} \frac{\binom{n-s}{i} \times \binom{s}{k-i}}{\binom{n}{k}} \quad (4)$$

Preuve :

Soit x une chaîne de bits de longueur n et par définition de l'opérateur k -flip : $\mathcal{N}_k(x) = \{x', h(x, x') = k\}$. Si $x' \in \mathcal{N}_k(x)$ et $|x'|_1 > |x|_1$, alors plus de zéros que de uns ont été flippés dans x . Soit $s = |x|_1$ le nombre de uns dans x . La probabilité de flipper exactement i zéros avec un opérateur k -flip ($0 \leq i \leq k$ et k fixé) est égale à la probabilité de sélectionner i zéros et $k - i$ uns, à savoir : $\binom{n-s}{i} \times \binom{s}{k-i} / \binom{n}{k}$. L'opération est améliorante si $i > k - i$, soit $i > k/2$, donc $i \geq \lfloor k/2 \rfloor + 1$. Par conséquent, la probabilité de sélectionner plus de zéros que de uns est la somme des probabilités de sélectionner exactement i zéros pour tout $i \geq \lfloor k/2 \rfloor + 1$ (opérations améliorantes) avec l'éventuelle restriction $k - s \leq i \leq n - s$ (opérations possibles parmi les améliorantes). □

La probabilité d'améliorer un individu x de taille n et de score s avec un opérateur de mutation bit-flip peut alors se déterminer immédiatement en utilisant la proposition 1.

Proposition 3 Soit $x, x' \in \{0, 1\}^n$ deux chaînes de bit de scores $s = |x|_1$ et $s' = |x'|_1$, telles que x' est générée par une application de l'opérateur bit-flip sur x . On a :

$$\mathbb{P}(s' > s) = \sum_{k=1}^n \left(\frac{1}{n^k} \times \left(1 - \frac{1}{n}\right)^{n-k} \times \sum_{i=\max(\lfloor k/2 \rfloor + 1, k-s)}^{\min(k, n-s)} \binom{n-s}{i} \times \binom{s}{k-i} \right) \quad (5)$$

Preuve :

La probabilité d'améliorer un individu en flipant exactement k bits est donnée par la proposition 2 et l'opérateur k -flip. Puisque tous les mouvements k -flip ($0 \leq k \leq n$) sont indépendants, la probabilité d'amélioration d'un mouvement bit-flip peut être calculée en ajoutant les probabilités d'amélioration pour chaque k , pondérées par les probabilités de flipper k bits avec un mouvement bit-flip (voir Proposition 1). Par conséquent,

$$\begin{aligned} \mathbb{P}(s' > s) &= \sum_{k=1}^n \left(\binom{n}{k} \times \frac{1}{n^k} \times \left(1 - \frac{1}{n}\right)^{n-k} \times \sum_{i=\lfloor k/2 \rfloor + 1}^k \frac{\binom{n-s}{i} \times \binom{s}{k-i}}{\binom{n}{k}} \right) \\ &= \sum_{k=1}^n \left(\frac{1}{n^k} \times \left(1 - \frac{1}{n}\right)^{n-k} \times \sum_{i=\lfloor k/2 \rfloor + 1}^k \binom{n-s}{i} \times \binom{s}{k-i} \right) \end{aligned}$$

avec comme restriction $k - s \leq i \leq n - s$.

□

Nous représentons sur la figure 4 l'évolution des probabilités d'amélioration de chacun des opérateurs en fonction du score des individus. On remarque que la probabilité d'améliorer un individu de score $n/2$ avec un opérateur k -flip est de 0.5 quel que soit k . C'est à partir de cette valeur que la probabilité d'amélioration décroît en fonction de k . Seulement, en cas de succès, la variation de score (*gain*) peut être plus élevée sur des valeurs de k supérieures. Cette probabilité d'amélioration n'est ainsi pas suffisante pour déterminer les opérateurs les plus appropriés, nous amenant à considérer l'espérance de gain des différentes opérations.

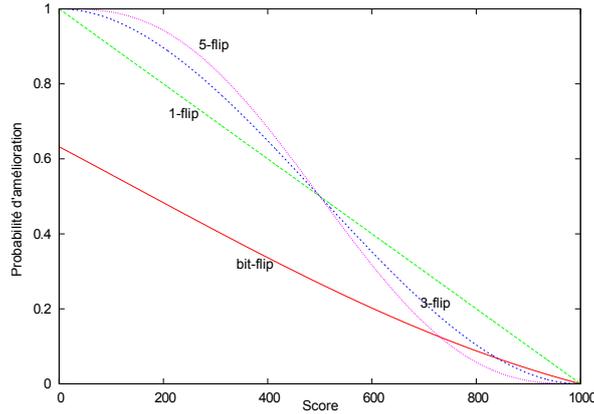


FIGURE 4 – Probabilité d'améliorer un individu avec les opérateurs bit-flip, 1-flip, 3-flip et 5-flip (cas du problème One-Max de taille $n = 1000$).

Au cours d'une recherche locale, seules les mutations (opérations) améliorantes sont

effectives ; les mutations détériorantes étant annulées. En conséquence, le gain d'une mutation $x \rightarrow x'$ (x' résulte d'une mutation appliquée à x) peut être formalisé par :

$$\Delta s(x, x') = \max(0, |x'|_1 - |x|_1) \quad (6)$$

Proposition 4 *L'espérance de gain d'une mutation k -flip sur un individu $x \in \{0, 1\}^n$ de score $s = |x|_1$ est égale à :*

$$\sum_{i=\max(\lfloor k/2 \rfloor + 1, k-s)}^{\min(k, n-s)} (2i - k) \times \frac{\binom{n-s}{i} \times \binom{s}{k-i}}{\binom{n}{k}} \quad (7)$$

Preuve :

Soit $x \in \{0, 1\}^n$, $x' \in \mathcal{N}_k(x)$, $s = |x|_1$, $s' = |x'|_1$. Soit i le nombre de zéros flippés lors d'une mutation k -flip $x \rightarrow x'$. Le nombre de *uns* dans x' (s') est égal au nombre de *uns* dans x (s), moins le nombre de *uns* flippés ($k - i$), plus le nombre de zéros flippés (i). $s' = s - (k - i) + i = s - k + 2i$, soit $s' - s = 2i - k$. Si $i \leq \lfloor k/2 \rfloor$, alors $s' \leq s$, et $\Delta s(x, x') = 0$. Sinon, $\Delta s(x, x') = 2i - k$. L'espérance de $\Delta s(x, x')$ se déduit alors de la proposition 2 et sa preuve. □

Proposition 5 *L'espérance de gain d'une mutation bit-flip sur un individu $x \in \{0, 1\}^n$ de score $s = |x|_1$ est égale à :*

$$\sum_{k=1}^n \left(\frac{1}{n^k} \times \left(1 - \frac{1}{n}\right)^{n-k} \times \sum_{i=\max(\lfloor k/2 \rfloor + 1, k-s)}^{\min(k, n-s)} (2i - k) \times \binom{n-s}{i} \times \binom{s}{k-i} \right) \quad (8)$$

Preuve :

Ce résultat se déduit des trois précédentes propositions. □

La figure 5 représente les espérances de gain des applications des différents opérateurs, en fonction du score des individus. Cela nous permet de comparer le gain moyen apporté par chaque opérateur sur un individu en fonction de son score ; en particulier l'impact du 5-flip en début de recherche, mais également le fait que le 5-flip est toujours le meilleur opérateur peu après $n/2$ alors que la probabilité d'amélioration est strictement favorable au 1-flip pour tout $s > n/2$.

Enfin, nous avons défini le taux de dominance χ comme mesure de pertinence d'un opérateur sur un ensemble d'opérateurs (voir définition 2). Ce taux représente la probabilité qu'un opérateur spécifique ait la plus grande espérance de gain relativement aux autres opérateurs applicables.

Définition 2 *Soit $x \in \{0, 1\}^n$ un individu de score s , $M = \{\mu_1, \dots, \mu_l\}$ un ensemble d'opérateurs de mutation et x_1, \dots, x_l l voisins de x tels que $x \xrightarrow{\mu_k} x_k$ (x_k résulte de la mutation de x par μ_k). s_1, \dots, s_l sont les scores respectifs des individus x_1, \dots, x_l . On*

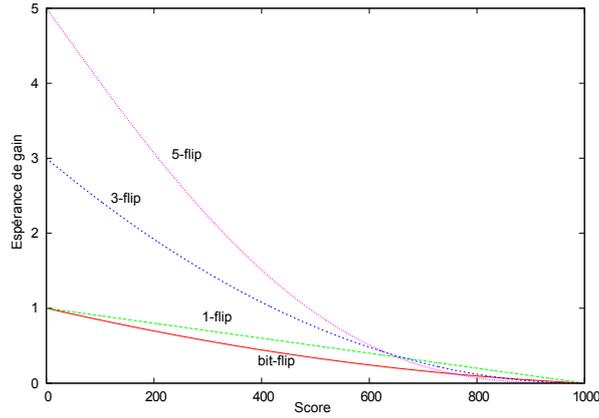


FIGURE 5 – Évolution de l'espérance de gain (cas $n = 1000$).

défini le taux de dominance $\chi(\mu_k, M, x)$ d'un opérateur μ_k sur M relativement à x comme la probabilité $\mathbb{P}(\Delta s(x, x_k) > \Delta s(x, x_{k'})), \forall k' \in \{1, \dots, l\} \setminus \{k\}$, égale à :

$$\sum_{i=1}^n \left(\mathbb{P}(\Delta s(x, x_k) = i) \times \prod_{\substack{k'=1 \\ k' \neq k}}^l \sum_{j=0}^{i-1} \mathbb{P}(\Delta s(x, x_{k'}) = j) \right) \quad (9)$$

Pour les opérateurs k -flip (x' , de score s' , est la transformation de x , de score s , par k -flip), on a (voir Proposition 2) :

$$\mathbb{P}_k(\Delta s(x, x') = i) = \begin{cases} \frac{\binom{n-s}{i} \times \binom{s}{k-i}}{\binom{n}{k}} & \text{si } i > 0, \\ 1 - \mathbb{P}(s' > s) & \text{si } i = 0. \end{cases} \quad (10)$$

Pour l'opérateur bit-flip, $\mathbb{P}_{\text{bit}}(\Delta s(x, x') = i) =$

$$\sum_{k=1}^n \left(\binom{n}{k} \times \frac{1}{n^k} \times \left(1 - \frac{1}{n}\right)^{n-k} \times \mathbb{P}_k(\Delta s(x, x') = i) \right) \quad (11)$$

L'évolution des taux de dominance des quatre opérateurs considérés relativement au score des individus est indiquée sur la figure 6 (avec $M = \{1\text{-flip}, 3\text{-flip}, 5\text{-flip}, \text{bit-flip}\}$). Il s'agit de repérer, pour le problème One-Max de longueur 1000 et en fonction du score d'un individu (en abscisse), quelle est la probabilité pour qu'une application aléatoire de l'un des quatre opérateurs considérés soit le meilleur choix en termes d'espérance de gain.

Dans un contexte de recherche locale appliquée à partir de la configuration 0^n , l'opérateur 5-flip est clairement le plus approprié en début de recherche et plus précisément lorsque $s \leq 601$. 3-flip est théoriquement le meilleur opérateur en termes d'espérance de gain lorsque $602 \leq s \leq 616$, mais en pratique les trois opérateurs ont une efficacité quasiment équivalente lorsque s est proche de 610. Ensuite, l'opérateur 1-flip est celui qui garantit la meilleure espérance de gain relativement aux autres opérateurs, même si l'espérance effective baisse lorsque s croît. De manière analogue au problème du collectionneur de

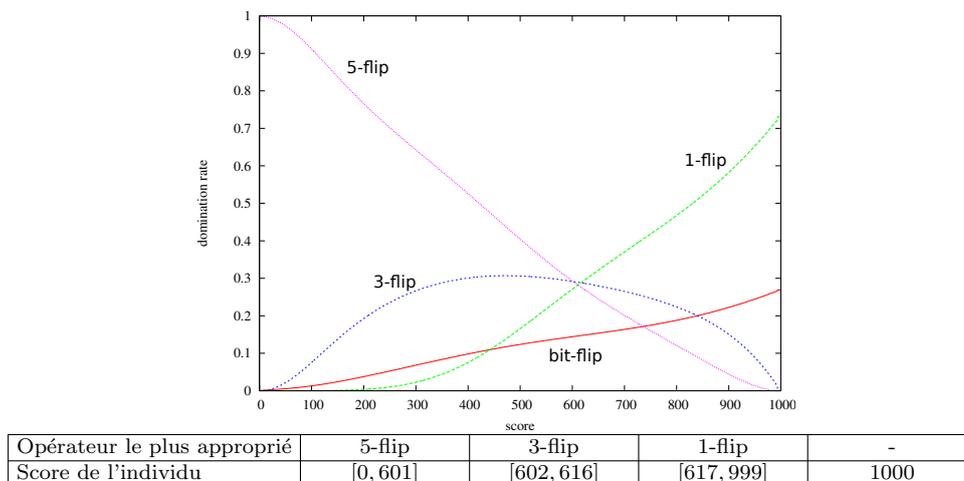


FIGURE 6 – Évolution des taux de dominance ($n = 1000$).

vignettes [Erdős et Rényi, 1961], il est bien moins probable d’augmenter le score dans les derniers pas de la recherche ; celle-ci se concentre ainsi majoritairement autour d’individus de scores élevés. Au final l’opérateur 1-flip est celui qui est le plus longtemps le plus approprié, et reste le plus approprié en moyenne au cours d’une recherche complète.

En début de recherche, l’opérateur bit-flip peut, avec une très faible probabilité, être plus performant que les trois autres opérateurs en étant notamment le seul capable de flipper au moins 6 zéros de plus que de *uns*. En théorie, il est même capable de résoudre le problème en une seule opération depuis toute configuration, mais avec une probabilité infinitésimale dans le cas général. Néanmoins il reste systématiquement dominé, en termes d’espérance de gain, par au moins un autre opérateur. Rappelons que bit-flip est un opérateur de mutation sans effet dans plus d’un tiers des applications, avec une probabilité d’effectuer un mouvement 0-flip égale à $(\frac{999}{1000})^{1000} \approx 0.3677$. Précisons enfin que les taux de dominance ne sont pas directement déterminés par le rapport s/n ; par exemple, considérant la fonction One-Max avec $n = 100$, 3-flip domine les trois autres opérateurs dans l’intervalle [63, 65].

3.3 Modèle en îles dynamiques pour la sélection adaptative d’opérateurs

3.3.1 Optimisation d’une fonction One-Max

Dans la section précédente, nous avons identifié de manière théorique l’opérateur idoine à appliquer à un individu en fonction de son score, afin d’optimiser une fonction One-Max. Optimiser le plus efficacement (en moyenne) cette fonction au moyen des quatre opérateurs de recherche locale définis précédemment consiste à appliquer à l’individu courant l’opérateur ayant le taux de dominance maximal en fonction de son score. Lorsque $n = 1000$, ces opérateurs sont indiqués dans le tableau joint à la figure 6.

Dans cette section, nous utilisons le modèle en îles défini dans le chapitre précédent —

dans sa version intégralement séquentielle — comme cadre algorithmique pour la sélection adaptative des opérateurs. À chaque île est affecté un opérateur différent ; par conséquent les tests ont été réalisés avec un modèle à 4 îles. Le mécanisme d’apprentissage et la teneur des informations de feedback respectent les spécifications par défaut mentionnées à la fin du chapitre précédent. Le reste du paramétrage, dont les éventuels effets seront discutés, est réalisé de la manière suivante :

- on initialise la population totale à 400 individus 0^n uniformément répartis ;
- les politiques initiales ne prévoient pas de migration (sur chaque île i , $M_i = 1$ et $M_j = 0, \forall j \neq i$) ;
- les paramètres d’apprentissage sont fixés à $\alpha = 0.8$ et $\beta = 0.1$.

La figure 7.a représente l’évolution de la taille des sous-populations correspondant aux différentes îles (bit-flip, 1-flip, 3-flip et 5-flip), le score moyen de la population totale (MOY) et celui du meilleur individu de la population (MAX) en fonction du nombre d’itérations ou migrations. Puisque l’opérateur évolutionnaire consiste à appliquer une fois l’opérateur à chaque individu, le nombre total d’opérations appliquées à chaque individu correspond au nombre de migrations. De par la définition des opérateurs, qui ne valident que les mutations bénéfiques, le score de chaque individu ne peut diminuer. Nous pouvons remarquer que le meilleur score est toujours proche du score moyen de la population, ce qui indique que tous les individus reçoivent des possibilités d’amélioration équivalentes au cours de la recherche.

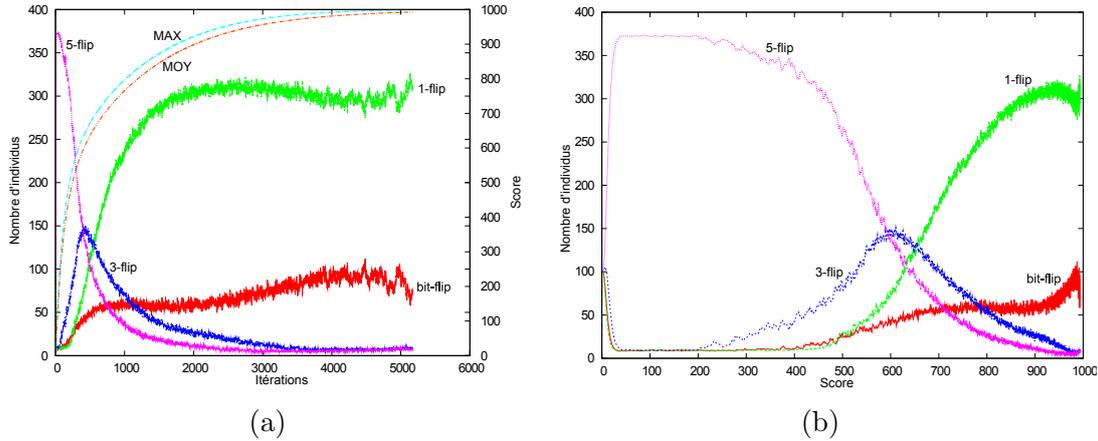


FIGURE 7 – Taille des îles (utilisation des opérateurs), et évolution du score des individus au cours de la recherche (a). Évolution de la taille des îles en fonction du score moyen des individus (b).

Ici, l’usage de chaque opérateur est indiqué par la taille des îles, qui correspond au taux d’application de chaque opérateur au cours de la recherche. Afin de comparer ces résultats expérimentaux aux valeurs théoriques, nous représentons figure 7.b les mêmes valeurs mais en fonction du score moyen des individus de la population. On constate à quel point le modèle en îles parvient à attirer de manière autonome les individus vers les opérateurs les plus pertinents au cours de la recherche lorsque l’on compare la figure 7.b à la figure 6, qui indique les taux de dominance théoriques. En outre, le fait que les opérateurs préférentiels aient un taux d’application supérieur à leur taux de dominance est naturel dans le contexte de maximiser l’espérance de gain. Une information supplémentaire

primordiale, qui n'apparaît pas sur cette figure, est que toutes les îles voient leurs politiques de migration quasiment synchronisées. En effet, en analysant l'évolution de la politique générale du modèle via la matrice de transition \mathcal{M} (voir chapitre précédent), il apparaît qu'à tout moment de la recherche à l'exception des toutes premières itérations, celle-ci est composée de lignes toutes très semblables. La politique générale du modèle évolue ainsi à la manière d'une roulette adaptative où l'opérateur à appliquer à chaque individu est fonction du vecteur de probabilité engendré par une ligne de la matrice. Comme nous le verrons dans le chapitre suivant, des probabilités de migrations conditionnées par les îles peuvent apparaître dès lors que l'application de séquences d'opérateurs devient pertinente. Dans ce cas, les lignes de la matrice \mathcal{M} pourraient être désynchronisées.

Afin d'apprécier l'apport de cette sélection adaptative des opérateurs, nous avons comparé la vitesse de résolution du modèle dynamique utilisant les quatre opérateurs à celles de cinq variantes statiques : quatre utilisant un seul des opérateurs disponibles (1-flip, 3-flip, 5-flip, bit-flip), ainsi qu'une stratégie de sélection uniforme des quatre opérateurs (roulette uniforme). Les spécifications et paramétrages du modèle en îles sont inchangés ; les cinq autres algorithmes utilisent également 400 individus initialisés à la configuration 0^n et améliorés par recherche locale à raison d'une évaluation pour chaque individu, à chaque itération.

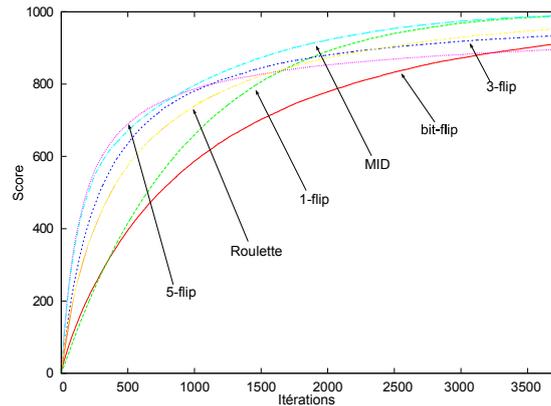


FIGURE 8 – Évaluation comparée du score du meilleur individu de la population pour six algorithmes : recherches locales mono-opérateur (bit-flip, 1-flip, 3-flip, 5-flip) et multi-opérateur (roulette et MID). Les scores reportés sont calculés en moyennant les résultats de 100 exécutions pour chaque algorithme.

La figure 8 reporte, pour chaque algorithme, le meilleur score de la population après chaque itération. Tout d'abord, nous pouvons vérifier empiriquement qu'aucun algorithme de recherche locale n'utilisant qu'un seul opérateur ne domine les trois autres à tout moment de la recherche. Une recherche 5-flip est logiquement la plus performante en début de recherche, mais est par la suite dominée par une recherche intégralement 3-flip, elle-même rapidement moins performante qu'une recherche n'utilisant que l'opérateur 1-flip, qui reste au final l'opérateur à privilégier pour une plus grande performance globale. Un point notable est que la roulette uniforme est à tout moment de la recherche battue par l'une au moins des trois stratégies fixes. Au contraire, le modèle en îles dynamique domine chaque variante statique et trouve plus rapidement des configurations de bonne

qualité, bien que rejoint en fin de recherche, lorsque celle-ci converge, par une recherche locale 1-flip. Rappelons que le modèle en îles ne requiert ni prétraitement ni connaissance préliminaire sur les opérateurs.

Cette validation au moyen d’une fonction One-Max est facilitée par la comparaison possible avec des valeurs d’utilités calculées théoriquement. D’autres validations expérimentales, que nous ne détaillerons pas dans ce manuscrit, ont été effectuées au moyen de fonctions NK et d’instances du problème du voyageur de commerce (TSP) utilisant plusieurs opérateurs de voisinage, avec pour référence indicative les performances d’algorithmes de réglage automatique statique des paramètres comme ParamILS [Hutter *et al.*, 2009] ou REVAC [Nannen *et al.*, 2008] — certes désavantagés sur ce type d’applications.

3.3.2 Influence des paramètres

Même s’il existe différentes manières de spécifier un modèle en îles dynamique (voir chapitre précédent), ce dernier requiert finalement peu de paramètres : nombre d’individus, distribution initiale des individus et vecteurs de transition initiaux, valeurs de α et β . Nous verrons dans cette section que leurs effets, notamment en ce qui concerne l’initialisation du modèle, sont très réduits.

Nombre d’individus

Nous avons observé expérimentalement, dans le contexte de cette étude sur le problème One-Max, que le comportement du modèle était presque identique avec une petite population d’une vingtaine d’individus et une large population d’un millier d’individus. Avec davantage d’individus, le mécanisme est néanmoins plus robuste car moins sensible au bruit. Avec une petite population, il est nécessaire d’étendre le nombre de pas, mais le nombre d’opérations total est plus faible et le traitement des informations plus rapide. Dans notre application présente, une taille de population de 400 ($100 \times$ nombre d’îles) permet, en des temps de calcul raisonnables, d’inférer une affectation des opérateurs très proche du schéma théorique optimal. Pour des usages de résolution pratiques, étant donné que le modèle dynamique est suffisamment réactif, de plus petites tailles de population produisent des résultats équivalents en termes de qualité, même si le comportement général du modèle sera légèrement plus chaotique (voir figure 9).

Politique initiale

On peut naturellement imaginer deux politiques migratoires de base sur chaque île afin de déterminer un état initial au modèle. La première consiste à partitionner initialement les îles en laissant l’exploration débloquent les premières migrations (stratégie utilisée ci-avant) ; la seconde initialise toutes les trajectoires d’individus de manière équiprobable et laisse l’exploitation partitionner les îles si besoin. Des stratégies intermédiaires fixant la probabilité de rester sur chaque île et distribuant uniformément les autres probabilités de migration peuvent être envisagées, de même qu’une initialisation aléatoire des vecteurs de transition.

Les deux stratégies de base étant les plus extrêmes, nous avons comparé leur impact respectif sur le comportement de l’algorithme. La seule différence réside dans les premières

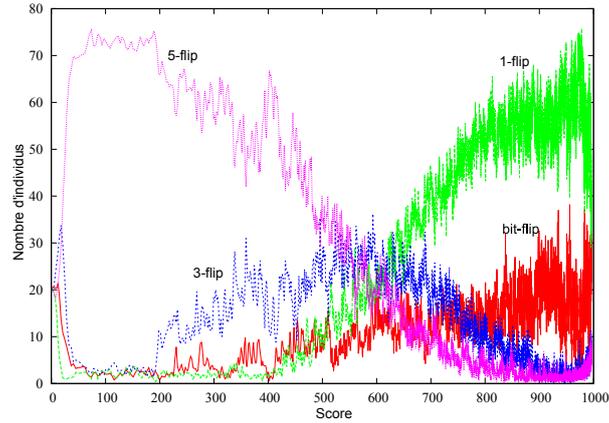


FIGURE 9 – Évolution de la taille des îles en fonction du score moyen des individus (cas d’une population de 80 individus).

itérations, mais le modèle se stabilise rapidement (à condition que β ne soit pas nul ou trop faible en regard de la taille de population utilisée). La figure 10 montre l’évolution des probabilités de rester sur chacune des îles avec les deux stratégies d’initialisation. On constate qu’avec les paramètres de base, les valeurs coïncident en moins de 30 épisodes migratoires ; ceci confirme que la capacité réactive du modèle est suffisante pour que l’initialisation des vecteurs de transition ne détermine pas son efficacité générale.

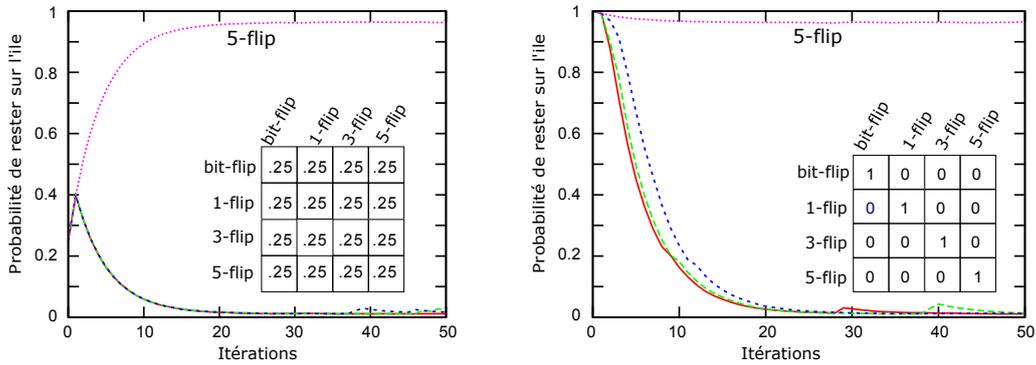


FIGURE 10 – Évolution des probabilités de rester sur une même île lorsque celles-ci sont initialisées à 0 (à gauche) ou à 1 (à droite).

Paramètres d’apprentissage

Les valeurs par défaut de α et β sont fixées respectivement à 0.8 et 0.01 ; ces valeurs ont été déterminées au moyen de tests préliminaires sur différents types d’applications, et semblent fournir un équilibre approprié entre exploration et exploitation. Précisons toutefois que, de par le fonctionnement du mécanisme d’apprentissage, ces valeurs doivent être en adéquation avec d’autres choix conceptuels : l’influence de α dépend naturellement de la fréquence de mise à jour de la politique de migration, et la prise en compte de β dépend du nombre d’individus visitant l’île. Cependant, tout paramétrage égal par ailleurs,

une valeur trop élevée de α va ralentir la recherche puisque l'information obtenue par les migrations récentes sera moins prise en compte dans l'évolution de la politique. À l'opposé, une valeur trop faible de α minimisera l'influence de la connaissance établie et surestimera l'effet des dernières migrations. Dans ce dernier cas la recherche sera plus sensible aux aléas stochastiques des opérateurs. Le rôle de β est déterminant, mais ce paramètre n'a pas besoin d'être réglé finement pour un fonctionnement satisfaisant de l'algorithme. Une trop grande valeur ralentira cependant la recherche en autorisant de plus nombreuses migrations peu pertinentes. Le réglage $\beta = 0$ peut en revanche conduire à rendre des îles inatteignables. Nous pouvons visualiser sur les figures 11 et 12 l'effet des paramètres α et β sur le comportement de l'algorithme.

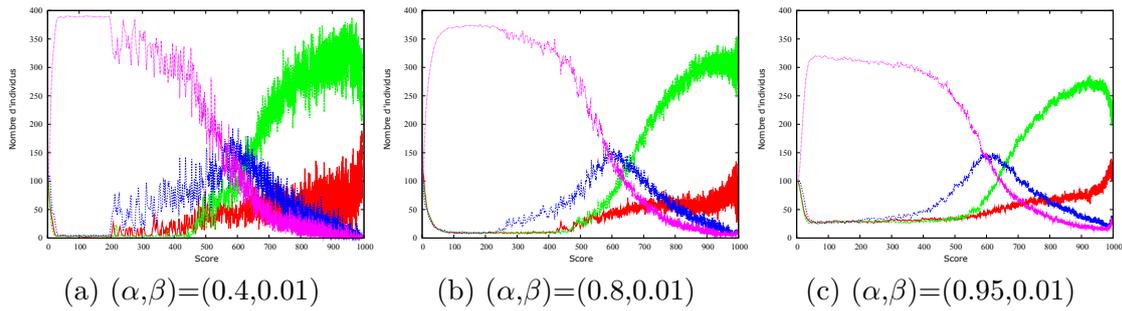


FIGURE 11 – Influence de α : davantage d'inertie rend le modèle plus stable mais ne modifie pas la distribution globale des individus sur les opérateurs.

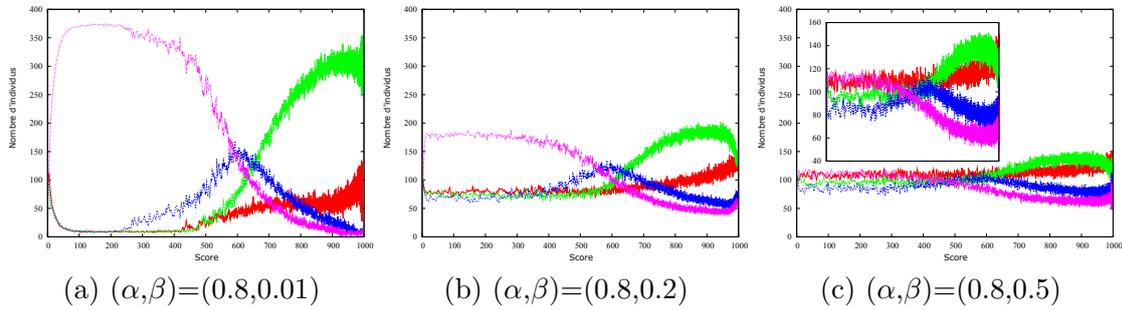


FIGURE 12 – Influence de β : davantage de bruit rend la distribution des individus entre les îles plus uniforme.

Chapitre 4

Bandits comme modèle de sélection adaptative des opérateurs

Nous avons évoqué section 3.1 la problématique de la sélection adaptative des opérateurs, c'est-à-dire de définir une politique de sélection d'opérateurs qui s'ajuste en fonction des informations de feedback collectées au cours du processus. Plusieurs algorithmes de sélection adaptative d'opérateurs évolutionnaires furent développés dans les années 1990, faisant évoluer leurs probabilités de sélection en fonction de leur capacité observée à engendrer de bons individus [Davis, 1989; Srinivas et Patnaik, 1994; Julstrom, 1995; Tuson et Ross, 1998; Barbosa et Sá, 2000]. Dans les deux derniers travaux cités, la stratégie relève de la méthode du *probability matching* [Goldberg, 1990] en faisant évoluer des probabilités de sélection d'opérateurs à partir des bénéfices observés au moyen d'un mécanisme d'apprentissage par renforcement.

Des travaux plus récents relatifs au développement de stratégies de sélection adaptative d'opérateurs utilisent comme cadre applicatif le modèle du bandit à plusieurs bras [Da Costa *et al.*, 2008; Fialho *et al.*, 2009; Maturana *et al.*, 2009]. Celui-ci y est utilisé comme modèle d'abstraction pour simplifier et unifier la problématique de la sélection d'opérateurs. Dans ce chapitre, nous partons de cette base de travail, puis proposons une extension des modèles de bandits non stationnaires classiques s'inspirant de l'évolution du potentiel des opérateurs dans les algorithmes évolutionnaires.

4.1 Problème du bandit multi-bras

Le problème du bandit multi-bras formalise la problématique du joueur cherchant à maximiser son gain face à un ensemble de machines à sous (bandits manchots) dont les récompenses sont déterminées par des distributions de probabilités propres à chaque machine, mais inconnues. À partir des récompenses acquises précédemment en fonction des différentes actions, le problème consiste à déterminer les bras de quelles machines actionner de manière à maximiser ses futures récompenses.

On peut attribuer les premiers travaux relatifs à cette problématique à Thompson [1933], qui cherche à déterminer la vraisemblance qu'une probabilité inconnue soit supérieure à une autre à partir de l'observation de deux échantillons dont les événements sont

respectivement associés à l'une et l'autre des probabilités. Dans l'exemple pris par l'auteur, deux traitements distincts sont appliqués respectivement à deux échantillons d'individus. En fonction de la taille des échantillons et des taux de succès de chaque traitement, il s'agit de calculer la vraisemblance que le premier traitement soit meilleur que le second, afin de déterminer la proportion d'individus à soigner avec le traitement semblant être le meilleur. Thompson précise que l'espérance de sauver un maximum d'individus est augmentée par le sacrifice hypothétique de soigner une certaine proportion d'individus avec le traitement semblant être, relativement à l'observation, le moins efficace. Plus généralement, l'*échantillonnage de Thompson* consiste à choisir l'action à réaliser en fonction de la probabilité qu'il s'agisse de la meilleure.

Robbins [1952] formule la problématique du bandit à partir de pièces. On présente à un joueur deux pièces pipées A et B , dont les probabilités respectives de tomber sur *pile* sont inconnues. Celui-ci dispose de T lancers de pièces et doit maximiser son gain, correspondant au nombre de *pile* obtenus. Pour maximiser son gain potentiel, le joueur a clairement intérêt à utiliser l'information des lancers précédents pour adapter sa stratégie de sélection de la pièce à lancer à chaque coup. Tout comme avec la problématique des patients à soigner, cette formulation illustre clairement le dilemme entre exploration et exploitation ; exploiter l'information consiste à lancer la pièce ayant entraîné la plus grande proportion de cas favorables, tandis qu'explorer permet d'affiner l'estimation de l'autre pièce.

Le problème du bandit à deux bras a été étendu en problème multi-bras donnant lieu à de nombreuses définitions [Rodman, 1978; Kelly, 1981; Herkenrath, 1983], selon que les bras sont affectés à des lois de probabilité réelles ou binomiales, ou sont uniquement décrits par des séquences de récompenses. Dans la suite, on désigne un bras par un ordinal ; $\mathcal{B} = \{1, \dots, B\}$ représente un ensemble de B actions possibles dans un problème de bandit à B bras. Nous partons ici d'un modèle de bandit simple, où chaque activation du bras $b \in \mathcal{B}$ apporte une récompense $r_b \in [0, 1]$ avec une probabilité $p_b \in]0, 1]$ (et 0 sinon). Dans la version stationnaire où les p_b et r_b sont inchangés au cours du processus, une instance de bandit se définit par un ensemble de B couples (p_b, r_b) . Un cas particulier impliquant des distributions binomiales consiste à fixer tous les r_b à 1. Notons que restreindre l'espace des lois de probabilités envisageables ne réduit pas la difficulté du problème, car le nombre d'observations nécessaires permettant d'inférer l'espérance d'une distribution de probabilités avec un certain niveau de vraisemblance n'est pas dépendant en général du type de loi de probabilité. Dans cette formalisation, les bras et les actions sont indépendants, à savoir que les récompenses ne dépendent ni des paramètres des autres bras, ni de l'historique des actions.

La problématique à horizon fini — qui nous intéresse en particulier — est de déterminer, pour un horizon T représentant le nombre d'actions à effectuer, la séquence de bras $(x_1, \dots, x_T) \in \mathcal{B}^T$ à actionner, permettant de maximiser le gain G égal à la somme des récompenses acquises R_t . Une stratégie de bandit détermine chaque action x_t en fonction de la connaissance acquise de la séquence d'actions et récompenses associées $((x_1, R_1), \dots, (x_{t-1}, R_{t-1}))$. Dans notre modèle, R_t vaut r_{x_t} avec une probabilité p_{x_t} , et 0 sinon.

Soit $Q_t^*(b)$ la *vraie valeur d'action* du bras $b \in \mathcal{B}$ à l'itération t . Il s'agit de l'espérance réelle de récompense de l'action b . La *valeur optimale* $V_t^* = \max_b Q_t^*(b)$ est l'espérance maximale de l'action effectuée à l'itération t . Le *regret* l_t de l'itération t est la perte,

en termes d'espérance, de l'action effectuée par rapport à l'action de valeur optimale : $l_t = \mathbb{E}[V_t^* - Q_t^*(x_t)]$. Le *regret total* \mathcal{L}_T d'une stratégie cumule l'ensemble des regrets : $\mathcal{L}_T = \sum_{t=1}^T l_t$. Dans un contexte de bandit stationnaire, les vraies valeurs d'action sont inchangées au cours du temps ; les $Q^*(b)$ et V^* ne sont alors pas dépendantes de t . Ainsi $Q^*(b) = \mathbb{E}[R_t | x_t = b] = p_b \times r_b$, et $\mathcal{L}_T = T \times V^* - \sum_{t=1}^T Q^*(x_t)$. Comme les vraies valeurs d'action sont inconnues, les stratégies de bandit les estiment en fonction des itérations précédentes (*historique*). On note alors $\hat{Q}_t(b)$ la *valeur d'action estimée* du bras b à l'itération t , c'est-à-dire avec comme connaissance la séquence $((x_1, R_1), \dots, (x_{t-1}, R_{t-1}))$. Ces valeurs d'action estimées correspondent aux utilités attendues de chacun des bras. Une stratégie de bandit se compose ainsi de deux éléments : l'estimation des valeurs d'action, et la stratégie de sélection d'action à partir des valeurs d'action estimées $\hat{Q}_t(b)$. À chaque itération t , $\operatorname{argmax}_b \hat{Q}_t(b)$ représente l'ensemble des meilleures actions empiriques, tandis que $\operatorname{argmax}_b Q_t^*(b)$ est l'ensemble des meilleures actions théoriques (généralement une unique action). Un certain nombre de stratégies de bandit sont présentées section 4.2. L'estimation des valeurs d'action, dont la problématique est commune à toutes les stratégies, est discutée lors de l'évocation de la première d'entre elles (gloutonne).

Traditionnellement, l'efficacité d'une stratégie de bandit à horizon fini se mesure à son regret total \mathcal{L}_T , qui représente la différence entre son espérance de gain et celle de la stratégie optimale. Il peut également être envisagé un regret effectif $V^* - G$ à partir du gain effectivement obtenu — ce regret pouvant être négatif — voire un regret optimiste $T \times \max_{b \in \mathcal{B}} r_b - G$ comparant le gain obtenu avec le plus grand gain envisageable. À horizon infini, l'efficacité d'une stratégie se mesure à l'évolution du regret total \mathcal{L}_T lorsque T tend vers l'infini.

Une stratégie — ou politique — de bandit se définit le plus souvent comme le choix du bras à activer en fonction d'un ensemble d'actions passées et leurs récompenses associées, sans référence explicite à une séquence optimale de choix à effectuer. C'est pourquoi la plupart des formulations d'une instance de bandit ne considèrent pas le modèle caché des distributions de probabilités affectées au bras mais seulement un historique d'actions.

Nous tenons à préciser que cette rapide introduction aux bandits ne prétend dresser un panorama exhaustif de ce sujet¹⁵ dont la littérature extrêmement riche provient de différentes communautés scientifiques (statistiques, probabilités, économie, automatique, apprentissage, *etc.*). Nous y empruntons cependant la problématique naturelle dont une formalisation simple nous permet de simuler le comportement de politiques de sélection d'opérateurs et orienter la réflexion vers de nouvelles pistes de recherche. En effet, un bras symbolise ici un opérateur dont une application au sein d'un algorithme évolutionnaire permet ou non d'apporter un certain bénéfice (récompense). Les opérateurs considérés comme préférables à un certain instant le sont au sens d'un indicateur basé sur l'historique (par exemple le bénéfice moyen constaté), dont la précision dépend naturellement de la fréquence d'utilisation de l'ensemble des opérateurs. Définir une stratégie efficace de sélection adaptative d'opérateurs revient à minimiser, en moyenne, le regret total de la séquence d'opérateurs appliquée.

¹⁵Pour aller plus loin, nous invitons le lecteur intéressé à se référer aux ouvrages [Sutton et Barto, 1998] et [Cesa-Bianchi et Lugosi, 2006].

4.2 Stratégies de bandit

Dans la littérature, un bandit peut aussi bien désigner une problématique précise de bandit — que l'on pourrait abusivement décrire comme une instance de problème — ou une stratégie de bandit à appliquer — qui, selon le même raccourci et appliquée itérativement, correspondrait à un algorithme de résolution. Pour éviter les confusions, une stratégie de bandit sera systématiquement mentionnée explicitement comme telle.

Nous présentons ici quelques stratégies de bandit traditionnellement utilisées, basiques ou plus sophistiquées. L'efficacité d'une stratégie de bandit dépend de son équilibre entre exploitation (sélectionner le meilleur bras selon l'historique) et exploration (sélectionner un autre bras). Aux extrêmes, les stratégies basiques dites *gloutonne* (*greedy*) et *par roulette uniforme* représentent les deux alternatives respectivement purement exploitatrice et purement exploratrice. Chacune de ces deux stratégies possède une variante ajoutant respectivement exploration (stratégie quasi-gloutonne ϵ -*greedy*) et exploitation (*probability matching*).

Stratégie gloutonne (*Greedy*)

La stratégie gloutonne sélectionne systématiquement un bras ayant la meilleure valeur d'action estimée, c'est-à-dire $x_t \in \operatorname{argmax}_{b \in \mathcal{B}} \hat{Q}_t(b)$. On remarque que l'efficacité d'une telle stratégie dépend uniquement de la qualité de l'estimation des valeurs d'action. En particulier, si $\forall t, \forall b, \hat{Q}_t(b) = Q_t^*(b)$, alors la stratégie gloutonne est optimale et son regret sera systématiquement nul.

Sans connaissance *a priori* des $Q_t^*(b)$, une estimation naturelle des valeurs d'action s'obtient en moyennant les récompenses observées sur chacun des bras :

$$\hat{Q}_t(b) = \frac{\sum_{i < t, x_i = b} R_i}{\#\{i < t, x_i = b\}} \quad (12)$$

Un problème de définition apparaît cependant lorsque certains bras n'ont pas encore été actionnés (b tels que $\forall i < t, x_i \neq b$), ce qui est en particulier vrai pour tous les bras lorsque $t = 1$. Dans ce cas, $\hat{Q}_t(b)$ vaut une valeur initiale qui peut être soit *optimiste* (par exemple 1, borne supérieure de toute récompense), soit simplement nulle. Dans le cas optimiste, chaque bras sera actionné au moins une fois avant que la stratégie gloutonne n'opère réellement. Dans le cas où les valeurs d'action estimées sont initialisées à 0, la stratégie gloutonne sélectionnera indéfiniment le premier bras qui apporte une récompense non nulle.

L'estimation des valeurs d'action par moyenne arithmétique des récompenses obtenues est appropriée pour les bandits stationnaires. Dans un contexte non stationnaire, les valeurs d'action sont généralement estimées selon une moyenne pondérée (mise à jour adaptative) qui accorde davantage de crédit aux observations plus récentes et classiquement utilisée en apprentissage par renforcement :

$$\hat{Q}_t(b) = \begin{cases} (1 - \alpha) \times \hat{Q}_{t-1}(b) + \alpha \times R_{t-1} & \text{si } x_{t-1} = b, \\ \hat{Q}_{t-1}(b) & \text{sinon.} \end{cases} \quad (13)$$

$\alpha \in]0, 1]$ est un paramètre d'adaptation qui détermine l'importance de la dernière observation dans l'estimation de la valeur d'action du bras correspondant. Une autre alternative consiste à utiliser une moyenne glissante pour l'établissement des $\hat{Q}_{t-1}(b)$.

Stratégie quasi-gloutonne (ϵ -greedy)

Une variante simple de la stratégie gloutonne incorpore un mécanisme exploratoire consistant à effectuer une certaine proportion d'actions aléatoires. À chaque itération, la stratégie ϵ -greedy [Sutton et Barto, 1998] sélectionne la meilleure action estimée avec une probabilité $1 - \epsilon$, et une action aléatoire avec une probabilité ϵ . En horizon infini, la constante d'exploration permet de faire converger les valeurs d'action estimées vers les valeurs d'action réelles. La vitesse de cette convergence sera plus rapide avec un ϵ élevé, mais en retour le bénéfice des valeurs d'actions plus précises sera amoindri par des appels moins fréquents (proportion de $1 - \epsilon + \frac{\epsilon}{B}$) au bras considéré comme meilleur. En particulier, lorsque le meilleur bras a été détecté, avec une marge d'erreur ne laissant plus de place au doute, seule une stratégie purement gloutonne permet d'assurer des regrets définitivement nuls; bien qu'en théorie, ce doute ne peut être levé avec certitude. Lai et Robbins [1985] prouvèrent que le regret total progresse au moins logarithmiquement en utilisant la meilleure stratégie possible. Dans cette hypothèse, le meilleur bras est joué un nombre exponentiel de fois plus que chacun des autres bras, mais ces derniers ne doivent jamais cesser d'être actionnés. En horizon fini, la vitesse de convergence de la stratégie, variable en fonction des paramétrages, a un impact sur son efficacité. Des stratégies efficaces en horizon infini ou sur un grand nombre d'itérations peuvent ne pas l'être sur un horizon court.

Une sophistication naturelle de ϵ -greedy consiste à ne pas utiliser de paramètre ϵ constant au cours du processus mais de diminuer sa valeur progressivement. Plusieurs schémas de décroissance logarithmique, par paliers ou inspiré du recuit simulé ont été proposés [Sutton et Barto, 1998; Auer *et al.*, 2002; Tokic et Palm, 2011].

Roulette uniforme

La roulette uniforme sélectionne à chaque itération un bras aléatoirement selon une probabilité $\mathbb{P}[x_t = b] = \frac{1}{B}$. Cette stratégie, qui revient à ϵ -greedy avec $\epsilon = 1$, est purement exploratrice et n'implique aucune estimation de valeur d'action. La roulette uniforme est naturellement peu efficace dans le cas général, mais les performances relatives d'autres stratégies par rapport à celle-ci sont intéressantes à mesurer, en particulier lors de l'étude de bandits non stationnaires où les valeurs d'action réelles peuvent changer dans le temps.

Probability matching et Softmax

Le principal inconvénient de la stratégie quasi-gloutonne et qu'elle considère uniformément tous les bras n'ayant pas la plus grande valeur d'action estimée. Au contraire, les algorithmes de type *probability matching* [Goldberg, 1990] discriminent les bras en fonction de leur pertinence et font évoluer dans le temps les probabilités de sélection. La stratégie *probability matching* de base est une roulette proportionnelle adaptative, qui choisit à chaque itération un bras aléatoirement selon une probabilité proportionnelle à

l'estimation de sa valeur d'action. Pour éviter qu'un bras ne puisse plus être activé, il est d'usage d'affecter à chacun une probabilité minimale p_{\min} , même si sa valeur d'action estimée est très faible ou nulle. Cette stratégie reste grandement exploratoire par rapport aux autres, mais une certaine exploitation est garantie par le fait que les meilleurs bras estimés seront actionnés plus souvent. À l'itération t , la probabilité de sélectionner le bras b est la suivante :

$$\mathbb{P}[x_t = b] = p_{\min} + (1 - B \times p_{\min}) \times \frac{\hat{Q}_t(b)}{\sum_{b'=1}^B \hat{Q}_t(b')} \quad (14)$$

En fonction de sa valeur d'action estimée, chaque bras voit sa probabilité de sélection comprise entre $p_{\min} \leq \frac{1}{B}$ et $p_{\max} = 1 - (B - 1) \times p_{\min}$. Les probabilités de sélection associées aux bras doivent être mises à jour à chaque itération.

Une autre manière d'établir des probabilités de sélection en fonction des $\hat{Q}_t(b)$ est d'utiliser une distribution de Gibbs qui, au moyen d'une température τ qui décroît progressivement quand t augmente, permet d'initialiser le processus avec un fonctionnement proche de celui de la roulette uniforme, puis de le faire converger vers une stratégie gloutonne. Entre temps, les probabilités de sélection restent dégressives en fonction de l'estimation des valeurs d'action. Plus précisément, cette stratégie de bandit dite *softmax*, qui trouve ses origines dans les travaux de Luce [1959], emploie les probabilités de sélection suivantes :

$$\mathbb{P}[x_t = b] = \frac{e^{\frac{\hat{Q}_t(b)}{\tau}}}{\sum_{b'=1}^B e^{\frac{\hat{Q}_t(b')}{\tau}}} \quad (15)$$

Enfin, les stratégies de type *randomized probability matching* sélectionnent chaque bras selon la probabilité qu'il s'agisse du meilleur, reprenant l'idée initiale de Thompson [1933]. Peu de travaux cependant utilisent l'échantillonnage de Thompson affectant à chaque bras une probabilité d'activation égale à la vraisemblance qu'il s'agisse de la meilleure action selon l'historique. En particulier, Scott [2010] utilise l'inférence bayésienne pour calculer efficacement ces probabilités.

Poursuite

L'algorithme de poursuite [Thathachar et Sastry, 1985] est une autre proposition cherchant à renforcer au maximum le meilleur bras estimé, tout en discriminant les autres actions par des probabilités de sélection propres. Comme les stratégies de type *probability matching*, la poursuite choisit chaque action suivant une probabilité de sélection associée à chaque bras. Si dans les politiques précédentes, les valeurs des probabilités étaient directement calculées en fonction des valeurs d'action estimées, la poursuite distingue au contraire la mise à jour des probabilités de sélection $P_t(b)$ de celle des valeurs d'action. À chaque itération, le meilleur bras estimé voit sa probabilité de sélection renforcée selon un paramètre d'apprentissage β , les autres probabilités étant diminuées par normalisation :

$$P_t(b) = \begin{cases} (1 - \beta) \times P_{t-1}(b) + \beta & \text{si } b = \operatorname{argmax}_{b'} \hat{Q}_{t-1}(b')^{16}, \\ (1 - \beta) \times P_{t-1}(b) & \text{sinon.} \end{cases} \quad (16)$$

Une version adaptative (*adaptive pursuit*) a été proposée par Thierens [2005] pour traiter les bandits non stationnaires. En plus du paramètre β , cette stratégie requiert une probabilité de sélection minimale p_{\min} pour chaque bras, afin de conserver une exploration constante nécessaire pour ce type de bandits. On en déduit $p_{\max} = 1 - (B - 1) \times p_{\min}$ (voir ci-avant). Associée à une mise à jour adaptative des valeurs d'action estimées, les probabilités de sélection évoluent de la manière suivante :

$$P_t(b) = \begin{cases} (1 - \beta) \times P_{t-1}(b) + \beta \times p_{\max} & \text{si } b = \operatorname{argmax}_{b'} \hat{Q}_{t-1}(b'), \\ (1 - \beta) \times P_{t-1}(b) + \beta \times p_{\min} & \text{sinon.} \end{cases} \quad (17)$$

UCB

Si la stratégie purement glotonne se trouve annihilée par une mauvaise estimation des valeurs d'action, elle reste très efficace lorsque celles-ci sont estimées finement. Le compromis de laisser une fenêtre d'exploration peut avoir l'inconvénient de ne pas discriminer les bras en fonction des valeurs d'action (ϵ -greedy) ou de ne pas tenir compte de la confiance que l'on a en ces valeurs (*probability matching*). Bien que les stratégies de type poursuite ou *softmax* renforcent progressivement le ou les meilleur(s) bras, leur caractère stochastique nécessaire à une efficacité théorique à horizon infini maximisé peut pénaliser leur performance en pratique. En effet, un joueur ayant une estimation qu'il juge suffisante — à défaut d'être certaine — de la valeur de chaque bras aurait certainement intérêt à actionner systématiquement, pour maximiser son gain à court, moyen ou même long terme, le bras jugé le meilleur. C'est le principe des stratégies dites à bornes de confiance, dont le mécanisme opératoire suit la stratégie glotonne mais en considérant des utilités ne rendant pas uniquement compte des valeurs d'action estimées, mais également de la confiance que l'on a en chacune de ces valeurs.

Cette confiance peut s'exprimer par une marge d'erreur qui diminue, pour chaque bras, en fonction du nombre d'observations. La marge d'erreur $E_t(b)$ de $\hat{Q}_t(b)$ se définit de manière à ce que $\mathbb{P}[\hat{Q}_t(b) - E_t(b) \leq Q_t^*(b) \leq \hat{Q}_t(b) + E_t(b)]$ soit supérieure à un seuil de confiance attendu. $[\hat{Q}_t(b) - E_t(b), \hat{Q}_t(b) + E_t(b)]$ est l'intervalle de confiance de $Q_t^*(b)$ et plus précisément, $\hat{Q}_t(b) + E_t(b)$ représente sa borne de confiance supérieure (*upper confidence bound*, UCB).

En se basant sur les travaux de Lai et Robbins [1985] puis Agrawal [1995], Auer *et al.* [2002] formalisent les algorithmes de bandit à borne de confiance supérieure dont la première variante, UCB1, consiste à sélectionner à chaque itération le bras b maximisant l'*index* $\hat{Q}_t(b) + \sqrt{\frac{2 \ln n}{N_t(b)}}$, qui représente une *valeur d'action optimiste*. $\hat{Q}_t(b)$ reste la moyenne des récompenses obtenues par le bras b et $N_t(b)$ est le nombre d'activations du bras. UCB requiert que chacun des bras soit actionné une fois avant que la stratégie ne puisse s'appliquer. L'unique différence avec la stratégie glotonne purement exploitative provient des marges d'erreur représentant ici la part exploratrice de la stratégie qui reste pourtant déterministe. Ces termes diminuent lorsque les bras correspondants sont

¹⁶On suppose que b est arbitrairement choisi parmi $\operatorname{argmax}_{b'} \hat{Q}_{t-1}(b')$ si cet ensemble contient plus d'un élément, ce qui est induit par cette notation abusive néanmoins largement répandue.

sélectionnés. UCB1 choisit ainsi à chaque itération le bras le meilleur selon un jugement optimiste, ce qui conduit à affiner les valeurs d'action estimées les plus pertinentes au moyen d'une exploration ayant moins d'impact négatif sur le regret total. Plus les valeurs d'action estimées sont incertaines, plus l'exploration est nécessaire.

Les particularités des algorithmes à borne de confiance supérieure viennent du calcul de leur index de confiance. L'index de UCB1 utilise la marge d'erreur $\sqrt{\frac{2 \ln n}{N_t(b)}}$ qui s'obtient à partir des bornes de Chernoff-Hoeffding [Chernoff, 1952; Hoeffding, 1963]. Auer *et al.* [2002] proposent d'autres variantes (UCB1-TUNED, UCB2) plus fines mais dont les calculs d'index sont plus complexes. Da Costa *et al.* [2008] discutent de l'intérêt d'ajouter un facteur d'échelle constant à la marge d'erreur d'UCB1 afin de paramétrer l'effort exploratoire en fonction, par exemple, du nombre de bras. Liu *et al.* [2014] proposent de faire évoluer dynamiquement ce facteur d'échelle de manière à ce que la taille de l'intervalle de confiance soit toujours minimale tout en respectant un taux d'erreur fixe (stratégie UCB-EXPLORE).

DMAB

Les stratégies de type UCB ne sont pertinentes que dans un contexte de bandits stationnaires. En effet, les marges d'erreur décroissent au cours du temps puisque l'on suppose que les vraies valeurs d'action sont constantes. Da Costa *et al.* [2008] étendent l'algorithme UCB1 en lui permettant de se réinitialiser au cours de la recherche dès qu'un changement d'hypothèse est détecté au moyen d'un test de Page-Hinkley ([Hinkley, 1970], d'après [Page, 1954]). Il s'agit de comparer après chaque itération la somme courante $|m_t| = |\sum_{i=s+1}^t (R_i - \bar{R}_i + \delta)|$ des différences entre récompenses effectives et récompenses moyennes $\bar{R}_i = \sum_{j=s+1}^i \frac{r_j}{i}$ (incluant une tolérance δ) depuis la dernière réinitialisation à l'itération s (avec $s = 0$ si aucune réinitialisation n'a été effectuée), avec la valeur maximale de cette somme atteinte au cours de cette période ($\max_{i=s+1}^{t-1} |m_i|$). Lorsque l'écart entre ces deux valeurs dépasse un certain seuil γ , alors un changement dans les lois de probabilités régissant les récompenses est détecté et le processus UCB est réinitialisé. Avec cette stratégie DMAB (*Dynamic Multi-Armed Bandit*), les index à l'itération t ne sont plus calculés à partir des observations $((x_1, R_1), \dots, (x_{t-1}, R_{t-1}))$, mais des observations $((x_{s+1}, R_{s+1}), \dots, (x_{t-1}, R_{t-1}))$.

Comme le notent Fialho *et al.* [2010], DMAB permet de détecter d'abrupts changements dans l'efficacité des bras, tandis que les changements d'efficacité graduels des bras sont plus difficiles à déceler. En outre, la totale réinitialisation des paramètres ne permet pas d'utiliser la connaissance passée au cas où l'efficacité relative de l'ensemble des bras n'a pas été totalement redistribuée. Dans ce cas de figure, les auteurs proposent d'estimer les valeurs d'action à partir de la moyenne arithmétique des récompenses restreinte à la fenêtre des W dernières itérations (*Sliding Multi-armed Bandit*). Cette stratégie, qui a été également proposée par Garivier et Moulines [2011], dispense l'usage du test de Page-Hinkley.

4.3 Bandits à bras interconnectés

Bandits classiques et sélection adaptative des opérateurs

Dans sa formulation classique, un bandit est décrit par un ensemble fini d'observations — éventuellement vide — des récompenses obtenues par différentes actions. Chaque action (bras) est associée à une loi de probabilité inconnue, que l'analyse des observations consiste à inférer. Les lois de probabilités sont inchangées au cours du temps (*bandit stationnaire*) et sont indépendantes (*bras indépendants*). Néanmoins, avec leurs très nombreuses applications, en particulier dans les systèmes de recommandation (requêtes web, sites marchands, publicité, *etc.*) ou les problèmes de tarification automatique, des modèles de bandits plus généraux sont proposés. Les *bandits non stationnaires*, où les lois de probabilités régissant les bras peuvent varier au cours du temps, sont de plus en plus étudiés [Thierens, 2005; Cohen *et al.*, 2007; Da Costa *et al.*, 2008; Koulouriotis et Xanthopoulos, 2008; Agarwal *et al.*, 2009; Yu et Mannor, 2009; Granmo et Berg, 2010]. D'autres modèles de bandit, contextuels [Wang *et al.*, 2005; Langford et Zhang, 2008; Li *et al.*, 2010] ou avec bras dépendants [Pandey *et al.*, 2007], ont été proposés en levant l'hypothèse d'indépendance des bras. Les modèles les plus généraux (*restless bandits* [Whittle, 1988]) font évoluer les récompenses des bras au moyen d'un processus de Markov.

Déterminer le prochain opérateur à appliquer à un individu dans un algorithme évolutionnaire à partir des observations précédentes peut être abstrait en un problème de bandit, où un bras représenterait un opérateur. Des mesures d'utilité — ou de valeurs d'action estimées — de chaque opérateur peuvent être calculées en fonction de ses précédentes applications. Sélectionner le meilleur opérateur à partir de ces informations consiste à appliquer une stratégie de bandit. Le caractère adaptatif est inhérent aux bandits, puisque chaque nouvelle action apporte une nouvelle observation qui peut entraîner un changement dans le choix, ou la probabilité de sélection, du prochain opérateur.

La problématique de la sélection d'opérateurs a été discutée dans le chapitre 3 et illustrée à partir du problème One-Max. Dans ce cadre, nous avons vu que l'efficacité relative des opérateurs de recherche locale utilisés évoluait au cours du temps. Dans des problèmes plus classiques où les paysages de recherche comportent de nombreux optima locaux, combiner plusieurs opérateurs de recherche locale basés sur des relations de voisinage, des stratégies de sélection, voire des fonctions d'évaluation différentes, permet à des algorithmes évolutionnaires d'atteindre de bonnes solutions. Cela est néanmoins conditionné par l'application des opérateurs appropriés au bon moment. En particulier, un opérateur de diversification appliqué trop fréquemment ou trop rarement pénalisera ou bloquera la recherche. Des opérateurs d'intensification orthogonaux pourront être moins efficaces utilisés successivement que périodiquement. De même, un opérateur a tendance à perdre de son efficacité à force d'être utilisé, mais peut en regagner après l'application intermédiaire d'autres opérateurs, y compris d'opérateurs dont l'apport immédiat semble très faible voire nul.

Dans un algorithme faisant évoluer une population d'individus, le problème de sélection d'opérateurs peut être modélisé par un problème de bandit particulier qui considérerait simultanément plusieurs séquences d'actions. Chaque individu correspond à une séquence d'actions, et l'action suivante peut être déterminée à partir de l'observation des séquences

analogues de l'ensemble des individus. Des stratégies de bandit alternatives pourraient être envisagées dans le cas où seul importe le gain maximal d'un seul individu. On pourrait alors imaginer des individus spécifiques dont le rôle est d'explorer les différents bras, par exemple en suivant la stratégie de la roulette uniforme, et des individus coucourant pour le meilleur gain à qui on appliquerait une stratégie gloutonne. Cette combinaison de stratégies parallèles, envisageable dans le cadre de bandits à bras indépendants, même non stationnaires, montre la limite de cette abstraction pour modéliser l'utilité des opérateurs évolutionnaires sur des individus. En effet, l'utilité d'un opérateur évolutionnaire pertinent sur un individu dépend de l'état de l'individu, qui est lui-même défini à partir des opérateurs qui lui ont été appliqués précédemment. Une stratégie similairement employée pour la sélection d'opérateurs dans un algorithme évolutionnaire serait inefficace, car les valeurs d'action des opérateurs appliqués aux individus explorateurs ne seraient pas représentatives de celles des mêmes opérateurs appliqués à des individus de meilleure évaluation, c'est-à-dire ayant cumulé un gain plus important.

Modèle de bandits à bras interconnectés

S'il est évident que la sélection adaptative d'opérateurs peut s'abstraire en un problème de bandit non stationnaire, il l'est beaucoup moins de déterminer des schémas de distribution de récompenses évoluant dans le temps et simulant, même grossièrement, les interactions entre opérateurs ou l'efficacité évolutive de mêmes actions itérées. Nous avons récemment proposé un modèle de bandit non stationnaire original dit *à bras interconnectés* basé sur un bandit classique — donc éventuellement stationnaire — mais dont les récompenses dépendent en outre des bras activés récemment [Candan *et al.*, 2013; Goëffon *et al.*, 2014]. Dans ce modèle, l'évolution de la valeur d'action réelle d'un bras est proportionnelle à son utilisation au cours d'une fenêtre de temps.

En extension du modèle de bandit considéré précédemment, où les lois de probabilité régissant les bras sont données par B couples (p_b, r_b) , nous ajoutons un unique paramètre supplémentaire W inconnu du joueur (de la stratégie) représentant la taille de la fenêtre à partir de laquelle les précédentes activations impactent les récompenses. Étant donné un historique d'actions (x_1, \dots, x_{t-1}) , la récompense de l'action b à l'itération t est :

$$R_t(b) = \begin{cases} r_b \times \left(1 - \frac{\#\{x_{t'}=b\}_{t' \in \{\max(1, t-W), \dots, t-1\}}}{W}\right) & \text{avec une probabilité } p_b, \\ 0 & \text{avec une probabilité } 1 - p_b. \end{cases} \quad (18)$$

Le terme $\#\{x_{t'} = b\}_{t' \in \{\max(1, t-W), \dots, t-1\}}$ représente le nombre d'occurrences de b dans la séquence composée des W derniers mouvements — ou de la séquence totale si $W \geq t-1$. Des bras qui n'ont pas été activés au cours des W dernières itérations ont une vraie valeur d'action de $p_b \times r_b$. Celle-ci décroît linéairement en fonction du nombre d'actions b parmi les W dernières actions. Ainsi, considérant $t > W$, nous en déduisons la propriété suivante :

$$Q_{t+1}^*(b) = \begin{cases} Q_t^*(b) - \frac{p_b \times r_b}{W} & \text{si } x_{t-W} \neq b \text{ et } x_t = b, \\ Q_t^*(b) + \frac{p_b \times r_b}{W} & \text{si } x_{t-W} = b \text{ et } x_t \neq b, \\ Q_t^*(b) & \text{sinon.} \end{cases} \quad (19)$$

Poursuivons notre parallèle entre stratégie de bandit et politique de sélection d’opérateurs dans un algorithme évolutionnaire. L’espérance de gain de l’application d’un opérateur sur un individu est grandement dépendante de l’individu lui-même. Si l’on considère que, schématiquement, un individu qui évolue au moyen d’opérateurs est en partie défini par la séquence d’opérateurs qui lui a été appliquée, alors cette espérance de gain est également dépendante de cette séquence d’opérateurs. Le modèle de bandits à bras interconnectés est basé sur cette hypothèse, puisque les récompenses sont indexées au moyen de la fenêtre qui mémorise ses W derniers opérateurs appliqués — éventuellement toute la séquence si W est choisi très grand. La part indépendante du rendement d’un bras est donnée par sa valeur d’action maximale $p_b \times r_b$, permettant à des opérateurs d’être plus rémunérateurs en moyenne que d’autres. L’interconnexion des bras apparaît dans le sens où une action détermine les prochaines valeurs d’action de l’ensemble des bras (voir équation 19). Enfin, supposons que plusieurs stratégies soient appliquées simultanément sur un tel bandit, afin de simuler le cas d’un algorithme évolutionnaire à population d’individus ; dans ce cas, les vraies valeurs d’action de chaque bras seront propres à chaque stratégie (individu), ce qui est impossible à modéliser avec une formulation classique.

Bandits à bras interconnectés simplifiés

Afin d’exhiber au maximum la problématique du bandit à bras interconnectés, nous définissons un cas particulier simplifié, où $\forall b \in \mathcal{B}, p_b = 1$ et $r_b \in \{0, 1\}$. Les récompenses R_b sont toujours indexées par une fenêtre d’actions (voir équation 18) et prennent leur valeur dans l’ensemble $\{\frac{k}{W}\}_{k \in \{0, \dots, W\}}$. Dans ce modèle simplifié, les récompenses R_b des bras tels que $r_b = 1$ (appelés *bras utiles*) sont uniquement déterminées par l’historique des actions. Les bras tels que $r_b = 0$ sont dits *neutres* et engendrent systématiquement une récompense nulle ; la vraie valeur d’action de ces bras est donc à tout moment égale à 0. Un bandit à bras interconnectés simplifié est décrit par son nombre de bras utiles B_1 , son nombre de bras neutres B_0 et sa taille de fenêtre W .

Nous appelons *oracle* la stratégie optimale ayant connaissance de l’ensemble des informations d’un bandit — incluant le mode de calcul des récompenses — et *oracle myope* la stratégie gloutonne basée sur les vraies valeurs d’action. L’oracle myope connaît ainsi avant chaque action la récompense espérée associée à chaque bras, tandis que l’oracle peut prédire l’ensemble des valeurs d’action futures, jusqu’à l’horizon T , à partir de toutes les combinaisons d’actions envisageables. Sur les modèles de bandit classiques, stationnaires ou non stationnaires, les deux stratégies sont équivalentes. Comme nous l’avons vu, une stratégie gloutonne basée sur les vraies valeurs d’action est une stratégie de bandit optimale. Cependant, ce n’est plus nécessairement le cas sur un bandit à bras interconnectés, tout comme sur un bandit dont les récompenses suivent un processus de Markov [Ortner *et al.*, 2012].

Prenons l’exemple du bandit à deux bras interconnectés simplifié où $B_0 = 1, B_1 = 1, W = 3$, et avec un horizon $T = 12$ (voir table 2). Par commodité ici, les bras sont indicés à partir de 0, soit $\mathcal{B} = \{0, 1\}$, de manière à ce que 0 symbolise le bras neutre ($r_0 = 0$) et 1 le bras utile ($r_1 = 1$). À tout moment, $Q_t^*(0) = 0$, tandis que $Q_t^*(1)$ prend ses valeurs dans l’ensemble $\{0, \frac{1}{3}, \frac{2}{3}, 1\}$ en fonction des trois actions précédentes (0 si le bras utile vient d’être activé trois fois, 1 si le bras neutre vient d’être activé trois fois). L’oracle myope sélectionne

t	1	2	3	4	5	6	7	8	9	10	11	12
Meilleur oracle myope (gain : 4)												
$Q_t^*(0)$	0	0	0	0	0	0	0	0	0	0	0	0
$Q_t^*(1)$	1	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0
x_t	1	1	1	0	1	1	1	0	1	1	1	(0)
R_t	1	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0
Oracle (gain : 5)												
$Q_t^*(0)$	0	0	0	0	0	0	0	0	0	0	0	0
$Q_t^*(1)$	1	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$								
x_t	1	1	0	1	0	1	0	1	0	1	0	1
R_t	1	$\frac{2}{3}$	0	$\frac{2}{3}$								

Paramètres du bandit
$B = 2$
$(p_0, r_0) = (1, 0)$
$(p_1, r_1) = (1, 1)$
$W = 3$
Horizon
$T = 12$

TABLE 2 – Comparaison entre oracle myope et oracle sur un bandit à deux bras interconnectés. Les caractéristiques du bandit sont indiquées à droite.

à chaque fois le bras $\operatorname{argmax}_{b \in \mathcal{B}} Q_t^*(b)$ — ou l'un de ces bras si $Q_t^*(0) = Q_t^*(1)$. Tant que le bras utile n'a pas été activé trois fois, celui-ci est sélectionné, impliquant $Q_4^*(1) = 0$. L'intérêt du bras neutre apparaît alors, malgré sa vraie valeur d'action à tout instant nulle, puisque son activation permet ensuite de débloquent le bras utile. L'oracle myope n'a pas cette connaissance et peut actionner arbitrairement l'un de deux bras. Supposons néanmoins le cas le plus favorable, avec $x_4 = 0$. Le choix du bras neutre implique des récompenses non nulles sur les trois actions suivantes en sélectionnant le bras utile. La meilleure stratégie envisageable pour un oracle myope est ainsi d'effectuer, itérativement, trois actions du bras utile suivies d'une action du bras neutre ; ce qui apporte, à un horizon $T = 12$, un gain de 4. La table 2 décrit la séquence d'actions effectuée par le *meilleur* oracle myope. Tout mauvais choix — pourtant arbitraire du point de vue de la stratégie — à savoir sélectionner le bras 1 lorsque $Q_t^*(1) = Q_t^*(0) = 0$ — entraîne ici une perte de gain de $\frac{1}{3}$, sauf lorsque $t = T$.

La table 2 compare en outre le gain obtenu par cet oracle myope avec le gain maximal atteignable sur cette configuration de bandit (à un horizon $T = 12$). Une stratégie optimale (oracle) consiste ici à actionner chaque bras alternativement, ce qui permet une récompense de $\frac{2}{3}$ toutes les deux itérations contre une récompense cumulée de 1 toutes les quatre actions, dans le meilleur des cas, avec un oracle myope. Notons qu'une autre stratégie optimale consiste à appliquer successivement deux actions du bras utile, deux actions du bras neutre, puis d'itérer le processus ; soit une récompense cumulée de $\frac{4}{3}$ sur une période de quatre actions. Il apparaît d'ailleurs que quel que soit le problème de bandit à bras interconnectés considéré, la stratégie optimale consiste à répéter une même séquence d'actions. Seules les *amorces* (début et fin de la séquence complète) peuvent différer de cette succession d'actions périodique optimale, en fonction de T .

Dans notre exemple, la stratégie optimale a été déterminée en envisageant toutes les séquences d'actions possibles. La problématique de déterminer une stratégie optimale $S_{\text{opt}} \in \mathcal{B}^N$ peut se modéliser à partir d'un graphe — que l'on pourrait assimiler à un automate — associé au bandit. Sur la figure 13, on considère quatre variations de notre bandit-exemple à deux bras interconnectés simplifié avec un bras neutre (0), un bras utile (1), et différentes valeurs de W (de 1 à 4). Les vraies valeurs d'action $Q_t^*(b)$ dépendent

des W dernières actions effectuées, que l'on associe à un état $q \in \mathcal{B}^W$. Chaque action b implique alors une transition $\delta(q, b) = q'$, où q' est tel que $\forall i < W, q'_i = q_{i+1}$, et $q'_W = b$. Chaque transition $\delta(q, b)$ apporte la récompense $\frac{1-|q|_b}{W}$ si $r_b = 1$, et 0 si $r_b = 0$; $|q|_b$ est le nombre d'occurrences de b dans le mot q . Dans notre exemple, les transitions $\delta(b, 1)$ apportent une récompense égale à $\frac{|q|_0}{W}$, et les transitions $\delta(q, 0)$ une récompense nulle. On note $\Delta = \{(q, \delta(q, b))\}_{(q,b) \in \mathcal{B}^W \times \mathcal{B}}$ l'ensemble des transitions possibles d'état à état. Une stratégie à horizon T est ainsi un chemin de longueur T dans le graphe (\mathcal{B}^W, Δ) et débutant à l'état q_0 tel que $|q_0|_0 = W$, c'est-à-dire l'état dont le mot est composé uniquement de zéros (0^W). Le gain associé à une stratégie est la somme des récompenses collectées. La stratégie optimale est un chemin de longueur T qui maximise le gain.

Il semble évident qu'à horizon infini, la stratégie optimale est composée d'une amorce éventuellement vide puis d'un cycle élémentaire dans le graphe. Sous cette hypothèse — qui nous reste à démontrer —, amorce mise à part, déterminer la stratégie optimale consiste à identifier un cycle élémentaire qui maximise la moyenne des récompenses. À horizon fini, la stratégie optimale est la même à l'exception d'une éventuelle amorce de fin.

Sur les quatre exemples de la figure 13, les cycles optimaux (et de longueur minimale) ont été calculés au moyen d'un algorithme exact (non polynomial) :

- (a) Cycle d'états optimal : (0,1) ;
Cycle de bras optimal : (1,0) ;
Récompense moyenne : 0.5.
- (b) Cycle d'états optimal : (00,01,11,10) ;
Cycle de bras optimal : (1,1,0,0) ;
Récompense moyenne : 0.375.
- (c) Cycle d'états optimal : (010,101) ;
Cycle de bras optimal : (1,0) ;
Récompense moyenne : 0.333 (voir aussi table 2).
- (d) Cycle d'états optimal : (011,111,110,100,000,001) ;
Cycle de bras optimal : (1,1,1,0,0,0) ;
Récompense moyenne : 0.333.

On y observe dans tous les cas étudiés une séquence d'actions utiles suivie d'une séquence d'actions neutres. Nous pouvons avancer un parallèle avec l'utilisation pertinente des différents opérateurs dans les algorithmes évolutionnaires. Ici le bras utile correspond à un opérateur d'intensification, dont l'utilité baisse après un certain nombre d'actions répétées. Le bras neutre représente l'opérateur de diversification qui n'apporte aucun gain immédiat en termes de qualité, mais permet à l'opérateur d'intensification, après quelques applications, de retrouver un certain potentiel.

La table 3 référence les longueurs de cycles optimaux, ainsi que les récompenses moyennes par action associées, jusqu'à $W = 8$.

Utiliser les bandits comme modèle d'abstraction pour la sélection adaptative des opérateurs est une voie intéressante, à condition de disposer de modèles de bandit dont la dynamique des récompenses permet de simuler des schémas cohérents d'évolution des utilités d'opérateurs évolutionnaires. Avec les bandits à bras interconnectés, nous proposons

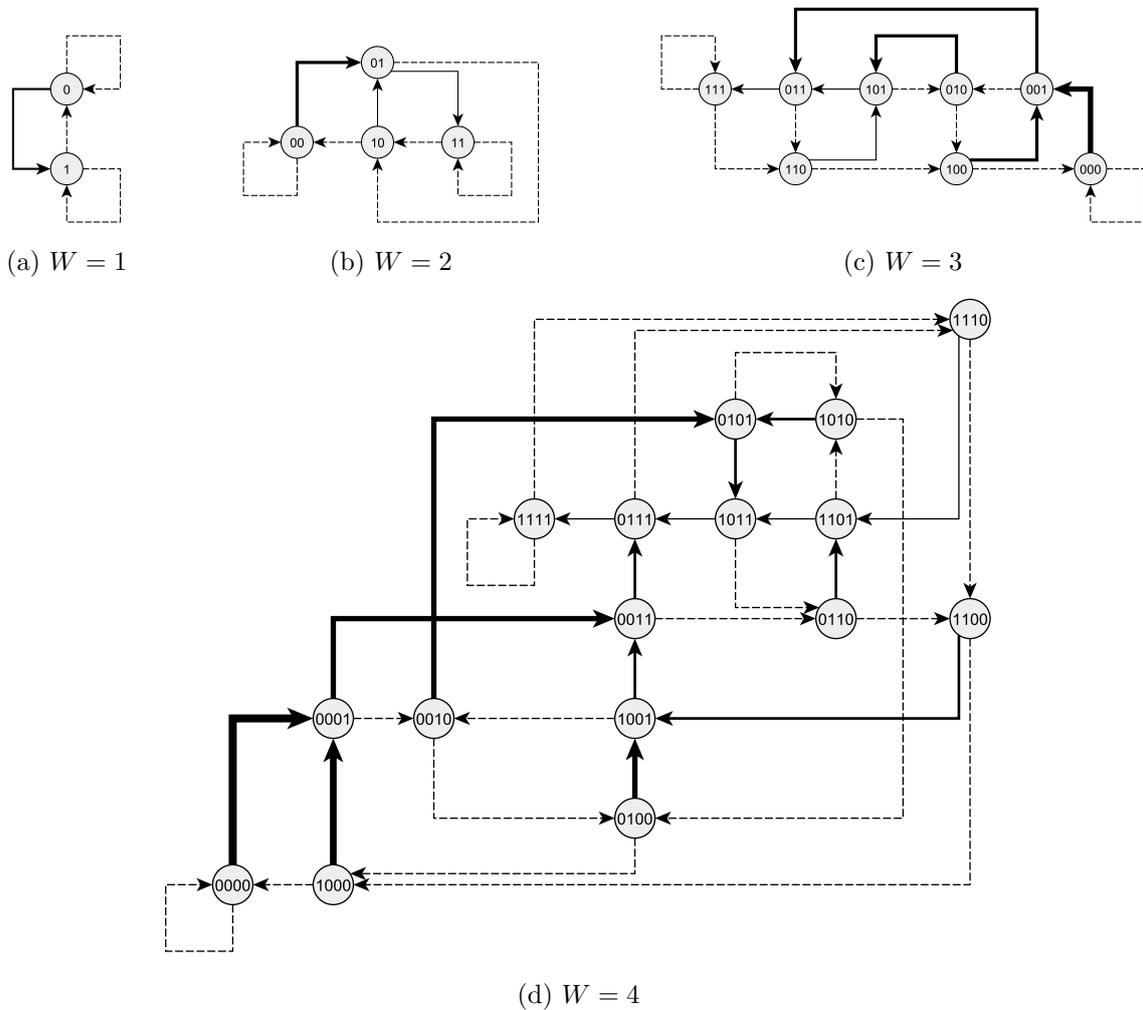


FIGURE 13 – Représentation graphique de la distribution des récompenses sur quatre bandits à bras interconnectés simplifiés composés d'un bras neutre (0) et un bras utile (1), avec $W \in \{1, 2, 3, 4\}$. Les états représentent les W derniers bras visités. L'épaisseur des transitions (selon une échelle différente en fonction des graphes) symbolise les valeurs des différentes actions possibles. Les transitions représentées en pointillé apportent une récompense nulle.

W	1	2	3	4	5	6	7	8
Longueur du cycle optimal	2	4	2	6	8	10	10	13
Récompense moyenne	0.500	0.375	0.333	0.333	0.325	0.317	0.314	0.308

TABLE 3 – Longueurs et récompenses moyennes des cycles optimaux de bandits à deux bras interconnectés simplifiés avec $B_0 = 1$ et $B_1 = 1$, pour différentes valeurs de W .

un modèle de bandit original qui rend compte de certaines spécificités de ces schémas. Déterminer des stratégies efficaces pour ce type de bandits non stationnaires est alors une problématique ouverte. Nous avons vu que même construire la stratégie optimale (oracle) en pleine connaissance des caractéristiques du bandit semble déjà être un problème difficile. Dans la section suivante, nous appliquons quelques stratégies de bandit classiques sur les bandits interconnectés afin d’en améliorer la compréhension. Le propos n’est naturellement pas de comparer de manière absolue leur efficacité, puisque ces stratégies sont conçues spécifiquement pour aborder des bandits classiques, dont la distribution des récompenses suit certaines hypothèses que ne respectent pas les bandits à bras interconnectés. Nous discuterons aussi de la possibilité d’utiliser le modèle en iles dynamique comme stratégie de bandit.

4.4 Application de stratégies classiques aux bandits à bras interconnectés

Nous avons appliqué les différentes stratégies de bandit présentées section 4.2 à des bandits à bras interconnectés. 64 bandits simplifiés à 8 bras sont utilisés pour l’expérimentation, obtenus à partir de toutes les combinaisons de $B_1 \in \{1, \dots, 8\}$ et $W \in \{1, \dots, 8\}$. B_1 est le nombre de bras utiles du bandit, composé également de $B_0 = 8 - B_1$ bras neutres. Théoriquement, les bras neutres sont tous équivalents et actionner l’un plutôt qu’un autre à tout moment a exactement le même effet. Un oracle pourrait n’utiliser qu’un seul bras neutre durant l’intégralité du processus. Cependant, certaines stratégies de bandit — qui n’ont pas de connaissance *a priori* sur les bras — peuvent être davantage affectées que d’autres par la présence de plusieurs bras neutres. Précisons que lorsque plusieurs bras utiles sont considérés, l’activation d’un bras utile a le même effet sur les autres bras que l’activation d’un bras neutre. La taille de la fenêtre est également un élément déterminant. Lorsque le nombre de bras utiles est strictement supérieur à la taille de la fenêtre (W), une stratégie optimale consiste à appliquer alternativement au moins W bras utiles. Dans ce cas de figure, chaque action apporte une récompense de 1. Par exemple, considérant le bandit tel que $B_1 = 5$ et $W = 4$, l’unique possibilité d’atteindre un gain de T à l’horizon T est de définir un ordre sur les 5 bras afin de les actionner un à un, toujours selon le même ordre, en répétant le processus jusqu’à l’horizon T . Cette stratégie est détectée par un oracle myope qui parvient ainsi à déterminer une séquence optimale lorsque le nombre de bras utiles est strictement supérieur à W . Les stratégies optimales sont en revanche plus difficiles à déterminer lorsque $B_1 \leq W$, y compris en ayant pleine connaissance du modèle.

Les stratégies retenues pour nos expérimentations sont les suivantes :

- ϵ -greedy (ϵ -G), avec $\epsilon = 0.05$;
- Roulette uniforme (RU) ;
- *Probability matching* (PM), avec $p_{\min} = 0.05$;
- Poursuite adaptative (PA), avec $p_{\min} = 0.1$ et $\beta = 0.7$;
- UCB, avec l'algorithme UCB1.

Précisons que DMAB a également été testée, mais la particularité de ces bandits à bras interconnectés fait que le test de Page-Hinkley ne détecte pas de changement d'hypothèse. DMAB est donc ici équivalente à UCB1.

Les cinq stratégies, ainsi que l'oracle myope (OM) qui connaît les récompenses de chacun des bras avec un coup d'avance, ont été appliquées 1000 fois sur chacun des 64 bandits, à un horizon $T = 1000$. Nous reportons dans la table 4 les moyennes des récompenses obtenues par chaque stratégie. On y remarque que UCB peut être relativement efficace sur ce type de bandits, sauf cas particulier ($W = 1, B_1 = 1$). Considérant un seul bras utile, l'oracle myope devient inefficace lorsque la taille de la fenêtre augmente. Comme nous l'avons vu, celui-ci ne distingue pas bras utiles et bras neutres. Lorsqu'il n'y a qu'un bras utile, celui-ci ne peut se recharger qu'en activant un bras neutre. Plus qu'une comparaison de stratégies de bandit, cette expérimentation apporte une vision plus concrète des bandits à bras interconnectés.

W = 1								
B ₁	1	2	3	4	5	6	7	8
ϵ -G	42	71	99	121	150	174	196	212
RU	109	218	328	438	547	657	766	874
PM	227	448	557	635	701	760	817	870
PA	210	303	396	486	577	668	758	848
UCB	86	394	472	688	912	970	980	987
OM	467	1000	1000	1000	1000	1000	1000	1000

W = 2								
B ₁	1	2	3	4	5	6	7	8
ϵ -G	43	80	121	160	196	228	259	282
RU	109	219	329	438	547	657	765	875
PM	227	449	558	635	701	761	817	871
PA	210	306	399	491	582	674	765	855
UCB	116	547	697	734	953	977	985	993
OM	319	667	1000	1000	1000	1000	1000	1000

W = 3								
B ₁	1	2	3	4	5	6	7	8
ϵ -G	44	87	139	192	236	275	311	342
RU	109	219	329	437	547	656	766	875
PM	227	450	558	636	701	762	818	872
PA	210	308	401	494	586	677	769	859
UCB	141	585	694	798	851	957	985	994
OM	242	667	834	1000	1000	1000	1000	1000

W = 4								
B ₁	1	2	3	4	5	6	7	8
ϵ -G	44	102	161	219	256	320	356	391
RU	109	219	329	437	547	657	766	875
PM	227	450	558	636	702	762	818	872
PA	210	310	404	496	588	679	771	861
UCB	164	549	731	832	851	892	978	994
OM	196	601	801	900	1000	1000	1000	1000

W = 5								
B ₁	1	2	3	4	5	6	7	8
ϵ -G	44	110	178	249	294	351	394	428
RU	109	218	328	437	547	657	766	875
PM	228	450	558	636	702	762	818	872
PA	210	311	405	498	590	682	773	864
UCB	184	562	724	820	871	885	910	980
OM	165	601	801	867	934	1000	1000	1000

W = 6								
B ₁	1	2	3	4	5	6	7	8
ϵ -G	45	115	192	261	319	376	431	463
RU	109	219	329	438	547	657	766	876
PM	227	451	559	636	702	762	818	872
PA	210	313	408	500	592	684	775	866
UCB	200	552	737	801	871	897	908	924
OM	143	573	763	858	905	953	1000	1000

W = 7								
B ₁	1	2	3	4	5	6	7	8
ϵ -G	45	124	208	284	351	407	462	501
RU	109	219	328	438	547	656	766	875
PM	228	451	559	637	702	763	819	873
PA	210	313	408	501	593	685	776	867
UCB	217	543	699	794	843	897	917	928
OM	126	573	751	858	893	929	964	1000

W = 8								
B ₁	1	2	3	4	5	6	7	8
ϵ -G	46	135	222	304	361	438	482	525
RU	109	219	329	438	547	657	767	875
PM	228	451	559	637	703	763	819	873
PA	210	314	409	503	594	686	777	867
UCB	229	534	703	773	843	863	911	934
OM	113	557	751	834	889	917	945	972

TABLE 4 – Gain moyen des stratégies de bandit appliquées à 64 bandits à bras interconnectés simplifiés différents ($B = 8, B_1$ et W variables).

Un point notable est que sur ce type de bandits, les gains obtenus par une roulette uniforme (RU) ne sont pas affectés par la taille de la fenêtre, et croissent linéairement

en fonction de B_1 . Nous pouvons montrer¹⁷ que la récompense espérée par itération est $\frac{B_1}{B} \times (1 - \frac{1}{B})$, donc le gain espéré sur un horizon T est $T \times \frac{B_1}{B} \times (1 - \frac{1}{B})$.

D'autres types de tests ont été réalisés sur des bandits à bras interconnectés non simplifiés, où chaque bras b se voit affecter différents paramètres (p_b, r_b) [Candan *et al.*, 2013]; les stratégies classiques s'y voient moins discriminées.

4.5 Modèle en iles dynamique comme stratégie de bandit

Comme nous l'avons montré dans les chapitres précédents, une stratégie de sélection adaptative des opérateurs efficace peut être obtenue au moyen d'un modèle en iles dynamique. Ce modèle peut également être utilisé comme stratégie de bandit. Une ile représente un bras, et un individu représente une séquence d'actions. La matrice de transition \mathcal{M} du modèle détermine les probabilités de choix d'actions; cependant, contrairement à un algorithme de type *probability matching*, on détermine les probabilités d'effectuer chaque action relativement à la dernière action réalisée. Plutôt qu'un ensemble de B probabilités $\mathbb{P}[x_t = b]$, le modèle fait évoluer un ensemble de B^2 probabilités $\mathbb{P}[x_t = b | x_{t-1} = b']$ (pour tout couple $(b, b') \in \mathcal{B}^2$)¹⁸. La particularité du modèle par rapport aux stratégies classiques nécessite de faire évoluer parallèlement une population d'individus; ce qui revient à effectuer simultanément plusieurs séquences d'actions.

Dans le cas de bandits classiques, analyser plusieurs séquences simultanées permet d'affiner bien plus rapidement l'estimation des valeurs d'action; ce qui représente donc un net avantage ne permettant pas de comparaison directe avec les autres stratégies, et s'éloigne de la problématique initiale du bandit. Néanmoins, dans l'hypothèse d'utiliser des modèles de bandits spécifiques — comme les bandits à bras interconnectés — comme cadre de travail pour le développement de stratégies de sélection d'opérateurs pertinentes, tous les mécanismes permettant de récolter en moyenne des récompenses plus élevées présentent un intérêt.

Nous avons pu mesurer que le modèle en iles dynamique était capable de retrouver des séquences d'opérateurs pertinentes que les stratégies classiques n'étaient pas en mesure de mémoriser. Par exemple, sur un problème de bandits à bras interconnectés simplifié composé uniquement de B bras utiles avec une fenêtre $W \geq B$, le modèle parvient à régler automatiquement ses probabilités de migrations, de manière à ce qu'un ordre d'application des bras apparaisse — à condition de faire évoluer un nombre significatif d'individus. La figure 14 illustre ce point sur deux exemples, l'un avec un bandit à deux bras ($B = B_1 = 2$), l'autre avec un bandit à quatre bras ($B = B_1 = 4$) et dans les deux cas une fenêtre $W = 10$. Chaque graphique représente l'évolution de l'ensemble des B^2 probabilités de migrations (c'est-à-dire les probabilités d'actionner un bras j après avoir

¹⁷On suppose $t > W$. Soit $N_W(b) = \#\{x_{t'} = b\}_{t' \in \{t-W, \dots, t-1\}}$ le nombre d'actions du bras b au cours des W dernières actions. Calculons tout d'abord la récompense espérée moyenne d'un opérateur utile b . Celle-ci est indexée sur $N_W(b)$, or $\mathbb{P}[N_W(b) = k] = (\frac{1}{B})^k (1 - \frac{1}{B})^{W-k} \binom{W}{k}$, avec une récompense associée de $(1 - \frac{k}{W})$. Considérant que $N_W(b)$ est une variable aléatoire distribuée selon une loi binomiale de paramètres W et $\frac{1}{B}$, alors $\mathbb{E}[N_W(b)] = \frac{W}{B}$ et $\mathbb{E}[R_t | x_t = b] = \mathbb{E}[1 - \frac{N_W(b)}{W}] = 1 - \frac{1}{B}$. L'espérance de récompense par itération dépend ainsi simplement de la proportion de bras utiles : $\mathbb{E}[R_t] = \frac{B_1}{B} \times (1 - \frac{1}{B})$.

¹⁸De manière analogue, un algorithme de type *Q-learning* [Watkins, 1989; Watkins et Dayan, 1992] appliqué à un problème de bandit fera évoluer un ensemble de B^2 valeurs d'action estimées.

actionné un bras i). On y observe qu’à partir d’une certaine phase d’apprentissage — quasi immédiate pour $B = 2$, plus longue lorsqu’on augmente B —, B probabilités de transitions s’approchent de 1, tandis que les $(B - 1) \times B$ autres s’approchent de 0. Pour une meilleure compréhension, nous avons illustré la topologie du modèle lors de ces phases, qui s’apparente alors à un anneau unidirectionnel, avec de faibles probabilités d’effectuer d’autres migrations aléatoires. Précisons que n’importe quel ordre sur les îles peut être détecté, pour un rendement équivalent. Il faut néanmoins noter que cette stratégie n’est pas optimale. En effet, $(1, 2, 1, 2, 1, 2, 1, 2, \dots)$ apporte une récompense moyenne de 0.50, tandis que $(1, 2, 2, 1, 1, 2, 2, 1, 1, 2, \dots)$ apporte une récompense moyenne de 0.55.

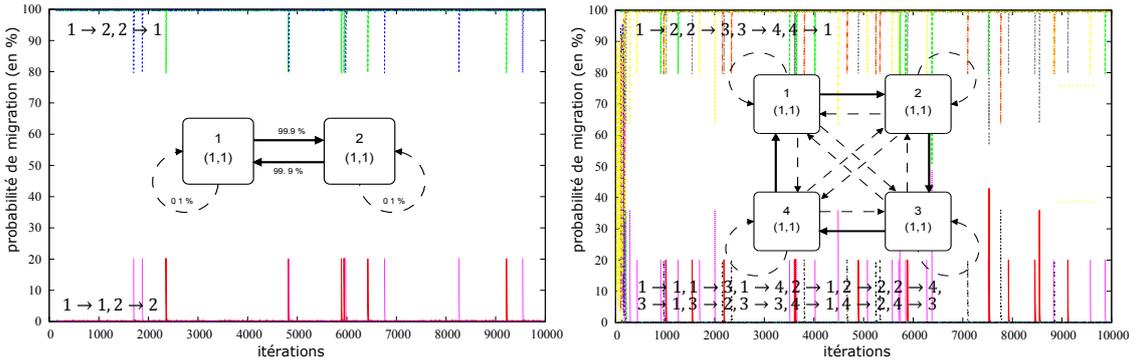


FIGURE 14 – Détection de séquences d’opérateurs avec un modèle en îles dynamique : cas stabilisé. Application aux bandits à bras interconnectés simplifiés, respectivement à 2 bras utiles (à gauche) et à 4 bras utiles (à droite). Paramètres du modèle en îles : $\alpha = 0.9$, $\beta = 0.001$, $100 \times B$ individus. Les flèches pleines représentent des migrations très probables (probabilité proche de 1), et les flèches en pointillé des transitions très peu probables (probabilité proche de 0).

Nous avons poursuivi l’expérimentation sur des bandits à davantage de bras (jusqu’à $B = 10$), et un tel circuit hamiltonien parvenait systématiquement à se constituer, mais après une phase d’apprentissage pouvant être relativement longue. Enfin, des tests supplémentaires sur des bandits à bras interconnectés non simplifiés nous ont indiqué que le modèle pouvait adapter périodiquement ses probabilités de migrations afin de définir des séquences prioritaires d’opérateurs plus complexes. Dans l’exemple de la figure 15, avec le bandit à deux bras tel que $p_1 = 0.7$, $r_1 = 0.3$, $p_2 = 0.4$, $r_2 = 0.6$, $W = 10$, la séquence d’opérateurs privilégiée par le modèle est $(2,1,2,1,2,1,2,2,2,2,2)$ pour une récompense moyenne de 0.11, proche de la récompense moyenne obtenue par un oracle myope (0.13). Pour obtenir un tel comportement oscillatoire, le modèle doit cependant être paramétré de manière à le rendre moins inerte, donc en baissant α et augmentant β .

Utilisant le modèle des bandits à bras interconnectés, ces derniers résultats indiquent qu’un mécanisme d’apprentissage basé sur une matrice de transition permet de retrouver des séquences d’opérateurs pertinentes. Il semble cependant évident que la mise à jour des politiques migratoires du modèle en îles telle que définie dans le chapitre 2 n’est pas optimisée pour assurer un équilibre idéal entre exploration et exploitation. Une perspective de notre travail sur les modèles en îles et la sélection adaptative des opérateurs à partir de bandits serait d’utiliser le modèle en îles comme une stratégie classique de bandit. Une première étape est de définir un mécanisme de mise à jour des probabilités de migra-

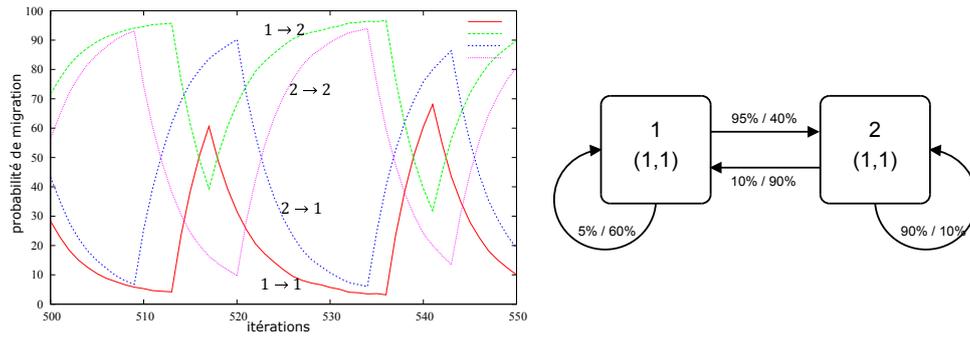


FIGURE 15 – Détection de séquences d’opérateurs avec un modèle en îles dynamique : cas oscillatoire. Application à un bandit à 2 bras interconnectés : bras $(0.7, 0.3)$ et $(0.4, 0.6)$, et $W = 10$. Paramètres du modèle en îles : $\alpha = 0.8$, $\beta = 0.01$, $100 \times B$ individus.

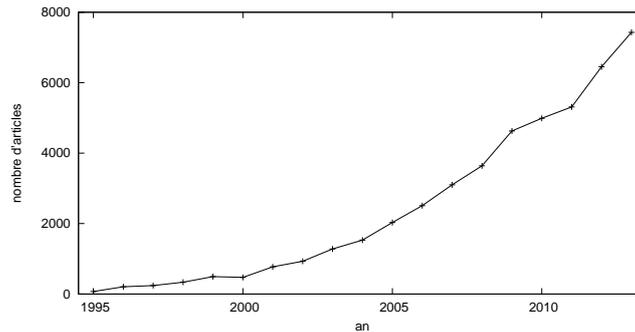
tions basé sur l’historique d’un unique individu. Dans un second temps et pour valider l’efficacité et l’intérêt de l’évolution des probabilités de choix d’opérateurs à partir d’une matrice de transition, il sera intéressant d’appliquer cette stratégie *en îles* à des bandits stochastiques (de type *restless*) régis par une matrice de Markov. Une régulation efficace des probabilités de transitions devrait conduire à apprendre du bandit afin d’inférer, non plus les valeurs d’action de chaque bras, mais le modèle de Markov caché. Nous pourrions alors poursuivre notre étude sur les bandits à bras interconnectés, que ce soit dans la détermination d’un oracle pour toutes les instances, ou dans la recherche d’une stratégie efficace sans connaissance *a priori* des caractéristiques du bandit. Enfin, affiner les liens entre sélection adaptative des opérateurs et bandits permettra d’établir des modèles d’abstraction plus pertinents, aidant à la conception de stratégies de sélection mieux adaptées et ainsi d’algorithmes autonomes plus efficaces.

Seconde partie

Escalader les paysages de fitness

Préambule

Comme l'indique le graphique ci-dessous, la conception et l'utilisation des métaheuristiques est grandissante depuis quelques années.



Nombre d'articles publiés par an mentionnant le terme *métaheuristique* (source : Google Scholar).

Pourtant, il nous semble que peu d'études apportent une réelle plus-value dans l'identification des mécanismes rendant la recherche par voisinage plus efficace. Un des points nous apparaissant particulièrement singulier est que les travaux sur les métaheuristiques de recherche locale, dont l'objectif est en premier lieu de déterminer les plus hautes solutions de l'espace de recherche, se focalisent presque exclusivement sur l'aspect perturbatif, donc sur les mouvements *descendants*. Cette seconde partie s'attache à déterminer les vecteurs d'efficacité des stratégies de recherche intensificatrices, en focalisant l'étude sur la famille d'algorithmes de recherche locale la plus basique, l'amélioration itérative (ou *hill-climbing*) dont le principe est d'établir une trajectoire *ascendante*, de solution en solution, dans le but d'atteindre une solution de la meilleure qualité possible sur une échelle d'évaluation.

Au moyen de larges études empiriques, nous tenterons d'obtenir une meilleure compréhension de la dynamique des stratégies d'intensification. Nous verrons à la fois qu'une simple technique de hill-climbing a la possibilité de mener la recherche vers les meilleures solutions de l'espace, mais aussi que les stratégies de mouvement ont un impact déterminant et prévisible quant à la qualité des solutions atteintes.

La réalisation de ce travail a été facilitée par l'étude de paysages de fitness, permettant de nous abstraire suffisamment de la problématique de résolution de problèmes et de la recherche par voisinage au sens large pour isoler celle liée à l'*ascension* dans l'espace de recherche, qui prend ici le sens d'*escalade*.

Chapitre 5

Définition et caractérisation des paysages de fitness

5.1 De la recherche par voisinage aux paysages de fitness

Recherche par voisinage : rappels et problématique

Résoudre un problème d'optimisation combinatoire revient à déterminer la (ou une) *solution optimale*, au sens d'une fonction objectif, parmi un (grand) ensemble discret fini de *solutions potentielles* appelé espace de recherche. Les espaces de recherche sont induits par les instances de problèmes traitées et sont, dans le cas des problèmes qui nous intéressent ici, de cardinalité exponentielle en la taille des instances. La problématique de résoudre de manière approchée un problème d'optimisation combinatoire avec un algorithme polynomial consiste donc à déterminer une stratégie permettant de n'examiner qu'un sous-ensemble de solutions de cardinalité polynomiale en la taille des instances. L'optimum de ce sous-ensemble sera alors la *solution approchée* déterminée par notre stratégie. Bien sûr, définir ce sous-ensemble *a priori* reviendrait à effectuer une simple restriction sur l'espace de recherche, ou un échantillonnage, et serait d'une faible efficacité. Au contraire, les algorithmes évolutionnaires construisent ce sous-ensemble itérativement en fonction des informations apprises au cours du processus ; c'est cette particularité qui les apparente à des techniques d'intelligence artificielle et même, en suivant notre propos de la partie précédente, des algorithmes *déjà* autonomes. Nous avons vu dans le chapitre 1 que les métaheuristiques parcouraient itérativement un sous-ensemble de solutions au moyen d'une relation de voisinage. Cela peut être représenté par un autre modèle d'abstraction avec les paysages de fitness.

L'utilisation d'une relation de voisinage, décrite par une fonction qui associe à chaque solution de l'espace de recherche un ensemble de solutions dites voisines, permet de *naviguer* dans un *paysage de recherche*, c'est-à-dire de parcourir de proche en proche un sous-ensemble de solutions. La relation de voisinage est généralement définie de sorte que la taille du voisinage de chaque solution soit suffisamment faible pour pouvoir être explorée — si besoin — exhaustivement. Grâce à l'exploration par voisinage, le choix de la prochaine solution de l'espace de recherche à évaluer est naturellement largement réduit. La sélection de la prochaine solution courante est dictée par la stratégie de mouvement et s'organise en

deux temps. La première décision est d'accepter ou de rejeter la solution proposée. Dans le cas d'une acceptation, c'est parmi les voisines de cette solution que la prochaine sera choisie ; dans le cas d'un rejet, c'est dans les voisines de la solution précédente. Ensuite, parmi le voisinage considéré, la seconde décision porte sur la solution à évaluer. En réalité ces deux étapes peuvent être regroupées en une seule, à savoir déterminer, parmi un voisinage que l'on aura exploré partiellement ou totalement, la solution sélectionnée pour le prochain pas de recherche. Naturellement, l'évaluation de la qualité des solutions voisines, à chaque itération, doit être un critère de décision essentiel dans le choix de la solution à sélectionner.

Paysages de fitness

Bien qu'aucune définition n'impose de critère strict pour définir une relation de voisinage, il est naturel de relier entre elles les solutions *proches*. Plus généralement, la distance de voisinage entre deux solutions, c'est-à-dire le nombre de mouvements nécessaires pour passer de l'une à l'autre à partir de la relation de voisinage, devrait dépendre de leur disparité : plus une solution est proche d'une autre en termes de distance de voisinage, plus elle devrait partager de propriétés structurelles communes¹⁹. De la même manière, on imagine que dans la plupart des problèmes combinatoires, il y a une corrélation substantielle entre l'évaluation des solutions et leurs propriétés structurelles. L'absence de l'un de ces deux prérequis rendrait de toute manière vaine toute exploration de l'espace de recherche par stratégie de voisinage, qui ne serait alors pas plus efficace qu'un échantillonnage. À partir de cette base, il est possible de faire une analogie entre la recherche par voisinage et la génétique. Un espace de recherche correspond à un ensemble de *génotypes* admissibles, une relation de voisinage décrit des *mutations* envisageables, tandis qu'une fonction d'évaluation rend compte du *phénotype* d'un individu. Le triplet (espace de recherche, voisinage, évaluation) peut ainsi être représenté par un triplet (génotype, mutation, phénotype), dont l'abstraction en paysages de fitness existe depuis le début des années 1930.

La notion de paysage de fitness, concept fondamental en biologie théorique, a été développée par le généticien Sewall Wright [1932]. Il s'agit de représenter visuellement les liens existant entre le génotype et le potentiel reproductif des individus. Dans cette représentation théorique des mécanismes évolutifs, Wright définit l'espace des génotypes par un hypercube, c'est-à-dire que les génotypes sont des chaînes de bits de longueur fixe, et les mutations sont représentées par des changements d'état de bits. Le potentiel reproductif (et de survie) d'un individu est représenté par une valeur de *fitness*, ou *hauteur*. En fonction de la répartition des valeurs de fitness, les paysages peuvent évoquer un ensemble de vallées ou de montagnes — bien que difficile à conceptualiser à partir d'hypercubes — sur lesquelles des populations d'individus grimpent aléatoirement au gré des mutations bénéfiques. Ces paysages sont alors décrits comme plus ou moins rugueux. Un paysage rugueux aura

¹⁹Précisons que les relations de voisinages sont définies en fonction de l'espace de recherche, c'est-à-dire de la représentation, ou *codage*, des solutions potentielles. Parfois, la représentation des solutions peut différer des solutions elles-mêmes, et ainsi les solutions proches syntaxiquement (relativement à leur codage) ne le seront pas nécessairement sémantiquement. On supposera ici que les solutions de l'espace de recherche ne font pas que représenter les solutions potentielles mais qu'elles *sont* les solutions potentielles, ce qui revient à considérer uniformément syntaxe et sens.

davantage de cimes, également appelées ici optimums locaux, points du paysage à partir desquels les individus ne peuvent plus être améliorés par une unique mutation.

D'un point de vue évolutionnaire, on peut aisément conceptualiser ce qui rend un paysage lisse ou rugueux. Gardons cette représentation en hypercube et supposons que chaque gène représente un bit ; il n'aura donc que deux états possibles, changeables par mutation génétique. Si l'état d'un gène particulier n'a aucune influence sur l'expression des autres gènes, alors toute mutation de ce gène aura le même effet quel que soit le reste du patrimoine génétique de l'individu. Par exemple, si le fait de muter le premier gène de 0 en 1 apporte une variation de fitness d'une unité quel que soit l'individu considéré (et nécessairement -1 pour la mutation inverse), alors tout mécanisme évolutif finira naturellement par fixer ce gène à 1, et ce dernier n'aura apporté au final aucune rugosité au paysage. En effet, à tout endroit du paysage, une mutation identique de ce gène aura toujours le même effet, et le potentiel d'amélioration des bits restants sera inchangé, avant ou après mutation. Si tous les gènes sont indépendants, alors le paysage de fitness ne présente aucune rugosité et l'optimum global est le seul optimum local du paysage²⁰. À partir de tout individu de départ, tout ordre d'application des mutations bénéfiques conduira à l'individu de plus haute valeur de fitness. Bien sûr, la plupart du temps les gènes sont en interaction avec un sous-ensemble de gènes, et l'effet de la mutation d'un gène sera conditionné par la valeur d'autres gènes. Ce phénomène est qualifié d'*épistasie* et la rugosité d'un paysage découle directement de ce phénomène. Plus les interactions entre les gènes sont nombreuses, moins la variation de fitness engendrée par la mutation d'un gène peut être clairement associée à la nouvelle valeur prise par ce gène.

Le concept de paysage de fitness a naturellement été repris en algorithmique évolutionnaire pour appréhender la résolution approchée de problèmes combinatoires [Mitchell *et al.*, 1992; Stadler, 1995; Merz et Freisleben, 1999; Ratle, 2001], car la correspondance entre solutions admissibles (individus) et évaluations est analogue à celle existant entre génotypes et fitness.

Utiliser l'abstraction des paysages de fitness pour décrire le comportement des algorithmes évolutionnaires, et en particulier de recherche locale, va nous permettre d'avoir une meilleure compréhension de la capacité d'une stratégie particulière à traiter une instance de problème spécifique. Tout d'abord, travailler sur un paysage plus rugueux rendra la recherche, et la découverte de bonnes solutions, plus difficile. Or il faut garder à l'esprit qu'une grande rugosité d'un paysage a nécessairement comme origine une faible corrélation entre génotype et fitness, cas où l'on observe régulièrement des différences de fitness importantes entre deux génotypes voisins. Ceci peut être naturellement induit par la difficulté de l'instance du problème, que l'on pourrait informellement définir comme la difficulté d'établir un lien fort entre la description d'une solution admissible et son évaluation. Une seconde possibilité est l'utilisation d'une relation de voisinage inadaptée, ne rendant pas suffisamment compte des proximités sémantiques entre solutions.

²⁰C'est le cas du problème One-Max présenté au chapitre 3.

5.2 Définitions

Afin de fixer les idées, nous définissons ci-après plus formellement la notion de paysage de fitness, ainsi que quelques concepts liés. Une grande partie du vocabulaire est logiquement commune avec celui utilisé en optimisation combinatoire et algorithmique évolutionnaire, puisqu'un simple couple (instance de problème combinatoire, relation de voisinage) suffit à définir un paysage de fitness.

Définition 3 (Paysage de fitness) *Un paysage de fitness est un triplet $(\mathcal{X}, \mathcal{N}, f)$, où \mathcal{X} est un ensemble de points²¹ appelé espace de recherche, $\mathcal{N} : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ est une fonction de voisinage²², et f une fonction de fitness. $f(x)$ est la fitness²³ (ou hauteur) d'un point $x \in \mathcal{X}$. $x' \in \mathcal{N}(x)$ est un voisin de x . $\mathcal{N}(x)$ est le voisinage de x .*

Définition 4 (Graphe de transition) *Le graphe de transition d'un paysage de fitness $\mathcal{P} = (\mathcal{X}, \mathcal{N}, f)$ est le graphe $(\mathcal{X}, \mathcal{A}_{(\mathcal{X}, \mathcal{N})})$, avec $\mathcal{A}_{(\mathcal{X}, \mathcal{N})} = \{(x, y) \in \mathcal{X}^2, y \in \mathcal{N}(x)\}$. On le notera $TG(\mathcal{P})$.*

Définition 5 (Distance) *La distance entre deux points x et y d'un paysage $\mathcal{P} = (\mathcal{X}, \mathcal{N}, f)$ correspond à la longueur du plus court chemin reliant x à y dans le graphe de transition associé. On pourra la noter $d_{\mathcal{P}}(x, y)$, $d_{\mathcal{A}_{(\mathcal{X}, \mathcal{N})}}(x, y)$ ou simplement $d(x, y)$.*

Nous introduisons également la notion de *graphe de transition contraint* associé à un paysage de fitness, comme un graphe composé uniquement de transitions satisfaisant un certain critère. Un graphe de transition contraint dépend d'un paysage de fitness et d'un critère de mouvement. Par exemple, on note $TG_{>}(\mathcal{P})$ le graphe $(\mathcal{X}, \{(x, y) \in \mathcal{A}_{(\mathcal{X}, \mathcal{N})}, f(y) > f(x)\})$, graphe de transition particulier issu de \mathcal{P} suivant le critère d'amélioration stricte.

La figure 16 illustre un paysage de fitness simple (au milieu). Le graphe de transition associé (à gauche) est le graphe directement construit à partir de la relation de voisinage ; on l'appelle également *graphe de voisinage*. Le graphe de transition contraint par le critère d'amélioration stricte (à gauche) comporte toutes les transitions valides au regard du critère de mouvement.

Les définitions des optimums d'un paysage de fitness sont naturellement identiques à celles d'usage en optimisation. Un *optimum local* (non strict) est un point $x \in \mathcal{X}$ tel que $\forall x' \in \mathcal{N}(x), f(x') \leq f(x)$. Un *optimum global* est un point $x^* \in \operatorname{argmax}_{x \in \mathcal{X}} f(x)$. Usuellement, les paysages de fitness sont définis dans un contexte de maximisation, mais ils peuvent être étendus aux problèmes de minimisation sans perte de généralité.

²¹Plutôt que *solutions* ou *individus*, nous privilégions la dénomination de *points* pour une plus grande abstraction. On parlera de solution lorsque le lien d'un paysage de fitness avec une instance de problème d'optimisation sera réalisé.

²²Si \mathcal{N} est une fonction qui décrit la relation de voisinage, il est d'usage de désigner \mathcal{N} directement comme la relation de voisinage, que ce soit dans le contexte des paysages de fitness ou de la recherche locale.

²³Au sein de la communauté francophone, il n'y a pas de consensus sur le genre du mot ; nous considérons pourtant ce terme comme francisé car il nous semble désormais plus adapté dans ce contexte spécifique que sa traduction la plus naturelle, *adaptation*.

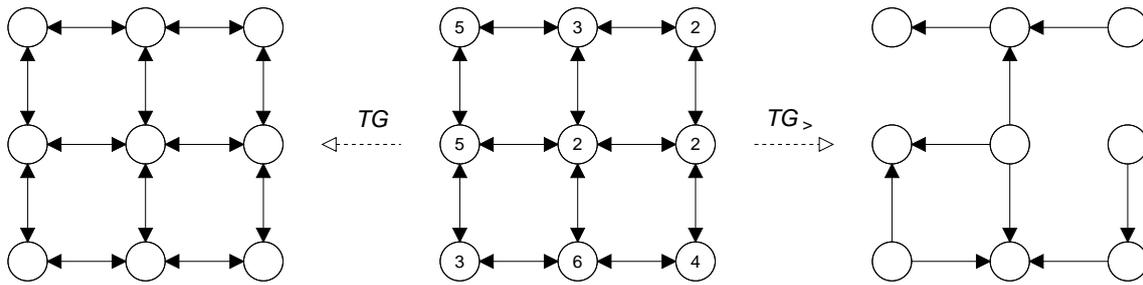


FIGURE 16 – Graphes de transition d’un paysage de fitness. À gauche, le graphe de transition non contraint associé au paysage de fitness exemple (au milieu). À droite, un graphe de transition contraint par le critère d’amélioration stricte.

5.3 Caractérisation des paysages de fitness

Au-delà de la simple étude des paysages de fitness combinatoires, ce chapitre vise à utiliser cette représentation alternative des instances de problèmes combinatoires pour parfaire notre compréhension des méthodes de voisinage traditionnelles, et permettre de définir de nouvelles stratégies de recherche efficaces. Ceci passe par une caractérisation des paysages de fitness, qui nous permettra dans un second temps d’établir des liens entre efficacité des méthodes de recherche locale et propriétés des paysages.

Les paysages de fitness sont caractérisables à partir de trois éléments essentiels : dimension, neutralité et rugosité. C’est dans l’évaluation de ces caractères que les indicateurs peuvent différer. En regardant la littérature, il est d’ailleurs intéressant de noter le nombre très important de travaux, notamment en biologie théorique, visant à caractériser les paysages et en particulier leur rugosité. Malan et Engelbrecht [2013] ont récemment publié une revue très détaillée sur la question. Notre propos ici est davantage de simplifier cette caractérisation, ou du moins de proposer des indicateurs appropriés à notre étude d’évaluation des méthodes de recherche locale, et rapidement estimables au sein même d’un algorithme de résolution.

5.3.1 Dimension

La dimension d’un paysage de fitness est déterminée par sa taille et sa connectivité. La *taille* d’un paysage se réduit simplement à la cardinalité de son espace de recherche, qui de manière évidente est un facteur déterminant de la difficulté à atteindre un point spécifique, en particulier l’optimum global. La *connectivité* d’un paysage dépend de la relation de voisinage et détermine, entre autres, les distances entre points. Les concepts classiques de la théorie des graphes sont autant d’indicateurs permettant de caractériser la connectivité d’un paysage : diamètre et densité du graphe de transition, distance moyenne entre solutions, *etc.*

5.3.2 Neutralité

Dans certains paysages de fitness, en particulier lorsque la fonction de fitness est faiblement discrétisée (taille du codomaine réduite), il peut exister une proportion non

négligeable de points voisins de fitness identique. C'est typiquement le cas lors de l'étude de problèmes d'optimisation combinatoires où le codomaine de la fonction objectif est de faible cardinalité, mais aussi de problèmes possédant une fonction objectif spécifique comme une fonction MinMax. Ce phénomène, appelé *neutralité*, a une grande influence sur le comportement des algorithmes de recherche [Smith *et al.*, 2002]. C'est ainsi que la neutralité constitue un élément prépondérant de caractérisation des paysages.

Nous rappelons ci-dessous quelques définitions relatives à la neutralité dans les paysages de fitness.

Définition 6 (Voisin neutre et degré de neutralité d'un point) *Un voisin neutre d'un point x est un point x' tel que $x' \in \mathcal{N}(x)$ et $f(x') = f(x)$. Le degré de neutralité ι de x dénote son taux de voisins neutres :*

$$\iota(x) = \frac{\#\{x' \in \mathcal{N}(x), f(x') = f(x)\}}{\#\mathcal{N}(x)} \quad (20)$$

Définition 7 (Plateau) *Un plateau (ou réseau neutre) est un ensemble de points connectés et de même fitness, c'est-à-dire un sous-graphe connexe induit par $\mathcal{S} \subseteq \mathcal{X}$ et tel que $\forall(x_i, x_j) \in \mathcal{S}^2, f(x_i) = f(x_j)$.*

Les indicateurs de neutralité utilisés dans [Reidys et Stadler, 1998; Vanneschi *et al.*, 2006] et discutés dans [Malan et Engelbrecht, 2013], se concentrent sur les réseaux neutres. Dans le but de comparer le comportement des méthodes de recherche locale sur des paysages de fitness présentant différents niveaux de neutralité, nous considérons prioritairement le taux de neutralité (défini ci-après) d'un paysage comme indicateur privilégié. En effet, ce sont uniquement les arêtes neutres du graphe de transition qui vont permettre de distinguer le comportement et l'efficacité des différentes politiques de sélection des mouvements neutres par les algorithmes de recherche locale. Comme nous le verrons ultérieurement, les graphes de transition contraints associés aux différents critères de transition neutre diffèrent uniquement par la présence ou l'absence de ces transitions neutres.

Définition 8 (Taux de neutralité) *Le taux de neutralité ν d'un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$ se définit par :*

$$\nu(\mathcal{X}, \mathcal{N}, f) = \frac{\#\{(x, y) \in \mathcal{A}_{(\mathcal{X}, \mathcal{N})}, f(x) = f(y)\}}{\#\mathcal{A}_{(\mathcal{X}, \mathcal{N})}} \quad (21)$$

Le taux de neutralité d'un paysage de fitness peut être estimé en échantillonnant son ensemble de transitions $\mathcal{A}_{(\mathcal{X}, \mathcal{N})}$. En particulier, si tous les points ont le même nombre de voisins — ce qui est fréquent dans la plupart des paysages de fitness —, le taux de neutralité est égal à la moyenne des degrés de neutralité de l'ensemble des points $x \in \mathcal{X}$:

$$(|\mathcal{N}(x)| = |\mathcal{N}(y)|, \forall x, y \in \mathcal{X}) \Rightarrow \nu(\mathcal{X}, \mathcal{N}, f) = \frac{1}{\#\mathcal{X}} \sum_{x \in \mathcal{X}} \iota(x) \quad (22)$$

Dans ce cas, le taux de neutralité peut être estimé en échantillonnant \mathcal{X} . Cet échantillonnage permet également d'estimer la dispersion des degrés de neutralité et apporte une

information supplémentaire sur la distribution de la neutralité dans le paysage. Cette méthode d'estimation de la neutralité a été employée notamment par Marmion *et al.* [2013].

5.3.3 Rugosité

En présentant la notion d'épistasie (section 5.1), nous avons vu que dans un contexte biologique, les interactions entre les gènes accroissent la complexité globale d'un paysage de fitness. Le phénomène d'épistasie signifie que l'effet d'une mutation est conditionné par la présence d'autres mutations [Bateson et Mendel, 1909]. Ici, nous nous intéressons au concept d'*épistasie de signe* [Poelwijk *et al.*, 2007], qui apparait lorsque des mutations sont bénéfiques sur certains individus — en fonction du reste de leur patrimoine génétique —, et défavorables sur d'autres (voir figure 17).



FIGURE 17 – L'épistasie se produit lorsque la présence d'une mutation (ici symbolisée par une transition $x \rightarrow y$) affecte l'effet d'une autre mutation ($x \rightarrow x'$, $y \rightarrow y'$). On observe une épistasie de signe si une mutation défavorable devient bénéfique sous condition d'une autre mutation (à gauche). Sur la partie droite, il n'y a pas d'épistasie de signe.

La notion de *rugosité* est une caractéristique essentielle dans la caractérisation de paysages de fitness. Cependant, il n'est pas évident d'établir une définition précise et formelle de la rugosité, de même qu'un indicateur pour l'évaluer (voir différents exemples de paysages de fitness représentés figure 18). Dans de nombreux travaux comme ceux de Barnett [1998], la rugosité d'un paysage est réduite à une mesure d'auto-corrélation. Dans leur revue, Malan et Engelbrecht [2013] distinguent le degré d'interdépendance des variables — autrement dit l'épistasie — et la rugosité, associée simplement au nombre et à la distribution des optimums locaux. Nous pensons au contraire qu'il est pertinent de lier directement le calcul de la rugosité d'un paysage à sa quantité d'interactions épistatiques car un nombre d'optimums locaux important implique nécessairement un haut niveau d'interdépendances entre variables, et ainsi une mesure précise de la difficulté d'escalader ces paysages. En outre, la rugosité des paysages de fitness — en tant que concept et non de métrique — est originellement présentée comme la conséquence de l'épistasie [Kauffman et Levin, 1987]. Dans un contexte de paysages combinatoires, une mesure de la rugosité devrait alors exprimer la corrélation *de signe* entre voisinage et fitness. Comme les relations de voisinage représentent ici par analogie des mutations génétiques, le signe des variations de fitness vont exprimer le caractère bénéfique ou défavorable d'un mouvement, représenté

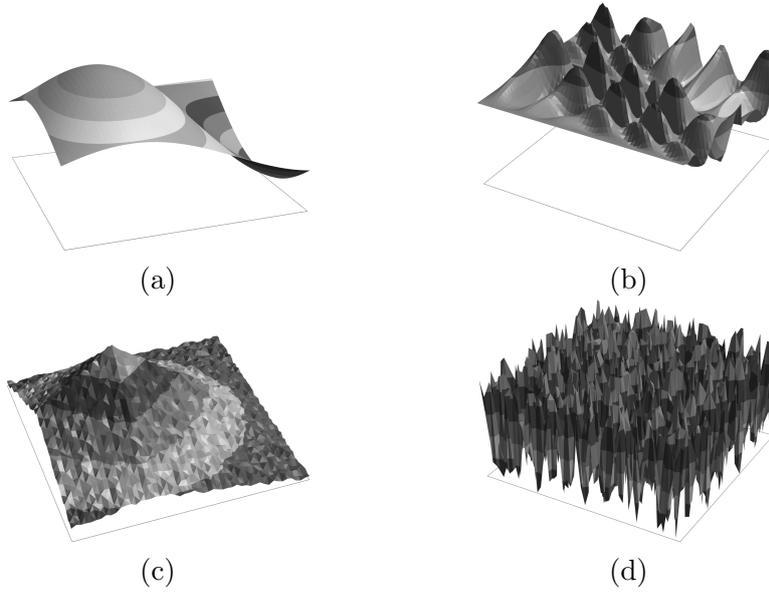


FIGURE 18 – Quatre différents paysages de fitness, avec différentes propriétés de rugosité : (a) lisse, (b) localement lisse mais globalement rugueux, (c) localement rugueux mais globalement lisse, (d) hautement rugueux (paysage à fitness aléatoires). Ces paysages-exemples ont été construits à partir d'un graphe de transition de type grille bidimensionnelle.

par une transition. Ceci nous a amené à définir une mesure de rugosité basée sur le concept d'échelle (définition 9 et figure 19) et l'indicateur d'épistasie de signe ξ (définition 10).

Définition 9 (*k*-échelle) On définit une *k*-échelle de sommets (x_0, x'_0, x_k, x'_k) comme un ensemble biparti de points d'un paysage $\{(x_0, \dots, x_k), (x'_0, \dots, x'_k)\}$ tel que $\forall i, j \in \{0, \dots, k\}, d(x_i, x_j) = d(x'_i, x'_j) = |i - j|$ et $d(x_i, x'_j) = |i - j| + 1$. On note \mathcal{L}_k l'ensemble de tous les quadruplets (x, x', y, y') étant des sommets de *k*-échelles, et $\mathcal{L} = \bigcup_k \mathcal{L}_k$.

Définition 10 (Indicateur d'épistasie de signe) Soit $(x, x', y, y') \in \mathcal{L}$. On définit l'indicateur d'épistasie de signe ξ de la manière suivante :

$$\xi^{\triangleright}(x, x', y, y') = \begin{cases} 1 & \text{si } (x' \triangleright x) \oplus (y' \triangleright y) & \text{(présence d'épistasie de signe)} \\ 0 & \text{sinon} & \text{(absence d'épistasie de signe)} \end{cases} \quad (23)$$

Ici, \oplus symbolise l'opérateur de disjonction exclusive, et \triangleright est un critère de comparaison ($\triangleright \in \{>, \geq\}$). Ces deux critères permettent deux interprétations de l'épistasie de signe, suivant que les fitness équivalentes (voisins neutres) sont considérées comme acceptables ou non — relativement au lien avec les paysages de recherche et la recherche locale. D'un côté, une recherche locale acceptant seulement des voisins strictement améliorants ($>$) ne distingue pas voisins neutres et voisins détériorants. D'un autre côté, considérant un critère de sélection qui accepte à la fois voisins améliorants et voisins neutres (\geq), ces derniers ne peuvent être considérés ni comme bénéfiques, ni comme défavorables, puisque

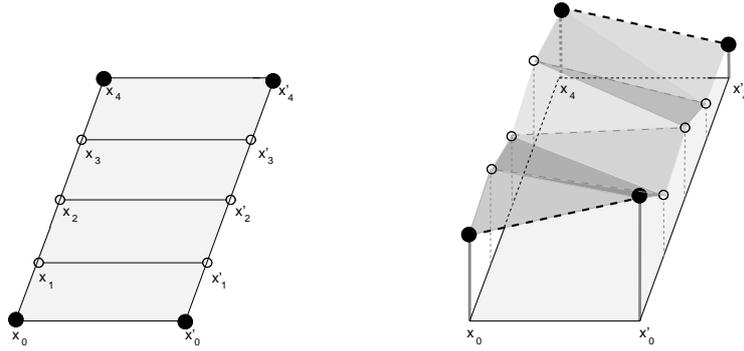


FIGURE 19 – Illustration d’une 4-échelle dans un graphe de transition (à gauche) et dans un paysage de fitness (à droite). Précisons que le cas d’une 1-échelle correspond aux illustrations de la figure 17.

le mouvement inverse est acceptable également. La table 5 indique la présence d’épistasie de signe pour les deux critères d’acceptation et les différentes configurations de variations de hauteur.

Δf	$f(x') - f(x)$	+	+	+	=	=	=	-	-	-
	$f(y') - f(y)$	+	=	-	+	=	-	+	=	-
Épistasie ($>$)		non	oui	oui	oui	non	non	oui	non	non
Épistasie (\geq)		non	non	oui	non	non	non	oui	non	non

TABLE 5 – Apparition de l’épistasie de signe en fonction de variations locales de fitness et considérant différents critères de comparaison ($>$, \geq).

L’indicateur ξ précédemment défini nous permet d’introduire l’indicateur de k -rugosité, qui correspond à la proportion d’épistasies de signe parmi les éléments de \mathcal{L}_k .

Définition 11 (k -rugosité) La k -rugosité d’un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$ est définie par :

$$\rho_k^{\triangleright}(\mathcal{X}, \mathcal{N}, f) = \frac{1}{\#\mathcal{L}_k} \sum_{q \in \mathcal{L}_k} \xi^{\triangleright}(q) \quad (24)$$

L’indicateur de k -rugosité, qui peut être estimé par échantillonnage, permet de caractériser un paysage de fitness au moyen d’un ensemble de valeurs. En particulier, la 1-rugosité ($k = 1$), qui correspond au taux d’épistasie de signe mesuré à partir de paires de mutations, sera appelée *rugosité locale*. Les autres valeurs de k -rugosité indiquent la rugosité du paysage à des échelles plus grandes, considérant l’effet sur la fitness d’une même mutation appliquée à des points distants de k (voir figure 20).

Les indicateurs de neutralité et de rugosité que nous venons de présenter considèrent tout l’espace de recherche uniformément, ce qui revient par conséquent à surreprésenter les points de fitness moyennes, largement majoritaires dans les paysages mais appartenant à des zones vite délaissées par les algorithmes de recherche locale qui explorent principalement les zones plus hautes, plus rares mais pouvant posséder des propriétés différentes. Les

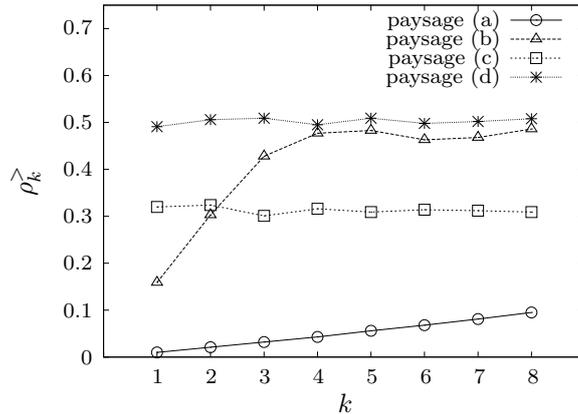


FIGURE 20 – k -rugosités des paysages illustrés sur la figure 18. Un paysage lisse (a) appelle de petites valeurs de rugosité. Sur le paysage (b), comportant de nombreuses cimes, la rugosité locale ($k = 1$) est faible mais la rugosité croît rapidement lorsque l’on augmente k . Au contraire, le paysage (c) présente une haute rugosité locale, mais le fait de présenter une unique large cime (malgré de nombreux optimums locaux) entraîne de plus faibles valeurs de rugosités pour $k \geq 3$ que le paysage (b). Un paysage totalement aléatoire (d) se traduit par des valeurs de k -rugosité avoisinant 0.5.

différences de propriétés structurales à différentes hauteurs des paysages de fitness sont régulièrement évoquées, notamment dans le cas de la neutralité [Smith *et al.*, 2002]. Nous discuterons ultérieurement d’indicateurs alternatifs effectuant l’échantillonnage à partir de points collectés au cours d’une recherche locale. Il s’agit cependant de mesures difficiles à établir, notamment en ce qui concerne la collecte des informations à partir de recherches autorisant les mouvements neutres. Par exemple, concernant l’estimation de la neutralité, nous proposons d’échantillonner séparément les degrés de neutralité des points collectés entre chaque mouvement améliorant, puis de leur affecter un poids logarithmique en la taille des sous-échantillons, ceci afin de réduire les biais engendrés par une stagnation de la recherche²⁴.

5.4 Construire des paysages de fitness

5.4.1 Paysages NK

Contrairement aux paysages de recherche combinatoires, la famille des paysages NK [Kauffman et Weinberger, 1989; Kauffman, 1993] est un modèle de construction de paysages de fitness indépendants de tout problème, et dont les propriétés peuvent être déterminées *a priori*. Ce modèle de paysages de fitness aléatoires provient de la communauté biologie théorique, mais le lien avec la difficulté de résolution des problèmes d’optimisation combinatoire et l’application à la recherche locale a été immédiatement établi [Weinberger, 1990].

²⁴Cette mesure de *neutralité observée* sera abordée dans le chapitre 8, section 8.3

Les paysages NK utilisent l'espace de recherche des chaînes de bits de taille fixée, et le simple flip comme relation de voisinage — deux points sont voisins si et seulement si leur distance de Hamming est égale à 1 — de manière à ce que les graphes de transition NK soient des hypercubes. Les caractéristiques d'un paysage NK sont déterminées par deux paramètres N et K . N est la longueur des chaînes de bits, et définit ainsi la taille de l'espace de recherche ($|\mathcal{X}| = 2^N$) et celle du voisinage ($|\mathcal{N}_{\text{flip}}(x)| = N, \forall x \in \mathcal{X}$). $K \in \{0, \dots, N-1\}$ spécifie le niveau d'interdépendance des variables, qui affecte directement la rugosité du paysage. En effet, les valeurs de fitness des points sont données par une somme de N termes, chacun dépendant de la valeur de $K+1$ de ses bits. Ainsi, en faisant varier la valeur de K de 0 à $N-1$, les paysages NK peuvent être réglés de lisses à rugueux. En particulier, si $K=0$, alors le paysage contient uniquement un optimum local (l'optimum global) et revient à un problème One-Max ; à la seule différence que la solution optimale ne maximise pas nécessairement le nombre de *uns*. Ce type de paysages sans épistasie, ou avec une très faible épistasie, est dit de type *Mont Fuji* [Aita *et al.*, 2000]. Au contraire, régler K à $N-1$ amène une affectation totalement aléatoire des fitness, et un paysage dit de type *château de cartes*, en référence au modèle de paysages de fitness continus proposé par Kingman [1978]. Celui-ci est basé sur l'idée que toute mutation génétique mène à une nouvelle valeur de fitness non corrélée avec celle de son génotype parent (*house of cards model*).

Dans les paysages NK, la fonction de fitness $F_{\text{NK}} : \{0, 1\}^N \rightarrow [0, 1[$ à maximiser peut se définir formellement de la manière suivante :

$$F_{\text{NK}}(x) = \frac{1}{N} \sum_{i=1}^N C_i(x_i, \Pi_i(x)) \quad (25)$$

x_i est le i -ème bit de l'individu symbolisé par le point $x \in \{0, 1\}^N$. Chaque (sous-) fonction Π_i définit les dépendances du bit i , avec $\Pi_i(x) = \{x_{\pi_1(i)}, \dots, x_{\pi_K(i)}\}$ tel que $\forall j \in \{1, \dots, K\}, \pi_j(i) \in \{1, \dots, N\} \setminus \{i\}$ et $\forall j' \neq j, \pi_{j'}(i) \neq \pi_j(i)$. Chaque (sous-) fonction $C_i : \{0, 1\}^{K+1} \rightarrow [0, 1[$ définit la valeur contributive de x_i (*fitness-contribution*) relativement à son ensemble de dépendances $\Pi_i(x)$.

Précisons que parmi les quelques définitions des paysages NK coexistant dans la littérature évolutionnaire (par exemple [Skellett *et al.*, 2005; Marmion *et al.*, 2011a; Daolio *et al.*, 2012]), aucune, de notre point de vue, n'est formellement précise. Cette nouvelle définition vise à réparer ce constat. Cependant, pour plus de simplicité et une meilleure compréhension, nous proposons d'appliquer le calcul de F_{NK} sur un exemple concret (voir ci-après).

Au final, un paysage NK est décrit par un triplet $(\{0, 1\}^N, \mathcal{N}_{\text{flip}}, F_{\text{NK}})$. Les instances NK sont déterminées par les $(K+1)$ -uplets $(x_i, x_{\pi_1(i)}, \dots, x_{\pi_K(i)})$ et une matrice de contributions C regroupant les $2^N \times (K+1)$ fitness-contributions possibles. Un paysage NK est ainsi défini en intension par une instance NK. Précisons que dans les paysages NK standards à *voisines aléatoires* [Kauffman, 1993], que nous utiliserons dans la suite du manuscrit, les dépendances entre les variables et les fitness-contributions sont générées aléatoirement selon une distribution uniforme. Le terme voisinage s'applique ici aux ensembles de dépendances entre bits. Une seconde alternative consiste à construire des ensembles de dépendances équilibrés et selon une règle déterministe, où chaque bit est en

correspondance avec ses K plus proches bits, à la manière d'un graphe de type anneau bi-directionnel d'ordre $K/2$ (*voisinages d'adjacence* [Kauffman, 1993]). D'après Ochoa *et al.* [2010], ces cas particuliers n'entraînent pas de différences de propriétés notables en termes de rugosité ou, par interprétation, de difficulté à être escaladés par des recherches locales stochastiques.

Exemple

Un exemple d'instance NK, pour $N = 8$ et $K = 2$, est donné table 6.

À chaque bit $i \in \{1, \dots, N\}$ ($N = 8$) est associé un ensemble de $K = 2$ bits (dépendances), permettant de définir N ($K + 1$)-uplets (ici 8 triplets). La matrice de contributions, de taille $2^N \times (K + 1)$ (ici 8×8) est constituée de fitness-contributions aléatoires, chacune prenant sa valeur à six décimales dans l'intervalle $[0.000000, 0.999999]$.

Bit # (i)	1	2	3	4	5	6	7	8
Dépendances Π_i	{3, 5}	{6, 8}	{5, 7}	{3, 6}	{3, 4}	{1, 8}	{1, 2}	{5, 6}
$(i, \pi_1(i), \pi_2(i))$	(1, 3, 5)	(2, 6, 8)	(3, 5, 7)	(4, 3, 6)	(5, 3, 4)	(6, 1, 8)	(7, 1, 2)	(8, 5, 6)
000	0.873764	0.966538	0.087415	0.721535	0.583928	0.532101	0.783181	0.077260
001	0.363020	0.543467	0.905829	0.920379	0.485074	0.660761	0.939623	0.104722
010	0.692868	0.883650	0.362665	0.442389	0.198575	0.338221	0.335135	0.293830
011	0.963021	0.456654	0.337281	0.310938	0.706982	0.646313	0.185540	0.599522
100	0.894493	0.365629	0.847824	0.366619	0.475664	0.354641	0.482327	0.245853
101	0.869927	0.179828	0.247594	0.390216	0.968644	0.675750	0.172104	0.110728
110	0.132720	0.430716	0.420294	0.040307	0.582080	0.498306	0.147929	0.527585
111	0.081548	0.476541	0.396649	0.343934	0.736206	0.204658	0.066980	0.847942

TABLE 6 – Exemple d'instance NK.

Soit le point $x = 11110000$. La fitness $F_{\text{NK}}(x)$ se détermine en moyennant les fitness-contributions de ses bits, c'est-à-dire une valeur par colonne, en respectant les dépendances et les valeurs des bits. Ici, la fitness-contribution du premier bit est la valeur de la première colonne correspondant à la valeur du triplet de bits $(x_1, x_3, x_5) = 110$, soit 0.132720. Les fitness-contributions suivantes dépendent ainsi des valeurs des triplets de bits suivants, soit respectivement $(x_2, x_6, x_8) = 100$ (0.365629), $(x_3, x_5, x_7) = 100$ (0.847824), $(x_4, x_3, x_6) = 110$ (0.040307), $(x_5, x_3, x_4) = 011$ (0.706982), *etc.* Ces fitness-contributions sont indiquées en gras dans la table 7. Finalement, $F_{\text{NK}}(11110000) = \frac{2.694483}{8} = 0.336810$.

Bit # (i)	1	2	3	4	5	6	7	8
Dépendances Π_i	{3, 5}	{6, 8}	{5, 7}	{3, 6}	{3, 4}	{1, 8}	{1, 2}	{5, 6}
$(i, \pi_1(i), \pi_2(i))$	(1, 3, 5)	(2, 6, 8)	(3, 5, 7)	(4, 3, 6)	(5, 3, 4)	(6, 1, 8)	(7, 1, 2)	(8, 5, 6)
000	0.873764	0.966538	0.087415	0.721535	0.583928	0.532101	0.783181	0.077260
001	0.363020	0.543467	0.905829	0.920379	0.485074	0.660761	0.939623	0.104722
010	0.692868	0.883650	0.362665	0.442389	0.198575	0.338221	0.335135	0.293830
011	0.963021	0.456654	0.337281	0.310938	0.706982	<u>0.646313</u>	0.185540	0.599522
100	0.894493	0.365629	0.847824	0.366619	0.475664	0.354641	0.482327	0.245853
101	0.869927	<u>0.179828</u>	0.247594	0.390216	0.968644	0.675750	0.172104	0.110728
110	0.132720	0.430716	0.420294	0.040307	0.582080	0.498306	0.147929	0.527585
111	0.081548	0.476541	0.396649	0.343934	0.736206	0.204658	0.066980	0.847942

TABLE 7 – Fitness-contributions utilisées pour le calcul de $F_{\text{NK}}(11110000) = 0.336810$ (en gras) et $F_{\text{NK}}(11110001) =$ (soulignées).

Soit le point $x' = 11110001$, voisin de x car différent sur la valeur d'un seul bit (x_8). Par conséquent, les seules fitness-contributions différentes, outre celle de x_8 , seront celles des bits avec qui x_8 est en dépendance — c'est-à-dire x_2 et x_6 . Précisons que dans le modèle à voisinages aléatoires, les dépendances ne sont pas nécessairement réciproques, à savoir que $j \in \Pi_i$ n'implique pas $i \in \Pi_j$. Les fitness-contributions utilisées pour le calcul de $F_{\text{NK}}(x')$ sont soulignées dans la table 7. On obtient $F_{\text{NK}}(11110001) = \frac{2.985367}{8} = 0.373171$. Comme $F_{\text{NK}}(x') > F_{\text{NK}}(x)$, il apparaît que x' est plus haut que x dans le paysage de fitness.

Puisqu'un bit apparaît dans $K + 1$ ensembles de dépendances en moyenne, chaque modification de la valeur d'un bit change la valeur de $K + 1$ fitness-contributions en moyenne. On remarque ainsi qu'à N constant, plus K est grand, plus les fitness des points voisins ont tendance à fluctuer. De la même manière, à K constant, augmenter N permet d'augmenter la proportion moyenne de fitness-contributions inchangées par la modification de la valeur d'un bit, et ainsi de rendre les variations de fitness entre voisins moins importantes.

5.4.2 Incorporer de la neutralité aux paysages NK

La précision des fitness-contributions générées aléatoirement fait que la neutralité est quasiment absente des paysages NK classiques. Plusieurs modèles — NKp, NKq, NKr — permettent cependant d'ajouter de la neutralité aux paysages NK en modifiant les fitness soit en amont dans l'établissement des fitness-contributions de l'instance NK — c'est le cas des modèles NKp et NKq — soit directement à partir des valeurs de fitness finales — modèle NKr. Ces modèles de paysages NK neutres ne modifient pas les dimensions des paysages (taille et connectivité).

Paysages NKp

Les *paysages NK probabilistes* [Barnett, 1998] sont construits à partir d'instances particulières dans lesquelles les matrices de contributions sont plus ou moins creuses. Durant la génération d'un paysage NKp, les valeurs de C sont fixées à 0 avec une probabilité de p , les autres étant générées aléatoirement sur $[0, 1[$. Un paysage NKp est alors déterminé par trois paramètres : N , K et p . Dans les cas extrêmes, des paysages NK classiques sont obtenus pour $p = 0$, et des paysages entièrement plats pour $p = 1$; dans ce cas, tous les points ont une fitness de 0. Une instance de paysage NKp obtenue à partir de l'instance NK standard de la table 6, pour $p = 0.5$, est présentée table 8.

Paysages NKq

Pour ajouter de la neutralité aux paysages NK, Newman et Engelhardt [1998] ont introduit dans des travaux simultanés à ceux des paysages NKp les *paysages NK quantifiés*, qui fixent le niveau de discrétisation des fitness-contributions. L'idée est d'augmenter le nombre de voisins neutres en limitant les valeurs de fitness-contributions possibles. Les paysages NKq disposent ainsi d'un paramètre entier additionnel $q \geq 2$ qui spécifie la cardinalité du codomaine des fonctions C_i . Le degré de neutralité maximum pour un NKq est atteint lorsque $q = 2$ — dans ce cas, C est une matrice binaire — et décroît lorsqu'on augmente q . L'instance de paysage NKq obtenue à partir de l'instance NK standard de la

Bit # (i)	1	2	3	4	5	6	7	8
Dépendances Π_i	{3, 5}	{6, 8}	{5, 7}	{3, 6}	{3, 4}	{1, 8}	{1, 2}	{5, 6}
$(i, \pi_1(i), \pi_2(i))$	(1, 3, 5)	(2, 6, 8)	(3, 5, 7)	(4, 3, 6)	(5, 3, 4)	(6, 1, 8)	(7, 1, 2)	(8, 5, 6)
000	0	0	0	0	0.583928	0	0.783181	0
001	0.363020	0	0	0	0	0	0.939623	0.104722
010	0.692868	0	0	0	0.198575	0	0	0.293830
011	0.963021	0.456654	0.337281	0.310938	0	<u>0.646313</u>	0.185540	0
100	0.894493	0	0	0.366619	0	0.354641	0.482327	<u>0</u>
101	0	<u>0</u>	0.247594	0	0.968644	0	0.172104	<u>0</u>
110	0.132720	0	0	0.040307	0.582080	0	0.147929	0
111	0	0.476541	0	0.343934	0	0	0.066980	0

TABLE 8 – Instance NKp construite à partir de l’instance NK de la table 6, avec le paramètre $p = 0.5$. À titre de comparaison, on indique également les fitness-contributions utilisées pour le calcul de $F_{\text{NK}}(11110000) = 0.044820$ (en gras) et $F_{\text{NK}}(11110001) = 0.125610$ (soulignées).

table 6, pour $q = 2$, est présentée table 9. Contrairement à la transformation $\text{NK} \rightarrow \text{NKp}$, la transformation d’une instance NK en instance NKq est déterministe. Naturellement, une instance NKp ou NKq peut être construite directement, sans se baser sur une instance NK déjà définie.

Bit # (i)	1	2	3	4	5	6	7	8
Dépendances Π_i	{3, 5}	{6, 8}	{5, 7}	{3, 6}	{3, 4}	{1, 8}	{1, 2}	{5, 6}
$(i, \pi_1(i), \pi_2(i))$	(1, 3, 5)	(2, 6, 8)	(3, 5, 7)	(4, 3, 6)	(5, 3, 4)	(6, 1, 8)	(7, 1, 2)	(8, 5, 6)
000	1	1	0	1	1	1	1	0
001	0	1	1	1	0	1	1	0
010	1	1	0	0	0	0	0	0
011	1	0	0	0	<u>1</u>	<u>1</u>	0	1
100	1	0	<u>1</u>	0	0	0	0	<u>0</u>
101	1	<u>0</u>	0	0	1	1	0	0
110	<u>0</u>	0	0	0	1	0	0	1
111	0	0	0	0	1	0	0	1

TABLE 9 – Instance NKq construite à partir de l’instance NK de la table 6 avec le paramètre $q = 2$. On indique les fitness-contributions utilisées pour le calcul de $F_{\text{NK}}(11110000) = 0.250000$ (en gras) et $F_{\text{NK}}(11110001) = 0.375000$ (soulignées).

Paysages NKr

Nous avons récemment proposé de créer artificiellement de la neutralité dans les paysages NK en tronquant la fonction de fitness d’un modèle de paysages NK standards [Basseur et Goëffon, 2013]. Dans ces *paysages NK arrondis*, un paysage NK originel est divisé en un nombre prédéfini de valeurs (niveaux) de fitness. À l’inverse des paysages NKq où ce sont les fitness-contributions qui sont discrétisées, les paysages NKr discrétisent seulement la fonction de fitness dans sa globalité, avec des fitness-contributions inchangées. Ainsi, contrairement aux paysages NKp et NKq, NKr n’est pas une généralisation du modèle NK dans laquelle la matrice de contributions dépend d’un paramètre de neutralité. En affectant uniquement les valeurs de fitness finales, le paramètre r n’entre aucunement en compte dans la politique de génération des matrices de contributions :

$$F_{\text{NKr}}(x, r) = \frac{\lfloor r \cdot F_{\text{NK}}(x) \rfloor}{r} \quad (26)$$

Dans ce modèle, on a la propriété suivante :

$$F_{\text{NK}}(x_1) < F_{\text{NK}}(x_2) \Rightarrow F_{\text{NKr}}(x_1, r) \leq F_{\text{NKr}}(x_2, r) \quad (27)$$

Ceci implique que chaque optimum global d'une instance de paysage NK est également un optimum global de toute instance de paysage NKr correspondant, décrite à partir de la même matrice de contributions.

5.4.3 Instances de problèmes combinatoires et paysages de fitness

Un paysage de recherche envisagé lors de la résolution d'instances de problèmes combinatoires (étant donné un espace de recherche, une relation de voisinage et une fonction d'évaluation) peut être vu comme un paysage de fitness. L'objet étudié est le même ; nous insistons simplement sur le point qu'une fois défini, un paysage de fitness est totalement déconnecté du problème, de la représentation des solutions ou encore de la relation de voisinage. Le paysage de fitness est une surface multidimensionnelle dont nous pouvons étudier les propriétés structurelles et le comportement des stratégies visant à atteindre de hauts optimums locaux. C'est avec cette vue que les paysages de fitness définissent un niveau d'abstraction supplémentaire par rapport aux paysages de recherche utilisés en algorithmique évolutionnaire — bien qu'il s'agisse, comme nous l'avons mentionné, exactement des mêmes objets.

Néanmoins, puisque le paysage de fitness est situé à un niveau d'abstraction supérieur, préserver la notion originelle de fitness impliquerait de normaliser les évaluations du paysage de recherche dans l'intervalle $[0,1]$, ce qui ne change rien à la problématique d'escalade. Dans les paysages de fitness dérivés d'instances de problèmes combinatoires que nous utiliserons, une telle normalisation ne sera pas effectuée. Les fitness des points atteints permettront alors de mieux estimer la qualité des solutions correspondantes, à partir de connaissances supplémentaires sur les instances de problème utilisées. Cela sort du cadre des paysages de fitness en tant que modèle d'abstraction, mais peut permettre d'utiliser les résultats expérimentaux dans des études dédiées aux problèmes correspondants. Les fonctions de fitness utilisées dans ces paysages dérivés d'instances de problèmes combinatoires (X, f) seront les fonctions objectif f . La seule exception viendra des problèmes de minimisation, où les fonctions de fitness choisies seront simplement les opposées des fonctions objectif afin de conserver le principe d'escalade sans perdre les propriétés structurelles des instances de problèmes.

Dans cette section, nous présentons trois types de paysages de fitness définis à partir de problèmes d'optimisation combinatoire parmi les plus étudiés : MAXSAT, Flowshop et QAP. Avec les paysages NK, ces derniers seront utilisés pour comparer les stratégies d'escalade dans les chapitres suivants.

MAXSAT

Le problème de satisfiabilité (SAT) consiste à déterminer une instanciation de n variables booléennes ($x_i \in \mathbb{B} = \{\perp, \top\}$) permettant de satisfaire une formule booléenne exprimée sous forme normale conjonctive. Le problème d'optimisation MAXSAT consiste à satisfaire un nombre maximum de clauses de la formule. Rappelons qu'une *clause*

$c_j = \bigvee_{k=1}^{n_j} l_{jk}$ est une disjonction de littéraux l_{jk} , et qu'un *littéral* est une variable booléenne (x_i) ou sa négation ($\neg x_i$).

Étant donnée une instance MAXSAT $\{c_1, \dots, c_m\}$ avec n variables et m clauses, la valeur objectif d'une instantiation $x \in \mathbb{B}^n$ est donnée par :

$$V(x) = \sum_{j=1}^m v(x, c_j), \text{ avec } v(x, c_j) = \begin{cases} 1, & \text{si } c_j \text{ est vraie selon } x \\ 0, & \text{sinon} \end{cases} \quad (28)$$

Très souvent, MAXSAT est traité comme un problème de minimisation, en employant la fonction objectif $m - V$, qui correspond au nombre de clauses non satisfaites. Afin de conserver cet objectif de clauses non satisfaites plus usuel, tout en l'exprimant comme problème de maximisation, nous utiliserons la fonction opposée $V - m$ comme fonction de fitness F_{SAT} .

Précisons enfin que les solutions MAXSAT sont usuellement représentées par des chaînes de bits ($0 \equiv \perp, 1 \equiv \top$), ce qui permet de les connecter naturellement avec le voisinage 1-flip ($\mathcal{N}_{\text{flip}}$). Un paysage de fitness dérivé d'une instance de problème MAXSAT sera donc noté $(\{0, 1\}^n, \mathcal{N}_{\text{flip}}, F_{\text{SAT}})$. On remarque que les paysages de fitness NK et MAXSAT ont des graphes de transition équivalents (hypercubes). La taille de ces paysages est de 2^n , et tout point $x \in \{0, 1\}^n$ possède n voisins.

Flowshop (FSP)

Le problème d'ordonnancement de type flowshop (*Flow-shop Scheduling Problem, FSP*) consiste à ordonner un ensemble de n travaux $\{J_1, J_2, \dots, J_n\}$ — plus communément désignés par le terme anglais *jobs* — sur m machines $\{M_1, M_2, \dots, M_m\}$. Les machines sont considérées comme des ressources critiques du fait que plusieurs jobs ne peuvent être affectés simultanément à une même machine. Chaque job J_i est composé de m tâches consécutives $\{t_{i1}, \dots, t_{im}\}$, où t_{ij} représente la j^{e} tâche du job J_i requérant la machine m_j . À chaque tâche t_{ij} est associé un temps de traitement p_{ij} . Nous considérons ici la variante la plus simple, le flowshop de permutation, où les jobs doivent être planifiés dans le même ordre sur toutes les machines ; c'est-à-dire qu'un job ne peut pas en dépasser un autre en arrivant sur une machine. L'intérêt de considérer cette variante du problème est qu'il nous permet de définir un paysage de fitness basé sur un espace de permutations.

Dans le flowshop de permutation, chaque tâche t_{ij} doit être programmée à l'instant s_{ij} de manière à respecter l'ordre π . Une fonction objectif naturelle et parmi les plus utilisées pour ce problème est la date de fin de la dernière tâche, définie par :

$$C_{\max}(\pi) = \max_{i \in \{1, \dots, n\}} (s_{im} + p_{im}) \quad (29)$$

Le problème de minimisation associé $\text{argmin } C_{\max}$ a été prouvé NP-difficile pour $m > 2$ [Lenstra *et al.*, 1977].

En considérant la fonction opposée $F_{\text{FSP}} = -C_{\max}$ comme fonction de fitness, la définition des paysages de fitness à partir d'instances du problème de flowshop consiste à adjoindre une relation de voisinage entre permutations. La taille des permutations (n) définit l'espace de recherche S_n , de taille $|S_n| = n!$ (S_n est l'ensemble de toutes les permutations de $\{1, \dots, n\}$). L'opérateur classique d'*insertion* permet l'utilisation d'une

relation de voisinage \mathcal{N}_{ins} et complète la définition du modèle de paysage de fitness $\text{FSP}_{\text{ins}} = (S_n, \mathcal{N}_{\text{ins}}, F_{\text{FSP}})$. L'opérateur d'insertion est le plus communément employé dans les algorithmes de voisinage définis pour la résolution du problème de flowshop de permutation avec optimisation de la date de fin de la dernière tâche [Taillard, 1990; Nowicki et Smutnicki, 1996; Ruiz et Stützle, 2007]. Il s'agit de déplacer un job d'une permutation π à une position différente. La taille du voisinage résultant est $|\mathcal{N}_{\text{ins}}(\pi)| = (n-1)^2, \forall \pi \in S_n$.

QAP

De par son aspect généralisateur de nombreux problèmes NP-difficiles classiques, le problème d'affectation quadratique (*quadratic assignment problem*, *QAP*) est considéré comme l'un des problèmes combinatoires NP-difficiles les plus difficiles [Sahni et Gonzalez, 1976; Pardalos et Wolkowicz, 1994], ayant en outre de nombreuses applications réelles (allocation de ressources, calcul distribué, analyse de données, *etc.*). QAP consiste à affecter des ressources à différents sites en fonction de distances entre sites et de flux entre ressources. Étant donné n ressources, n sites, et deux matrices $n \times n$ de distances et de flux D et F — où d_{ij} est la distance entre les sites i et j , et f_{rs} est le flux entre les ressources r et s —, la valeur objectif d'une affectation (ici une permutation) π correspond à la somme des produits entre les flux et les distances :

$$W_{\text{QAP}}(\pi) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} f_{\pi_i \pi_j} \quad (30)$$

π_i représente l'élément de position i dans la permutation $\pi \in S_n$. Ici, W_{QAP} doit être minimisé.

Le voisinage communément utilisé par les algorithmes d'optimisation dédiés, $\mathcal{N}_{\text{swap}}$, est basé sur l'opérateur *swap*, qui permet d'échanger deux éléments d'une permutation. Ainsi, $\text{QAP}_{\text{swap}} = (S_n, \mathcal{N}_{\text{swap}}, F_{\text{QAP}})$, avec $F_{\text{QAP}} = -W_{\text{QAP}}$, est un modèle de paysage de fitness naturellement dérivé du problème QAP. La taille du voisinage est $|\mathcal{N}_{\text{swap}}| = \binom{n}{2}, \forall \pi \in S_n$.

5.5 Paysages de recherche : panel d'instances et propriétés

Les instances de paysages de fitness utilisées dans les parties expérimentales des chapitres suivants seront tirées du panel de paysages introduit dans cette section. Il s'agit de paysages NK et de paysages définis à partir d'instances des problèmes combinatoires présentés dans la section précédente (MAXSAT, Flowshop, QAP).

Paysages NK

Nous avons généré 560 paysages NK classiques à partir de 56 paramétrages (N, K) différents. 10 instances NK ont été générées aléatoirement pour toute combinaison de paramètres $N = \{16, 24, 32, 64, 128, 256, 512, 1024\}$ et $K = \{1, 2, 4, 6, 8, 10, 12\}$. Ces instances seront dénotées $\text{NK}_{-[\text{N}]_{-}[\text{K}]_{-}[\text{i}]}$ où $[\text{N}]$ et $[\text{K}]$ désignent les valeurs des paramètres N et K , et $[\text{i}]$ l'identifiant de l'instance $i \in \{0, \dots, 9\}$. Par commodité, lorsqu'un seul paysage de chaque paramétrage (N, K) sera utilisé, $\text{NK}_{-[\text{N}]_{-}[\text{K}]}$ dénotera le paysage $\text{NK}_{-[\text{N}]_{-}[\text{K}]_{-}0}$.

En outre, chaque instance de paysage NK classique nous a servi de base pour la génération de 12 paysages NK neutres :

- 4 paysages NK_p, un pour chaque valeur de $p \in \{0.50, 0.80, 0.95, 0.99\}$ (du moins neutre au plus neutre).
- 4 paysages NK_q, un pour chaque valeur de $q \in \{10, 5, 3, 2\}$.
- 4 paysages NK_r, un pour chaque valeur de $r \in \{10000, 1000, 100, 10\}$.

Rappelons que les paysages NK_r utilisent l’instance du paysage NK standard inchangée, mais une fonction d’évaluation tronquée. Les instances NK_p et NK_q, au contraire, modifient les fitness-contributions d’origine.

Au total, ceci définit un ensemble de 7280 paysages NK, utilisant 728 paramétrages différents amenant tailles et niveaux de neutralité et de rugosité variés. Selon les études, plusieurs sous-ensembles de ces instances seront utilisés. En particulier, certains tests n’utiliseront qu’un exemplaire de paysage pour chaque couple (N, K) — dans ce cas, c’est l’instance NK-[N]-[K]-0 qui sera utilisée. De même, une grande partie des expérimentations seront centrées autour de quatre valeurs de K (1, 2, 4, 8). Les instances NK supplémentaires ont été ajoutées ultérieurement pour étendre une analyse en particulier. Les paysages de taille plus réduite ($N \in \{16, 24, 32\}$) ne seront utilisés que pour les études exactes du chapitre 6.

Paysages MAXSAT

Nous avons sélectionné 129 instances MAXSAT de la librairie SATLIB [Hoos et Stützle, 2000]. Cette librairie est composée de dizaines de milliers d’instances ; pour réduire ce nombre, nous n’avons inclus à notre ensemble qu’un exemplaire de chaque ensemble d’instances générées avec des jeux de paramètres identiques. Au final, nous distinguons 61 instances 3-SAT aléatoires, et 68 qui résultent de la transformation en SAT d’instances d’autres problèmes combinatoires. Les paysages sont définis à partir de ces 129 instances de problème MAXSAT et du voisinage 1-flip. Le choix de ne pas inclure des instances très larges de compétitions SAT récentes est motivé par l’objectif de notre étude. Les instances sélectionnées permettent ici de décrire des paysages de fitness particuliers en vue d’étudier le comportement de techniques d’escalade (recherches locales simples), et non de mettre à l’épreuve des algorithmes de résolution dédiés.

Paysages FSP_{ins}

Nous utiliserons 27 paysages de fitness FSP obtenus à partir de 27 instances du flowshop tirées de [Taillard, 1993], utilisant la fonction de fitness $F_{\text{FSP}} = -C_{\text{max}}$ et l’opérateur d’insertion comme relation de voisinage. Ces instances de problème sont communément utilisées pour mesurer l’efficacité des algorithmes de résolution du flowshop [Ruiz et Maroto, 2005] ; elles sont notées [n]-[m]-[i], désignant les valeurs des paramètres n (nombre de jobs), m (nombre de machines) et un numéro d’instance i . Les noms d’instances sont éventuellement suivis d’un autre identifiant à trois chiffres (taXXX pour identifiant Taillard).

La sélection des 27 instances s’est réalisée en conservant toutes les premières instances ($i = 1$) des paramétrages tels que $n \leq 200$ sur la collection de benchmarks sans incertitude disponible en ligne²⁵.

Paysages QAP_{swap}

QAPLIB [Burkard *et al.*, 1997] est une librairie d’instances de test pour le problème du QAP, comportant 137 instances provenant de 16 publications. Nous avons conservé celles respectant $n \leq 100$, plus représentatives pour nos études, et supprimé le paysage totalement plat induit par l’instance spécifique *esc16f*, ne présentant aucun intérêt dans notre cas. Au final, nous utiliserons 132 paysages de fitness, dont les noms sont composés des trois premières lettres de l’auteur de l’instance, suivi du nombre n et éventuellement d’une lettre supplémentaire permettant de distinguer les instances. Les paysages sont définis à partir de ces instances et du voisinage $\mathcal{N}_{\text{swap}}$.

Propriétés des instances

De par la diversité des problèmes et des instances de problèmes considérés, les instances de paysages utilisées dans nos études auront des caractéristiques variées en termes de dimensions, neutralités et rugosités. Ces caractéristiques, détaillées au long des chapitres suivants, sont résumées pour certains types d’instances dans la table 10. On remarque que les paysages NK, dont la taille, la rugosité et la neutralité — via les paysages NK neutres — sont paramétrables, permettent d’envisager une large gamme de paysages de caractéristiques différentes. Les paysages construits à partir d’instances de problèmes combinatoires ont logiquement des caractéristiques de rugosité et de neutralité moins variables. Les indices de neutralité et de rugosité sont estimés par échantillonnage, et représentent donc les caractéristiques moyennes des paysages de recherche, sans se focaliser sur des zones particulières, en particulier proches des optimums locaux. L’évaluation de la k -rugosité sera discutée ultérieurement.

Type de paysage	Taille du paysage	Taille du voisinage	Rugosité ($\rho_1^>$)	Neutralité (ν)
NK, NKp, NKq, NKr	$[2^{64}, 2^{1024}] \approx [10^{19}, 10^{308}]$	[64, 1024]	[< 0, 1%, 20%]	[0%, 95%]
MAXSAT	$[2^{20}, 2^{63624}] \approx [10^6, 10^{19153}]$	[20, 63624]	[< 0.01%, 2%]	[< 0.01%, 25%]
QAP	$[16!, 100!] \approx [10^{13}, 10^{158}]$	[120, 4950]	[2%, 24%]	[< 0.1%, 67%]
FSP	$[20!, 200!] \approx [10^{18}, 10^{375}]$	[361, 39601]	[< 2%, 13%]	[5%, 36%]

TABLE 10 – Aperçu des propriétés des instances de paysages utilisées. Les indicateurs de neutralité ν et de rugosité $\rho_1^>$ sont estimés par échantillonnage.

5.6 Stratégies d’escalade de paysages

Le propos de cette seconde partie de manuscrit s’attache à définir et comparer des stratégies basiques d’escalade de paysages de fitness. Par basique, nous entendons qu’il n’est à aucun moment permis de descendre, c’est-à-dire de passer à un point strictement

²⁵<http://www.lifl.fr/~liefooga/benchmarks/benchmarks/>

plus bas que le point courant. Nous ne perdons pas de vue qu'il s'agit, sans autre sophistication, de réaliser des recherches locales strictes (*hill-climbing*) sur des paysages de recherche. Cette abstraction en paysages de fitness nous permettra d'établir plus directement des liens entre stratégies d'escalade, capacité à atteindre les meilleurs optimums, et caractéristiques des paysages. De plus, restreindre l'étude sur ces seules recherches strictes permet de dissocier les stratégies d'intensification pure des techniques permettant de sortir des optimums locaux, non discutées dans ce manuscrit.

Climbers

Nous définissons un *climber*²⁶ comme une stratégie de recherche locale naviguant dans l'espace de recherche au moyen de mouvements non détériorants uniquement. Ceci laisse la liberté de plusieurs choix conceptuels, dont les effets sur la capacité à atteindre de bons optimums locaux ne sont pas toujours clairement établis. Plus généralement, nous identifions quatre niveaux conceptuels : la règle pivot, la gestion de la neutralité, l'évaluation du voisinage et l'exploration du voisinage.

Règle pivot. Une *règle pivot* [Yannakakis, 1990] peut être formellement décrite comme une fonction de choix qui affecte à chaque voisin une probabilité de sélection. À l'usage, dans les climbers, il s'agit plus simplement d'une règle déterministe ou non qui permet de choisir un voisin respectant un certain critère, relativement aux autres voisins et au point courant. La règle du *meilleur améliorant* consiste à sélectionner, à chaque itération, le voisin de meilleure fitness. Ceci nécessite la génération du voisinage complet à chaque pas de la recherche, à moins qu'une évaluation incrémentale du voisinage soit envisageable. La règle du *premier améliorant* sélectionne au contraire le premier voisin envisagé qui satisfait la condition de mouvement, ce qui évite la génération systématique du voisinage et permet davantage d'options de conception, en particulier concernant la gestion de la neutralité. Ces deux règles pivot constituent les deux alternatives largement utilisées dans la littérature. D'autres règles peuvent cependant être envisagées ; ce point fera l'objet de la section 7.2.1. Un climber utilisant la règle du meilleur améliorant est souvent dénommé simplement *meilleur améliorant*, mais est aussi parfois appelé algorithme du *gradient discret* ou *climber glouton*.

Gestion de la neutralité. Un *climber basique* se déplace de point en point — selon une règle pivot — par améliorations strictes successives jusqu'à atteindre un optimum local ; il ne permet ainsi aucun mouvement neutre (mouvement vers un voisin neutre). La question des mouvements neutres peut être considérée pour sortir des optimums locaux sans pour autant effectuer de mouvement détériorant ; c'est la stratégie de la *perturbation neutre* (PN), possible lorsque le paysage de fitness contient une proportion non négligeable de transitions neutres. Une autre variante, le *climber stochastique* peut accepter indifféremment des voisins neutres ou améliorants à tout moment de la recherche, y compris avant d'avoir atteint un optimum local.

²⁶Bien que n'étant pas adepte des anglicismes, nous consentons après *fitness* à une nouvelle exception sur ce terme qui reviendra très régulièrement, les traductions *escaladeur* ou *montée* (par opposition à *descente*) nous semblant trop artificielles.

De nombreuses appellations coexistent entre les climbers que nous appelons ici basiques et stochastiques. Les climbers (ou descentes), sans autre précision, sont le plus souvent définis à partir du critère d'amélioration stricte mais également parfois avec un critère non strict [Pirlot, 1996] correspondant au climber stochastique; celui-ci a d'ailleurs été redéfini par Barnett [2001] avec une nouvelle appellation (*netcrawler*). Enfin, le climber stochastique peut parfois être confondu avec un *climber randomisé* [Hoos et Stützle, 2004], autorisant des mouvements détériorants avec une certaine probabilité, mais qui n'entre pas dans le cadre de notre propos sur les algorithmes de recherche locale non perturbatifs.

Évaluation du voisinage. Le cas d'arrêt naturel d'un climber est la détection d'un optimum local. Cette dernière est évidente en utilisant la stratégie du meilleur améliorant, dépendante d'autres choix conceptuels avec un premier améliorant basique, et conditionnelle avec un climber stochastique. Plus généralement, les optimums locaux peuvent être détectés en considérant une *évaluation exhaustive* du voisinage, soit en évaluant le voisinage complet incrémentalement, soit en générant successivement des voisins sans remise. Ne jamais considérer deux fois le même voisin avant d'avoir effectué un mouvement est une règle de bon sens, y compris en utilisant des climbers stochastiques où la recherche peut ne pas se terminer systématiquement en étant seulement conditionnée par la découverte d'un optimum local. Mais parfois, lorsque la représentation des solutions est complexe, lorsqu'un voisinage est très large ou implique un opérateur de mouvement spécifique, une évaluation exhaustive du voisinage peut être coûteuse voire impossible à mettre en place. Dans de tels cas de figure, mais aussi parfois par simple commodité, les voisins peuvent être générés *aléatoirement* avec remise; cela ne change aucunement la distribution des chemins de recherche empruntables par les climbers. En outre, dans notre abstraction en paysages de fitness, les considérations relatives à la représentation des solutions, l'opérateur de voisinage et le calcul de la fitness n'entrent pas en compte car relèvent de la couche opérationnelle (avant abstraction).

Exploration du voisinage. Techniquement, si l'on se réfère aux définitions proposées par Hoos et Stützle [2004], la stratégie du premier améliorant consiste à explorer le voisinage dans un ordre utilisant une même règle *déterministe* au cours de la recherche; par exemple, dans un espace de recherche constitué de chaînes de bits, de considérer les flips dans l'ordre des indices des bits. Le premier voisin satisfaisant la politique de mouvement est alors sélectionné. Une alternative, généralement préférée, consiste à explorer à chaque pas de recherche le voisinage dans un ordre *aléatoire*. Cette règle d'exploration réduit les risques de cycle pouvant apparaître lorsqu'un climber stochastique est combiné à une exploration déterministe du voisinage.

Ces quatre composants représentent selon nous les quatre points de choix apparaissant lors de la conception d'un climber. Nous pouvons séparer ces niveaux en deux couches, structurelle et technique, selon que les choix influent la capacité du climber à atteindre plutôt certains points que d'autres au cours de la recherche. La règle pivot et la politique de gestion de la neutralité, clairement, définissent une stratégie de recherche qui peut avoir une grande influence sur les hauteurs atteintes, et incidemment sur la qualité des solutions produites. Les stratégies d'évaluation et d'exploration du voisinage peuvent être imposées

par des considérations techniques — et donc opérationnelles — et n’ont d’influence, en moyenne, que sur la complexité de l’algorithme. En particulier, il est intéressant de noter qu’un climber basique de type premier améliorant utilisant une exploration déterministe du voisinage sera aussi efficace en moyenne qu’en considérant une exploration stochastique — bien que cette dernière permette logiquement au climber d’être itéré afin de diversifier la recherche. Ceci est suggéré par Hoos et Stützle [2004] et confirmé par des expérimentations que nous avons réalisées sur des paysages NK et qui ne seront pas détaillées dans ce manuscrit.

Il est à noter que la stratégie du premier améliorant pourrait relever à la fois de la couche structurelle et technique. L’aspect technique est qu’il permet de ne pas explorer systématiquement le voisinage avant de procéder à un mouvement. L’aspect structurel est qu’il sélectionne le *premier voisin* qui satisfait le critère de fitness ; ou plus exactement, en imposant l’exploration aléatoire du voisinage, un *voisin aléatoire*. En ce sens, la règle du premier améliorant se veut ici une stratégie de l’*améliorant aléatoire* ; c’est ainsi que nous l’entendrons tout au long du manuscrit. Dans notre contexte de paysage de fitness, c’est principalement son aspect structurel — quelle influence ce choix a-t-il sur la recherche — qui est retenu.

La figure 21 résume les différentes possibilités de conception d’un climber. Dans l’élaboration d’une (méta)heuristique de recherche locale, une représentation spécifique des solutions, la structure du voisinage ou la particularité de la fonction d’évaluation peut amener à favoriser certains choix, même si dans la plupart des cas chaque option peut être envisagée. Ces considérations techniques sont induites par le fait que l’espace de recherche, la relation de voisinage et la fonction d’évaluation sont définis en intention. L’abstraction en paysages de fitness pourrait revenir à considérer notre triplet $(\mathcal{X}, \mathcal{N}, f)$ en extension ; si ce n’est factuellement, au moins conceptuellement. On peut dès lors distinguer les stratégies d’escalade de paysages de fitness par leur stratégie de recherche, définie par la règle pivot et la politique de gestion de la neutralité.

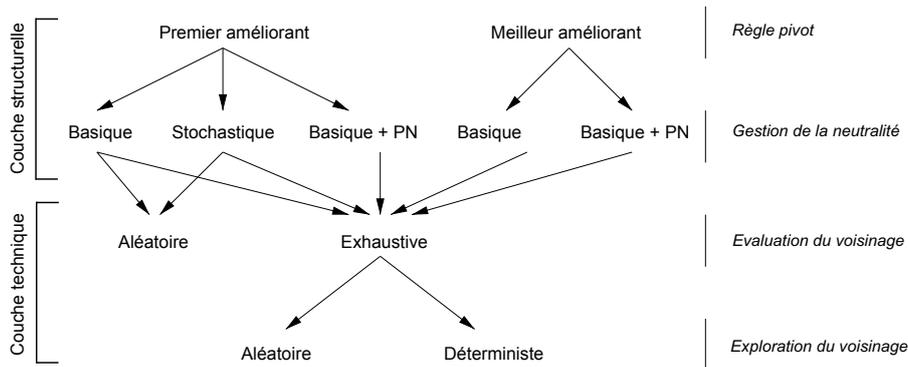


FIGURE 21 – Stratégies de hill-climbing.

Sans procéder à une analyse systématique de la littérature mais en parcourant les nombreux algorithmes de recherche locale qui sont régulièrement proposés, il nous apparaît que le plus souvent, ces derniers sont conçus autour de la stratégie du meilleur améliorant. C’est notamment la règle pivot utilisée dans la version classique de la recherche tabou,

une des métaheuristiques les plus utilisées. Lorsque les voisinages sont larges, que la fonction d'évaluation est couteuse et qu'une évaluation incrémentale des voisins est difficile voire impossible, la stratégie du premier améliorant reste utilisée, mais souvent sans autre justification que par commodité ou par rapidité d'exécution. De même, un climber stochastique est très rarement utilisé dans un contexte de premier améliorant — bien que la métaheuristique du recuit simulé accepte systématiquement les mouvements neutres —, comme si l'intuition de procéder autant que possible à de plus hauts pas permettait de monter plus haut. Le fait que, par expérience comme par évidence, nous pensons que c'est très souvent le contraire qui se vérifie, nous a amené vers cette étude empirique de comparaison et d'extension des stratégies d'escalade de paysages, détaillée dans les chapitres 7 et 8 :

- Le chapitre 7 compare dans un premier temps l'efficacité des stratégies du premier et du meilleur améliorant ; notre propos est ici de montrer que dans de nombreux cas, le premier améliorant est à privilégier. Nous lions également l'efficacité des règles pivot aux caractéristiques des paysages à escalader. L'étude est ensuite élargie à l'expérimentation de règles pivot alternatives, qui nous permet notamment d'indiquer que si la stratégie du premier améliorant est pertinente, ce n'est pas par introduction d'un mécanisme de sélection non déterministe, mais parce que les mouvements les moins améliorants permettent au climber d'atteindre de plus hauts optimums locaux en moyenne.
- Le chapitre 8 compare les différentes politiques de mouvements neutres afin d'évaluer l'intérêt du climber stochastique qui accepte les mouvements neutres tout au long de la recherche. Cela nous permet de mesurer l'apport de la neutralité dans les paysages de fitness, et nous discutons de pistes pour rendre les paysages plus simples à escalader, quitte à y incorporer de la neutralité artificiellement.

En espérant ne pas perdre le lecteur, nous proposons avant cette série d'études comparatives un travail plus fondamental sur l'atteignabilité des optimums globaux dans les paysages de fitness, par un climber de type premier améliorant — qui est la stratégie de climber strict offrant le plus de trajectoires possibles. Ce travail présenté au chapitre 6 vise à montrer que les hautes zones des paysages de fitness, considérées comme difficiles à atteindre, restent très largement atteignables en termes de possibilité par des stratégies de recherche purement intensificatrices. Ce résultat est l'une des motivations nous conduisant à étudier plus spécifiquement, lors des chapitres suivants, les stratégies de climber. Nous pensons en effet qu'une meilleure compréhension des stratégies de sélection des voisins — en premier lieu la règle pivot et la gestion de la neutralité — permettra d'améliorer sensiblement l'efficacité des algorithmes de recherche locale.

Chapitre 6

Atteignabilité des optimums globaux

Ce chapitre consiste à mesurer l'impact de la solution initiale sur les solutions potentiellement atteignables par une recherche locale purement intensificatrice. Via le modèle des paysages de fitness, nous évaluerons notamment la possibilité d'atteindre des points particuliers de l'espace — en particulier l'optimum global — au moyen de climbers.

6.1 Visualisation des paysages de fitness

Dans un souci de vulgarisation et de représentation mentale, nous avons parfois tendance à faire une analogie entre espaces de recherche combinatoires et univers. Pédagogiquement, pour rendre compte de la très grande taille de certains espaces de recherche combinatoires, il est d'usage de comparer le nombre de solutions envisageables à une instance de problème avec le nombre d'atomes dans l'univers. Bien souvent, les espaces de recherche combinatoires sont même bien plus larges que l'ensemble d'atomes de l'univers, dont le dénombrement est estimé à 10^{80} , soit environ 2^{266} . L'abstraction en paysages de recherche ou paysages de fitness entretient d'ailleurs particulièrement ce parallèle. Effectuer une recherche locale, ou escalader un paysage de fitness au moyen d'un climber, consiste à explorer les solutions (points, atomes) de l'espace de recherche (paysage, univers) de proche en proche jusqu'à satisfaire un critère d'arrêt, comme atteindre un optimum local. Le concept de voisinage se visualise alors comme une relation de proximité, où dans cette représentation les voisins d'un atome seraient les quelques centaines ou milliers d'atomes les plus proches. Un mouvement dans cet espace de recherche représenterait ainsi des pas de quelques microns au maximum. On imagine alors l'impossibilité pour un algorithme de recherche locale de traverser l'espace de recherche, donc les 30 années-lumière de l'univers, par ce moyen ; ce qui, même à raison d'un million de pas par seconde et en choisissant la trajectoire la plus directe, représenterait 3 millions d'années. La terminologie de recherche *locale* entretient d'ailleurs particulièrement cette analogie, notamment lorsque l'on aborde les algorithmes mémétiques, où les croisements cherchent à envisager des zones prometteuses de l'espace, tandis que le rôle des recherches locales, et à plus forte raison des climbers, est d'intensifier la recherche dans cette zone que l'on imagine de taille négligeable

en comparaison de la grandeur de l'espace. Cette correspondance imaginaire laisse ainsi à penser qu'une recherche locale ne permet pas de nous déplacer suffisamment loin dans un espace de recherche, mais aussi que le point de départ est extrêmement conditionnant dans les points visitables, et détermine alors fortement les optimums locaux trouvés — y compris en utilisant une stratégie de recherche non déterministe.

Une seconde représentation des espaces combinatoires, largement présente dans la littérature, consiste à les percevoir schématiquement par des surfaces bidimensionnelles (parfois même par de simples courbes), encore plus réductrices mais permettant de représenter des points de hauteurs différentes et ainsi de visualiser le concept d'optimum local ou de rugosité. Nous l'utilisons d'ailleurs dans le chapitre précédent (figure 18). La figure 22 illustre des exemples de surfaces mono et bidimensionnelles, avec des exemples de graphes de transition associés dans le cadre d'espaces de recherche discrets (suite de points, grille bidimensionnelle). Cette schématisation est généralement accompagnée du discours voulant qu'une stratégie purement intensificatrice (climber ou descente) ne peut atteindre que quelques optimums locaux proches, voire un seul, et que l'art de concevoir une stratégie de recherche efficace réside dans le mécanisme de diversification. Le passage à des graphes de transition plus denses implique des surfaces impossibles à représenter dans l'espace, et dont les propriétés structurelles sont très différentes. Par exemple, le simple ajout d'une transition dans les graphes simplistes de la figure 22 peut rendre caduc le propos prétendant que des points de départ particuliers conduiront inévitablement un climber vers un optimum local de mauvaise qualité. Comme les résultats exposés au long de ce chapitre le confirmeront, cette représentation générique en surface bidimensionnelle ne peut être utilisée comme une abstraction appropriée des paysages de fitness combinatoires, donc très fortement multidimensionnels, dans le but d'aborder des concepts algorithmiques originaux pour escalader les paysages ou sortir des optimums locaux plus efficacement.

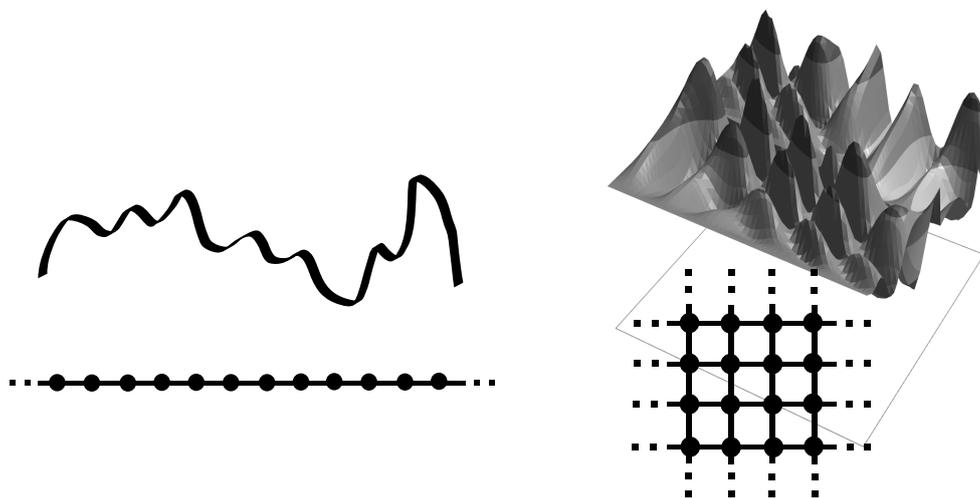


FIGURE 22 – Graphes de transition induits par les représentations schématiques classiques des paysages de fitness.

Contrairement à ce que peuvent laisser penser les représentations schématiques habituelles, les espaces de recherche combinatoires ont des diamètres relativement réduits. Par exemple, l'espace de recherche constitué des chaînes de N bits et muni de la relation de voisinage 1-flip — soit le graphe de transition $TG(\{0,1\}^N, 1\text{-flip})$ — est un hypercube de dimension N (voir figure 23.a pour $N = 9$). Lorsque $N = 266$, sa taille avoisine le nombre d'atomes dans l'univers, et chaque point ne possède que 266 voisins. Pourtant, son diamètre est 266 ; c'est-à-dire qu'au maximum 266 pas suffisent pour se déplacer d'un point x à un point y , quelconques. Plus généralement, la proportion de points situés à une distance d de tout point x est $\binom{N}{d}/2^N$. Ainsi, si $N = 266$, la distance entre deux points quelconques de l'espace — ou un point quelconque et un point fixé, comme un point de départ aléatoire et l'optimum global, même inconnu *a priori* — est comprise entre 112 et 154 dans 99.99% des cas, et entre 106 et 160 dans 99.999% des cas (voir figure 23.b). Ce rappel élémentaire nous confirme que dans les espaces de recherche combinatoires traditionnels, (presque) toutes les solutions sont (à peu près) à une même distance (raisonnable) ; une distance qui n'est pas d'un ordre de grandeur supérieur au nombre de pas habituellement réalisés par des algorithmes de recherche locale. Le caractère *local* de ces recherches ne constituerait donc pas un frein à la capacité exploratoire de ce type d'algorithmes.

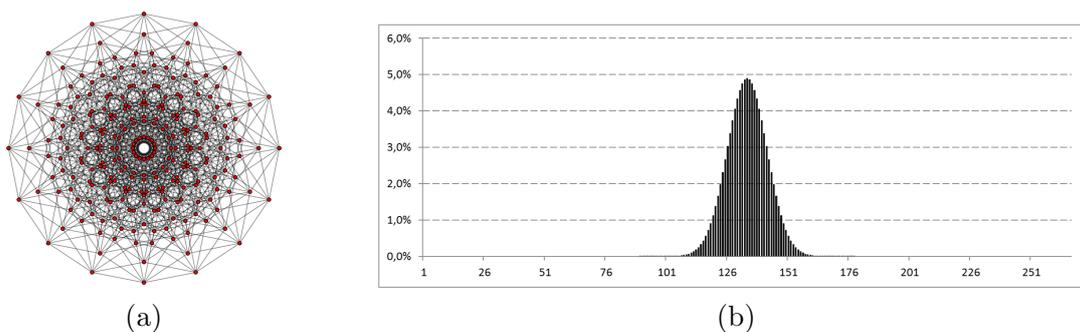


FIGURE 23 – (a) Hypercube de dimension 9, graphe de transition $TG(\{0,1\}^9, 1\text{-flip})$ possédant 512 sommets. (b) Distribution des distances entre sommets d'un hypercube de dimension 266, correspondant au graphe de transition $TG(\{0,1\}^{266}, 1\text{-flip})$ et possédant environ 10^{80} sommets.

Intéressons-nous à présent aux différents chemins empruntables par un algorithme de recherche locale au sein d'un paysage de fitness. Tout d'abord, une marche aléatoire, ou tout algorithme n'ayant pas de stricte restriction sur les mouvements²⁷, a la *possibilité* d'atteindre tout point particulier de l'espace de recherche, depuis tout point de départ, sous la seule condition que le graphe de transition est fortement connexe — condition toujours vérifiée en pratique. En particulier, $TG(\{0,1\}^N, 1\text{-flip})$ est fortement connexe et possède $N \times 2^N$ transitions. Nous avons introduit au chapitre précédent la notion de graphe de transition contraint, associé à un critère de mouvement. Par exemple, $TG_{>}((\mathcal{X}, \mathcal{N}, f))$ est le graphe $(\mathcal{X}, \{(x, y) \in \mathcal{A}_{(\mathcal{X}, \mathcal{N})}, f(y) > f(x)\})$. Il est intéressant de noter qu'un climber de type premier améliorant sur le paysage $\mathcal{P} = (\mathcal{X}, \mathcal{N}, f)$ équivaut à une marche aléatoire sur $TG_{>}(\mathcal{P})$. Dans le cas général, $TG_{>}(\mathcal{P})$ perd sa propriété de connexité forte. Néanmoins, en considérant un paysage non neutre dont le voisinage est symétrique (tel

²⁷Par exemple, un recuit simulé peut être considéré comme une marche aléatoire non uniforme.

que $y \in \mathcal{N}(x) \Rightarrow x \in \mathcal{N}(y)$, il existe exactement 50% de transitions améliorantes (soit $x \rightarrow y$, soit $y \rightarrow x$). En particulier, quelle que soit la fonction de fitness non neutre f définie, $TG_{>}(\{0, 1\}^N, 1\text{-flip}, f)$ conserve $N \times 2^{N-1}$ transitions, ce qui reste important et ne semble pas constituer le frein exploratoire que l'on associe aux recherches intensificatrices.

Nous cherchons dans ce chapitre à mesurer l'effet de cette restriction de moitié du nombre de transitions sur la *possibilité* d'atteindre un point particulier — à commencer par l'optimum global — depuis un point quelconque de l'espace de recherche. Pour cela, nous évaluons la proportion de points de l'espace de recherche à partir desquels l'optimum global peut être atteint par une recherche de type premier améliorant.

6.2 Bassin d'attraction et couverture

Étant donné un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$ et une stratégie d'escalade (premier ou meilleur améliorant), le *bassin d'attraction* d'un optimum local²⁸ x_{opt} est l'ensemble $B_{x_{\text{opt}}} = \{x \in \mathcal{X}, p_{x_{\text{opt}}}(x) > 0\}$, où $p_{x_{\text{opt}}}(x)$ représente la probabilité d'atteindre x_{opt} depuis le point de départ x [Garnier et Kallel, 2001]. Originellement, les bassins d'attraction se définissaient comme une partition de l'espace de recherche car regroupant les points de départ amenant aux mêmes optimums locaux par application de l'algorithme du gradient, déterministe (*Steepest Ascent Algorithm*). Ochoa *et al.* [2010] ont étendu la définition pour l'associer au hill-climbing de type premier améliorant. Les auteurs y définissent la taille d'un bassin d'attraction $B_{x_{\text{opt}}}$ comme la somme des probabilités $\sum_{x \in \mathcal{X}} p_{x_{\text{opt}}}(x)$, et non comme sa cardinalité, afin de représenter la probabilité pour un optimum local donné d'être atteint par un climber débutant sa recherche depuis une solution aléatoire (la probabilité globale est plus précisément $\#B_{x_{\text{opt}}}/\#\mathcal{X}$). Dans cette section, notre étude vise au contraire à étudier la cardinalité des bassins d'attraction afin de déterminer la *possibilité* d'atteindre un optimum local donné.

Nous proposons ici une formalisation alternative des bassins d'attraction, qui nous permettra davantage de généralité dans les définitions. De même, au niveau du vocabulaire, il nous paraît plus approprié dans notre étude de parler de *couverture* que de *bassin d'attraction*, en qualifiant l'ensemble de l'espace à partir duquel un point particulier peut être atteint relativement à une stratégie de mouvement.

Définition 12 (Couverture et taux de couverture) *On définit la couverture Γ d'un point $x \in \mathcal{X}$ sur un ensemble de points $\mathcal{S} \subseteq \mathcal{X}$ et relativement à un graphe de transition [contraint] G comme l'ensemble des points de \mathcal{S} pouvant atteindre x dans G (points auxquels x est relié par un chemin) :*

$$\Gamma(x, \mathcal{S}, G) = \{x' \in \mathcal{S}, d_G(x', x) \neq \infty\} \quad (31)$$

Le taux de couverture γ est la proportion de points de \mathcal{S} à partir desquels x est atteignable :

$$\gamma(x, \mathcal{S}, G) = \frac{\#\Gamma(x, \mathcal{S}, G)}{\#\mathcal{S}} \quad (32)$$

²⁸La définition des bassins d'attraction se réduit aux optimums locaux mais pourrait naturellement être généralisée à tout point de l'espace de recherche.

Notons que si $G = (\mathcal{X}, \mathcal{A})$ est fortement connexe, alors $\forall x \in \mathcal{X}, \forall \mathcal{S} \subseteq \mathcal{X}, \gamma(x, \mathcal{S}, G) = 1$. Enfin, même si nous ne l'utiliserons pas dans la suite de ce chapitre, nous définissons similairement la couverture restreinte Γ_k :

Définition 13 (Couverture restreinte et taux de couverture restreinte) *La couverture restreinte est l'ensemble des points à partir desquels x est atteignable en moins de k transitions :*

$$\Gamma_k(x, \mathcal{S}, G) = \{x' \in \mathcal{S}, d_G(x', x) \leq k\} \quad (33)$$

Le taux de couverture restreinte associé est :

$$\gamma_k(x, \mathcal{S}, G) = \frac{\#\Gamma_k(x, \mathcal{S}, G)}{\#\mathcal{S}} \quad (34)$$

Soit $\mathcal{P} = (\mathcal{X}, \mathcal{N}, f)$ un paysage de fitness. Le graphe de transition contraint $TG_{>}(\mathcal{P})$ contient tous les chemins possibles empruntables par un climber de type premier améliorant sur le paysage de fitness associé. Soit x^* un point cible de \mathcal{P} (par exemple l'optimum global). On considère que la *possibilité* d'atteindre x^* par un tel climber sera égale à la proportion de points de départ à partir desquels il est possible d'atteindre x^* , c'est-à-dire $\gamma(x^*, \mathcal{X}, TG_{>}(\mathcal{P}))$. La probabilité associée sera notée $\phi(x^*, \mathcal{X}, TG_{>}(\mathcal{P}))$ et correspond à la taille des bassins d'attraction telle que définie par Ochoa *et al.* [2010].

6.3 Atteignabilité des optimums globaux par un climber

Dans ce chapitre, toutes les études expérimentales seront réalisées sur des paysages NK non neutres. Cette première étude consiste à calculer l'atteignabilité (taux de couverture) des optimums globaux de différents paysages NK par un algorithme de hill-climbing, en fonction du point de départ (solution initiale). Autrement dit, la proportion de points d'un paysage à partir desquels il existe un chemin de recherche strict menant vers l'optimum global :

$$\frac{\#\{x_0 \in \mathcal{X} \mid \exists(x_1, \dots, x_k), x_k = x^*, \text{ avec } x_i \in \mathcal{N}(x_{i-1}) \text{ et } f(x_i) > f(x_{i-1})\}}{\#\mathcal{X}} \quad (35)$$

Nous notons ici $\mathcal{P}_{(N,K)}^i$ l'instance i de paysage NK générée au moyen des paramètres N et K . Nous utilisons dans cette étude 60 paysages NK ($N \in \{16, 24\}, K \in \{2, 4, 8\}, i \in \{1, \dots, 10\}$), dont on calcule de manière exacte les taux de couverture des optimums globaux. Une première étape consiste alors à déterminer exactement chaque optimum global $x_{(N,K,i)}^* = \operatorname{argmax}_{x \in \mathcal{X}} f_{(N,K)}^i(x)$, avec $\mathcal{X} = \{0, 1\}^N$ et $f_{(N,K)}^i$ la fonction de fitness du paysage $\mathcal{P}_{(N,K)}^i$. On calcule ensuite $\gamma(x_{(N,K,i)}^*, \mathcal{X}, TG_{>}(\mathcal{P}_{(N,K)}^i))$, qui représente la proportion du paysage pouvant mener vers l'optimum global au moyen d'un climber strict.

En pratique, on calcule de manière exacte le taux de couverture de l'optimum global par exploration arborescente du graphe de transition contraint complémentaire $TG_{<}(\mathcal{P}_{(N,K)}^i)$ depuis $x_{(N,K,i)}^*$, autrement dit en déterminant tous les chemins de descente possibles avec l'optimum global comme point de départ. Une approche naïve consisterait à générer récursivement tous les voisins détériorants jusqu'à atteindre un minimum local, et dénombrer le nombre de points distincts ainsi considérés. Cependant, une telle approche est

impraticable, puisque la plupart des points appartiennent à de nombreux chemins de recherche. L'utilisation d'une table de hachage avec la fonction de fitness comme fonction de hachage nous permet de nous assurer que chaque point n'est considéré qu'une seule fois. Combiné à un traitement des points par fitness décroissante, cette technique permet de libérer suffisamment de mémoire durant la recherche en supprimant les points qui ne pourront plus apparaître par la suite (voir l'algorithme 3).

Algorithme 3 : Calcul du taux de couverture d'un point de référence du paysage de fitness \mathcal{P} , relativement à $TG_{>}(\mathcal{P})$.

Données : un paysage de fitness $\mathcal{P} = (\mathcal{X}, \mathcal{N}, f)$ et un point de référence x^* .
Résultat : le taux de couverture $\gamma(x^*, \mathcal{X}, TG_{>}(\mathcal{P}))$.

- 1 $C \leftarrow 1$; // C comptabilise le nombre de points de $\Gamma(x^*, \mathcal{X}, TG_{>}(\mathcal{P}))$
- 2 $F^* \leftarrow \lfloor f(x^*) \cdot T \rfloor$; // T est la taille de la table de hachage H initialement vide ($\forall F \in \{0, \dots, T\}, H(F) \leftarrow \emptyset$)
- 3 $H[F^*] \leftarrow H[F^*] \cup x^*$;
- 4 **pour** $F \leftarrow F^*$ à 0 **faire**
- 5 **pour chaque** $x \in H[F]$ **faire**
- 6 $N \leftarrow \{x' \in \mathcal{N}(x), f(x') < f(x)\}$;
- 7 **pour chaque** $x' \in N$ **faire**
- 8 $F' \leftarrow \lfloor f(x') \cdot T \rfloor$;
- 9 **si** $x' \notin H[F']$ **alors**
- 10 $H[F'] \leftarrow H[F'] \cup x'$;
- 11 $C \leftarrow C + 1$
- 12 **fin**
- 13 **fin**
- 14 $H[F] \leftarrow \emptyset$
- 15 **fin**
- 16 **fin**
- 17 **retourner** $C/2^N$

La partie supérieure de la table 11 présente les taux de couverture des optimums globaux des 60 paysages NK considérés. Nous remarquons que le taux de couverture moyen constaté est supérieur à 70% pour chaque paramétrage (N, K) , supérieur à 50% pour chaque paysage étudié, et très souvent supérieur à 90% sur certains paramétrages de paysages ($K = 4$). Cela signifie que l'optimum global est atteignable par une recherche locale stricte à partir d'une majorité de points de départ; autrement dit, dans un contexte de résolution de problème, qu'il existe pour une majorité de solutions initiales une séquence de mutations améliorantes permettant d'atteindre la solution optimale. Ces taux de couverture ne décroissent pas lorsque la taille des paysages augmente — on observe même le contraire —, et nous pouvons supposer que les taux de couverture des optimums globaux restent élevés également sur des instances plus larges.

Sur la partie inférieure de la table 11, les lignes OL2 à OL5 indiquent la moyenne des taux de couverture des quatre autres meilleurs optimums locaux de chaque paysage (optimums triés par ordre décroissant, avec OL1 qui symboliserait les optimums globaux, dont la moyenne des taux de couverture est indiquée au-dessus). Globalement, on peut observer que les taux de couverture ont tendance à décroître légèrement tout en restant importants. Puisque le nombre d'optimums locaux est bien plus faible sur les instances lisses

Instance	$N = 16$			$N = 24$		
	$K = 2$	$K = 4$	$K = 8$	$K = 2$	$K = 4$	$K = 8$
1	94.4%	91.5%	68.4%	63.2%	96.0%	88.9%
2	55.7%	86.7%	74.8%	91.9%	96.1%	88.9%
3	91.0%	86.9%	74.3%	66.2%	95.6%	91.6%
4	67.8%	93.0%	67.0%	71.6%	95.8%	88.7%
5	54.7%	88.8%	69.4%	92.4%	95.0%	90.6%
6	75.0%	87.8%	68.5%	72.4%	97.7%	86.6%
7	68.9%	87.8%	69.4%	63.6%	94.3%	87.6%
8	74.0%	87.2%	66.8%	96.9%	96.6%	89.6%
9	96.8%	91.1%	71.3%	80.1%	93.0%	90.6%
10	78.7%	87.8%	71.9%	82.4%	96.1%	91.0%
Moy. (OG)	75.7%	88.9%	70.2%	78.1%	95.6%	89.4%
Global	Moyenne : 78.2% Ecart-type : 11.7%			Moyenne : 87.7% Ecart-type : 10.2%		
Moy. OL2	70.7%	87.2%	68.7%	76.2%	95.4%	90.0%
Moy. OL3	65.0%	87.1%	68.2%	69.6%	95.2%	88.9%
Moy. OL4	71.3%	86.6%	67.2%	72.0%	95.7%	90.5%
Moy. OL5	61.6%	85.8%	68.6%	73.1%	94.0%	90.2%

TABLE 11 – Taux de couverture des optimums globaux (OG) de 60 paysages NK aléatoires ($N \in \{16, 24\}$ and $K \in \{2, 4, 8\}$), exprimés en pourcentages. Sur la partie inférieure est reportée la moyenne des taux de couverture des quatre autres plus hauts optimums locaux (OL2 à OL5).

($K = 2$), leurs fitness respectives diminuent plus significativement, et l’on imagine ainsi que les taux de couverture des points d’un paysage de fitness sont principalement corrélés à leurs fitness ; cet aspect pourrait être examiné au moyen d’études supplémentaires.

Enfin, nous avons observé la distribution des points du paysage selon leur fitness, considérant s’ils appartenaient ou non à la couverture de l’optimum global. Ceci permet de visualiser la corrélation existant entre taux de couverture de l’optimum global et intervalles de fitness des ensembles de points considérés, comme montré sur la figure 24. La ligne continue y indique que l’optimum global est presque toujours atteignable depuis le point le plus bas de l’espace (minimum global), ce qui est relativement intuitif dans le cas général. Au contraire, il est moins souvent possible d’atteindre l’optimum global à partir de points de haute fitness. Insistons sur le fait qu’une large couverture ne signifie pas que l’optimum global est facile à atteindre (voir section suivante), mais réfère à la possibilité de l’atteindre.

6.4 Accessibilité des optimums

La section précédente estimait l’atteignabilité de points particuliers d’un paysage, notamment de l’optimum global, en déterminant depuis quelle proportion de l’espace de recherche il existait au moins un chemin de recherche strict menant vers ces points. Les résultats tendent à montrer que l’optimum global est atteignable avec un climber basique depuis une majorité de points de départ ; ce qui n’implique naturellement pas une probabilité significative de l’atteindre. Cette section donne quelques éléments d’informations

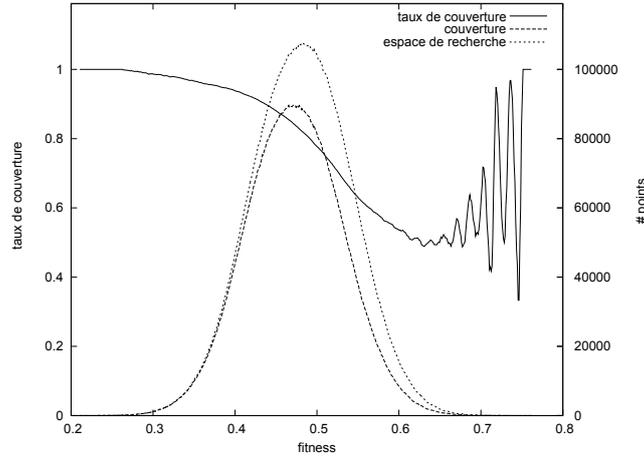


FIGURE 24 – Distribution de la couverture de l’optimum global d’un paysage NK ($N = 24$, $K = 2$, instance #9, taux de couverture global : 80.1%) en fonction des valeurs de fitness. Pour chaque intervalle de fitness (précision 10^{-3}), il est indiqué le nombre de points appartenant à la couverture de l’optimum global, et le nombre total de points de l’espace de recherche.

supplémentaires en considérant cette accessibilité²⁹ des points des paysages de fitness en fonction, notamment, de la hauteur des points et des propriétés des paysages.

La probabilité d’atteindre un point particulier du paysage — par exemple l’optimum global — avec un climber de type premier améliorant ne peut se calculer en testant exhaustivement tous les chemins possibles depuis tous les points de départ possibles. Nous proposons ici d’estimer cette probabilité à partir d’un échantillon de points de départ. Pour chacun d’entre eux, la probabilité d’atteindre la cible peut être calculée de manière exacte par l’algorithme 4 (non polynomial).

Contrairement à l’algorithme de calcul des taux de couverture, les chemins améliorants menant du point de départ x_0 au point cible x^* sont ici évalués à partir de x_0 ; en effet, le calcul de probabilité nécessite que tous les chemins de recherche possibles soient pris en compte. Dans un contexte de stratégie de type premier améliorant — les voisins étant considérés dans un ordre aléatoire, voir chapitre 5, section 5.6 —, nous considérons que tous les voisins améliorants ont la même probabilité d’être sélectionnés. Ainsi, si un point x a une probabilité p d’être atteint et a k voisins améliorants, alors la probabilité d’emprunter chacune de ces transitions est p/k . Si un point peut être atteint depuis plusieurs chemins, les différentes probabilités entrantes sont additionnées. Si x^* n’est pas atteignable depuis x_0 , la probabilité associée sera naturellement nulle.

La table 12 indique les probabilités moyennes d’atteindre les plus hauts optimums locaux de paysages NK, avec $N = 16$ et en fonction de différents niveaux de rugosité. Les moyennes et écarts-types reportés ont été obtenus à partir des valeurs exactes calculées sur 10 paysages différents. On peut y vérifier que les optimums globaux (OG) sont relativement peu accessibles lorsque $K = 4$ et $K = 8$ (respectivement 4.6% et 0.9%, ce

²⁹L’*atteignabilité* se réfère à ce qu’il est possible d’atteindre, tandis que l’*accessibilité* rapporte un potentiel d’accès.

Algorithme 4 : Calcul de la probabilité d’atteindre un point cible à partir d’un point de départ au moyen d’un climber basique de type premier améliorant.

Données : un paysage de fitness $\mathcal{P} = (\mathcal{X}, \mathcal{N}, f)$, un point cible x^* et un point de départ x_0 .

Résultat : la probabilité d’atteindre x^* depuis x_0 .

```

1  $F_0 \leftarrow \lfloor f(x_0).T \rfloor$ ;
2  $F^* \leftarrow \lfloor f(x^*).T \rfloor$ ; //  $T$  est la taille de la table de hachage  $H$  initialement vide ( $\forall F \in \{0, \dots, T\}, H(F) \leftarrow \emptyset$ )
3  $H[F_0] \leftarrow H[F_0] \cup x_0$ ;
4  $P[x_0] \leftarrow 1$ ; //  $P$  mémorise les probabilités d’atteindre les points rencontrés depuis  $x_0$ 
5 pour  $F \leftarrow F_0$  à  $F^*$  faire
6   pour chaque  $x \in H[F]$  faire
7      $N \leftarrow \{x' \in \mathcal{N}(x), f(x') > f(x)\}$ ;
8     pour chaque  $x' \in N$  faire
9        $F' \leftarrow \lfloor f(x').T \rfloor$ ;
10      si  $x' \notin H[F']$  alors
11         $H[F'] \leftarrow H[F'] \cup x'$ ;
12         $P[x'] \leftarrow P[x] / \#N$ 
13      sinon  $P[x'] \leftarrow P[x'] + P[x] / \#N$ 
14    fin
15     $H[F] \leftarrow \emptyset$ 
16  fin
17 fin
18 retourner  $P[s^*]$ 

```

qui reste néanmoins supérieur aux taux moyennés sur l’ensemble des optimums locaux, respectivement de 0.8% et 0.1%). Comme attendu, les taux de couverture de ces optimums globaux n’impliquent pas des valeurs d’accessibilité importantes, même si pour les paysages plus lisses et dits faciles ($K = 2$), les optimums globaux restent relativement accessibles (plus de 10%). La probabilité d’atteindre l’optimum global décroît naturellement lorsque la rugosité du paysage, et donc le nombre d’optimums locaux, augmente. Cependant, nous rappelons que ce phénomène n’est pas présent en ce qui concerne le taux de couverture. Comme la probabilité d’atteindre l’optimum global par une technique simple et non déterministe est un indicateur numérique évident de la difficulté d’un problème, et que le niveau de rugosité affecte directement son accessibilité, cela confirme que la rugosité d’un paysage affecte directement sa difficulté.

Les colonnes relatives aux différents optimums locaux, comme la colonne 50%, apportent des précisions complémentaires. On remarque ainsi que quelle que soit la rugosité du paysage, la probabilité d’atteindre l’un des 50% des meilleurs optimums locaux est supérieure à 70%. Cela signifie qu’une stratégie de type premier améliorant tend globalement à converger vers les plus hauts optimums. Des conclusions similaires ont été obtenues par Verel *et al.* [2008] via la stratégie du meilleur améliorant. Les auteurs ont observé que les plus hauts optimums locaux bénéficiaient en général de bassins d’attraction plus larges.

N	K	OG	10%	30%	50%	Top 5	#OL	1/#OL	γ
16	2	14.8%	21.8%	54.0%	75.9%	65.7%	35.7	2.8%	75.7%
		$\pm 13.9\%$	$\pm 11.4\%$	$\pm 12.4\%$	$\pm 9.0\%$	$\pm 22.6\%$	± 21.9	$\pm 2.0\%$	$\pm 14.9\%$
	4	4.6%	32.2%	65.3%	82.7%	18.4%	120.5	0.8%	88.9%
		$\pm 1.8\%$	$\pm 5.0\%$	$\pm 4.8\%$	$\pm 2.2\%$	$\pm 4.8\%$	± 18.0	$\pm 0.1\%$	$\pm 2.2\%$
	8	0.9%	29.6%	59.7%	78.1%	3.2%	755.4	0.1%	70.2%
		$\pm 0.7\%$	$\pm 2.5\%$	$\pm 1.6\%$	$\pm 1.0\%$	$\pm 1.2\%$	± 21.4	$\pm 0.0\%$	$\pm 2.8\%$

TABLE 12 – Probabilité d’atteindre différents optimums locaux sur 10 paysages NK aléatoires (avec $N = 16$ et $K \in \{2, 4, 6\}$). Les probabilités sont estimées par échantillonnage d’un ensemble de 100 points de départ aléatoires. La colonne OG correspond à l’accessibilité de l’optimum global, tandis que les colonnes 10%, 30%, 50% font respectivement référence à l’accessibilité des optimums appartenant aux 10%, 30% et 50% des plus hauts optimums locaux. La colonne *Top 5* indique l’accessibilité de l’un des cinq plus hauts optimums locaux. #OL correspond au nombre d’optimums locaux du paysage, dont l’inverse exprimé en pourcentage aide à apprécier la mesure d’accessibilité de l’optimum global et des cinq plus hauts optimums. Enfin, nous reportons les taux de couverture moyens γ des optimums globaux sur chacun de ces ensembles de paysages.

6.5 Influence de la solution initiale sur une recherche locale

Dans une troisième étude, nous nous sommes intéressés à évaluer l’atteignabilité des optimums globaux par un climber en fonction de la distance à l’optimum du point de départ. Nous distinguons ici les taux de couverture selon la distance de voisinage entre les points de départ et le point cible considéré (l’optimum global). L’existence d’une corrélation entre taux de couverture et distance de voisinage semble certes naturelle, mais nous cherchons à déterminer ici si les taux de couverture sont toujours significatifs depuis des zones plus lointaines du paysage, et à quel point ils peuvent être élevés à proximité des optimums globaux.

Soit $\mathcal{P} = (\mathcal{X}, \mathcal{N}, f)$ un paysage de fitness et $TG(\mathcal{P})$ le graphe de transition associé. Soit $\mathcal{S}_\delta^x \subseteq \mathcal{X}$ l’ensemble des points distants de δ dans le graphe de transition (donc selon la distance de voisinage) :

$$\mathcal{S}_\delta^x = \{x' \in \mathcal{X}, d_{TG(\mathcal{P})}(x, x') = \delta\} \quad (36)$$

Pour des faibles valeurs de δ , les valeurs de $\gamma(x_{(N,K,i)}^*, \mathcal{S}_\delta^{x^*}, TG_{>}(\mathcal{P}_{(N,K)}^i))$ associées représenteront les *couvertures locales* de l’optimum global x^* . Nous proposons ici d’étudier l’évolution des taux de couverture des optimums globaux en fonction de δ . Notons que l’évolution des probabilités $\phi(x_{(N,K,i)}^*, \mathcal{S}_\delta^{x^*}, TG_{>}(\mathcal{P}_{(N,K)}^i))$ en fonction de δ — qui elle ne sera pas étudiée — illustre l’attractivité de x^* .

Dans le cadre des paysages NK, les distances de voisinage entre points sont des distances de Hamming, variant entre 1 et N . Comme mentionné en début de chapitre, depuis tout point particulier x^* du paysage, le nombre de points étant à une distance de Hamming de δ , soit $\#\mathcal{S}_\delta^{x^*}$, est égal à $\binom{N}{\delta}$. Considérant l’optimum global comme point-cible x^* , les valeurs $\gamma(x_{(N,K,i)}^*, \mathcal{S}_\delta^{x^*}, TG_{>}(\mathcal{P}_{(N,K)}^i))$ peuvent s’obtenir à partir de l’algorithme 3 adapté en calculant simultanément N tailles de couverture plutôt qu’une seule, globale ; ce qui n’augmente pas l’effort calculatoire.

La figure 25 offre une vue plus précise de la couverture de l’optimum global (moyenne sur 10 paysages aléatoires, avec $N = 16$ et $K = 4$). Nous illustrons ici la quantité et la proportion de points appartenant à la couverture de x^* en fonction de leur distance de

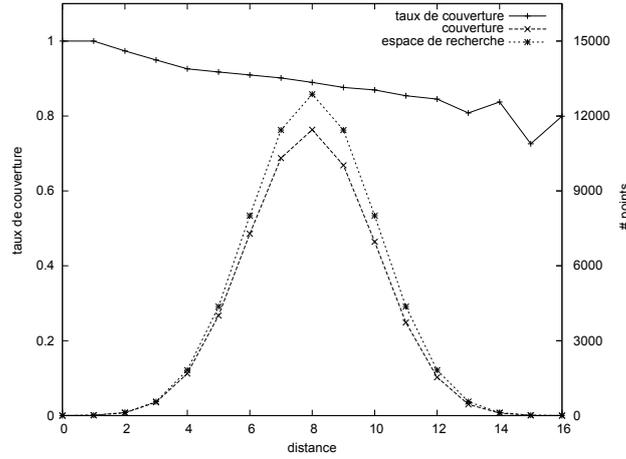


FIGURE 25 – Répartition des points et couverture moyenne des optimums globaux de paysages NK ($N = 16$, $K = 4$, 10 instances, taux de couverture global : 88.9%) selon les distances à l’optimum. Pour chaque distance, on indique le nombre moyen de points appartenant à la couverture de chaque optimum global, le nombre total de points de l’espace de recherche situés à cette distance, ainsi que le rapport des deux correspondant au taux de couverture.

voisinage à l’optimum (ici distance de Hamming). Bien que les taux de couverture aient tendance à décroître linéairement lorsque les points de départ s’éloignent, nous remarquons qu’ils restent relativement élevés même sur des zones éloignées du paysage. La table 13 regroupe les taux de couverture moyens obtenus sur tous les paysages NK utilisés dans le cadre de cette étude ($N = 16$) en fonction de K .

$K \setminus \delta$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	100	98	95	91	87	83	79	76	72	68	65	62	58	56	54	50
4	100	97	95	93	92	91	90	89	88	87	85	81	84	73	80	
8	100	95	88	81	77	74	71	70	68	67	65	63	63	57	56	40

TABLE 13 – Taux de couverture moyens constatés, en fonction de K et δ , avec $N = 16$ (valeurs arrondies exprimées en pourcentages).

Sur des paysages plus larges, il est possible de calculer la couverture *aux alentours* de l’optimum global — ou d’un optimum local de bonne qualité, optimum approché du paysage. Cela permet de nous indiquer à quel point être proche de l’optimum global, ou d’un bon optimum local, rend celui-ci atteignable par une recherche locale simple. Pour cela, on commence par calculer un optimum cible $x^\#$ par une recherche locale itérée, puis les valeurs de $\gamma(x^\#_{(N,K,i)}, \mathcal{S}_\delta^{x^\#}, TG > (\mathcal{P}^i_{(N,K)}))$, pour $\delta \leq 5$ et moyennées sur i . La table 14 reporte les taux de couverture correspondants, sur des paysages de paramètres $N = 32$ et $K \in \{2, 4, 8\}$. Ceux-ci sont très élevés, indiquant tout d’abord qu’une recherche arborescente énumérant tous les chemins de recherche améliorants courts — rares sur des zones plus hautes du paysage — permet presque à coup sûr de déterminer l’optimum si celui-ci se trouve (au plus) à la distance de recherche envisagée. En outre, si l’on prend comme hypothèse que les taux de couverture en fonction de la distance à l’optimum suivent un schéma de décroissance linéaire (voir figure 25 et tableau 13), alors les taux

calculés sur les premières valeurs laissent supposer des taux de couverture de l’optimum global — et plus généralement des points les plus hauts du paysage — du même ordre de grandeur. Autrement dit, nous conjecturons que même sur des paysages plus larges, l’optimum global est atteignable depuis une proportion non négligeable, voire une majorité, de points de départ. La principale perspective de ces travaux consiste à vérifier cette hypothèse sur des paysages de fitness et des problèmes combinatoires variés. Si celle-ci se vérifie, nous pourrions en déduire que des techniques d’intensification pure ont la *possibilité* de résoudre un problème combinatoire, la plupart du temps, depuis n’importe quelle solution initiale. Il serait alors intéressant de déterminer des techniques d’escalade plus efficaces (voir les chapitres suivants), plutôt que chercher exclusivement à développer des outils de diversification sophistiqués.

$K \setminus \delta$	1	2	3	4	5
2	100%	99.6%	98.2%	96.6%	94.4%
4	100%	99.5%	98.2%	97.6%	97.6%
8	100%	99.7%	98.7%	98.2%	98.1%

TABLE 14 – Taux de couverture moyens constatés, en fonction de K et δ , avec $N = 32$.

6.6 Atteignabilité de l’optimum global par une recherche locale courte

Dans une étude supplémentaire, nous avons comparé la distance de voisinage (Hamming) entre les points de départ et l’optimum global, avec la longueur du plus court chemin de recherche (strictement améliorant) reliant ces points. Une forte correspondance entre ces deux mesures pourrait permettre de réduire l’effort calculatoire pour résoudre des instances de problèmes combinatoires en considérant uniquement des chemins de résolution directs ou quasi-directs, c’est-à-dire ici sans flipper le même bit plusieurs fois.

Nous définissons la *couverture d’erreur e* d’un point x par un ensemble de points $\mathcal{S} \subseteq \mathcal{X}$, relativement à un graphe de transition (contraint) G et d’un graphe de transition de référence G_0 , par :

$$\Gamma_e^\sim(x, \mathcal{S}, G, G_0) = \{x' \in \mathcal{S}, d_G(x', x) - d_{G_0}(x', x) = e\}$$

Le taux de couverture d’erreur e associé, noté $\gamma_e^\sim(x, \mathcal{S}, G, G_0)$, vaut $|\Gamma_e^\sim(x, \mathcal{S}, G, G_0)| / \#\mathcal{S}$.

Étant donné un paysage $\mathcal{P} = (\mathcal{X}, \mathcal{N}, f)$ dont x^* est l’optimum global, $\Gamma_0^\sim(x^*, \mathcal{X}, TG_{>}(\mathcal{P}), TG(\mathcal{P}))$ est la couverture d’erreur 0 de x^* , c’est-à-dire l’ensemble de points à partir desquels un climber a la possibilité d’atteindre l’optimum global en un nombre de mouvements égal à sa distance de voisinage. À titre d’exemple, pour un paysage One-Max $\mathcal{P} = (\{0, 1\}^N, \mathcal{N}_{\text{flip}}, \text{One-Max})$, on a $\gamma_e^\sim(1^N, \{0, 1\}^N, TG_{>}(\mathcal{P}), TG(\mathcal{P})) = 1$.

Nous pouvons également extraire les propriétés suivantes, triviales mais utiles pour la suite de notre étude :

1. $\sum_{e=0}^\infty \gamma_e^\sim(x, \mathcal{S}, G, G_0) = \gamma(x, \mathcal{S}, G)$
2. $\gamma_e^\sim(x, \Gamma(x, \mathcal{S}, G), G_0) = \frac{\gamma_e^\sim(x, \mathcal{S}, G, G_0)}{\gamma(x, \mathcal{S}, G)}$

Le ratio γ_e^\sim/γ de la seconde propriété représente le taux de couverture d’erreur e d’un point x parmi sa couverture globale. Cette quantité détermine, parmi les points pouvant conduire à x , la proportion de ceux pouvant y accéder en $d_{G_0}(x', x) + e$ mouvements ; e étant le nombre de mouvements supplémentaires au plus court chemin empruntable par une marche aléatoire dans G_0 .

Notre étude expérimentale a consisté à mesurer les taux de couverture de faible erreur afin de déterminer si, depuis une partie non négligeable du paysage, il existait une recherche locale courte et strictement améliorante menant vers l’optimum global. Pour cela, nous avons repris le cadre expérimental utilisé dans la section 6.3 pour le calcul des taux de couverture globaux. Les mêmes 60 paysages NK ont été utilisés, notés $\mathcal{P}_{(N,K)}^i$ avec $N \in \{16, 24\}$, $K \in \{2, 4, 8\}$ et $i \in \{1, \dots, 10\}$. Nous avons alors calculé de manière exacte les valeurs $\gamma_e^\sim(x_{(N,K,i)}^*, \{0, 1\}^N, TG_{>}(\mathcal{P}_{(N,K)}^i), TG(\mathcal{P}_{(N,K)}^i))$, pour tout e pair³⁰ compris entre 0 et $N/2$. Afin de mieux illustrer la répartition des points de la couverture en termes d’erreur, et de déterminer si lorsque l’optimum est atteignable depuis un point de départ, alors il l’est avec un minimum de mouvements améliorants, nous nous intéressons en priorité aux ratios γ_e^\sim/γ .

N	K	γ_e^\sim			γ	γ_e^\sim/γ			DV/DR
		$e = 0$	$e = 2$	$e = 4$		$e = 0$	$e = 2$	$e = 4$	
16	2	69.4%	4.8% (74.2%)	1.1% (75.2%)	75.7%	91.7%	6.3% (98.0%)	1.4% (99.4%)	0.934
	4	80.0%	6.6% (86.6%)	1.6% (88.2%)	88.9%	90.0%	7.4% (97.4%)	1.8% (99.2%)	0.909
	8	57.6%	9.0% (66.5%)	1.8% (69.1%)	70.2%	82.0%	12.8% (94.8%)	3.6% (98.4%)	0.853
24	2	68.9%	6.0% (74.8%)	1.9% (76.7%)	78.1%	88.2%	7.6% (95.8%)	2.4% (98.2%)	0.904
	4	87.6%	6.0% (93.6%)	1.3% (94.9%)	95.6%	91.6%	6.3% (97.8%)	1.4% (99.2%)	0.936
	8	78.4%	7.3% (85.7%)	2.2% (88.0%)	89.4%	87.7%	8.2% (96.0%)	2.5% (98.4%)	0.853
Moyenne		73.6%	6.6% (80.2%)	1.8% (82.0%)	83.0%	88.5%	8.1% (96.6%)	2.2% (98.8%)	0.898

TABLE 15 – Taux de couverture d’erreurs 0, 2 et 4 moyens (γ_e^\sim) et ratios γ_e^\sim/γ des optimums globaux de paysages NK. Les taux cumulés sont indiqués entre parenthèses. La dernière colonne représente le rapport moyen entre distance de voisinage et distance de recherche.

La table 15 reporte les moyennes (sur 10 paysages) des taux de couverture à faible taux d’erreur ($e \in \{0, 2, 4\}$), pour chaque couple (N, K) . Les valeurs cumulées représentent les taux de couverture d’erreur au plus e , c’est-à-dire la proportion de l’espace de recherche telle qu’il existe un chemin de recherche ascendant ne dépassant pas e mouvements supplémentaires par rapport à la distance de voisinage. Nous pouvons constater que globalement, depuis près de 75% des points (solutions) aléatoires, il existe un chemin de recherche strict menant à l’optimum de même longueur que la distance de Hamming à l’optimum ; donc qu’il est possible, même sur des paysages rugueux, d’atteindre l’optimum en flipant une série de bits améliorant systématiquement la fitness, tout en n’opérant pas deux fois sur le même bit. Cela contredit l’intuition qu’une forte épistasie nécessite

³⁰Sur des espaces de recherche $\{0, 1\}^N$, $\gamma_e^\sim = 0$ pour tout e impair.

impérativement l'emploi de mouvements détériorants ou redondants pour atteindre de bons optimums locaux, et tout particulièrement l'optimum global. En revanche, la présence d'épistasie fait que l'ordre dans lequel les mutations améliorantes sont effectuées peut être déterminant.

En analysant les ratios γ_0/γ , on remarque que lorsqu'il existe un chemin ascendant vers l'optimum global, alors dans près de 90% des cas il existe un chemin minimal. Les moyennes des taux et ratios cumulés des couvertures d'erreur $e = 4$ au maximum, d'environ 82% et 99% respectivement, montrent que depuis une majorité de points de l'espace de recherche, il existe une séquence d'intensification courte — pas plus de $e/2 = 2$ bits reviennent à leur valeur d'origine — menant à l'optimum. En particulier, si un chemin existe, il est presque inutile d'envisager des chemins de recherche plus longs pour le déterminer. Ceci est confirmé par le rapport entre distance de voisinage et distance de recherche proche de 1 et fluctuant peu en fonction de la taille et de la rugosité des paysages. Ces observations indiquent que se focaliser uniquement sur les recherches locales qui évitent ou limitent l'application d'un mouvement plusieurs fois au cours de la recherche ne réduit pas significativement la possibilité d'atteindre l'optimum global. Ce résultat permet d'envisager des heuristiques de recherche arborescentes menant vers les meilleurs optimums.

Conclusion

En utilisant comme cadre expérimental les paysages NK, ce chapitre rend état de premiers résultats visant à améliorer la compréhension des méthodes de recherches locales. Nous nous sommes ici focalisés sur l'atteignabilité des optimums globaux par hill-climbing, c'est-à-dire en utilisant uniquement un mécanisme d'intensification. Les résultats tendent à montrer que les bonnes solutions (les plus hauts points des paysages), et particulièrement l'optimum global, sont atteignables depuis un nombre significatif de points de départ ; même, d'après nos expérimentations, depuis une majorité d'entre eux. Ce premier constat minimise l'impact du choix du point de départ et des mécanismes perturbatifs dans les algorithmes de recherche locale. Une seconde observation est que même sur les paysages plus rugueux, possédant un grand nombre d'interactions épistatiques, il est possible d'atteindre l'optimum global via des chemins de recherche très courts — que l'on peut approximativement estimer à la distance moyenne entre deux points dans l'espace de recherche, ou très légèrement supérieure —, sans nécessité d'appliquer des mouvements détériorants.

Cette étude nous mène à penser qu'il existe des possibilités d'améliorer l'efficacité des recherches locales, et des climbers en particulier, en définissant par exemple des règles pivot alternatives permettant d'accroître la probabilité d'atteindre les meilleurs optimums locaux et ainsi les fitness moyennes atteintes. Dans le chapitre suivant, la section 7.2.1 en constitue un premier élément de réponse.

Chapitre 7

Du choix de la règle pivot pour l'escalade de paysages de fitness

Nous avons vu dans le chapitre 5 (section 5.6) qu'une stratégie d'escalade purement intensificatrice (dénommée ici *climber*) est dictée par deux éléments structurels : la règle pivot et la gestion de la neutralité. Bien sûr, la politique des mouvements neutres peut constituer un élément de définition supplémentaire des règles pivot, mais nous choisissons de dissocier ces deux points de choix afin de mieux identifier les facteurs d'efficacité des climbers. Dans ce chapitre, nous comparons en premier lieu les règles pivot généralement utilisées dans les algorithmes de recherche locale — *premier améliorant* et *meilleur améliorant* —, puis nous explorons dans un second temps des règles moins conventionnelles mais s'avérant très pertinentes, en considérant en particulier la règle du *moins bon améliorant*.

7.1 Comparaison des stratégies du premier et meilleur améliorant

Comparer les stratégies de type premier et meilleur améliorant a fait l'objet de plusieurs études, mais généralement focalisées sur un problème particulier ([Breedam, 2001] pour un problème de tournées de véhicule, [Hansen et Mladenović, 2006] pour le TSP, ou plus récemment [Whitley *et al.*, 2013] pour MAXSAT). Des travaux sur les paysages NK ont été réalisés par Ochoa *et al.* [2010] via le modèle des réseaux d'optimums locaux [Tomassini *et al.*, 2008]. Par son caractère exact, cette étude impliquait des paysages NK de petite taille ($N = 16$ au maximum). Une des informations intéressantes pour notre propos présent est que sur ces paysages NK, les bassins d'attraction des optimums globaux relatifs au meilleur améliorant sont légèrement plus grands que ceux relatifs au premier améliorant. En revanche, la corrélation entre la taille des bassins d'attraction des optimums locaux et leur fitness dépend du niveau épistatique du paysage, comme l'indique la figure 26. Il y a quelques années, avant d'avoir eu connaissance de ces travaux, des tests comparatifs préliminaires sur divers paysages de fitness de plus grande taille nous conduisaient à des résultats similaires ; à la différence près que nous n'utilisons pas le modèle des réseaux d'optimums locaux mais simplement des moyennes de fitness atteintes depuis un

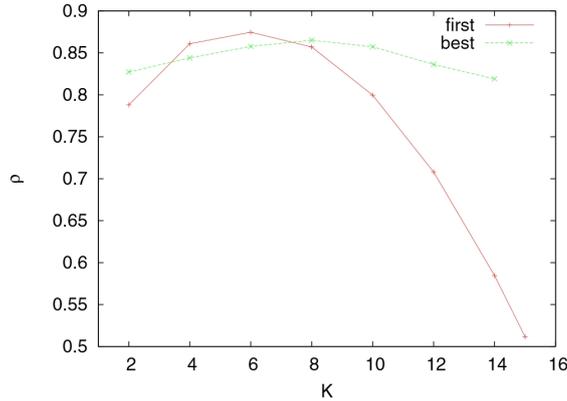


FIGURE 26 – Corrélacion entre taille des bassins d’attraction et fitness (ρ) en fonction de K , relativement à la stratégie du premier ou du meilleur améliorant. Figure extraite de [Ochoa *et al.*, 2010].

échantillonnage de points de départ. Ce qui, à l’approximation près, revient au même.

Nous résumons dans cette section la comparaison empirique des deux principales règles pivot sur un large panel de paysages de fitness, dans le de but de déterminer à quel point les propriétés des paysages ont une influence sur l’efficacité relative des stratégies de recherche. Cette abstraction en paysages de fitness, certes construits pour la plupart à partir d’instances de problèmes concrets, va nous permettre d’établir un premier lien entre propriétés des paysages mesurables par des indicateurs — comme la rugosité locale — et comportement prédictif des climbers. Nous rappelons plus précisément le mécanisme des deux climbers testés ici (algorithmes 5 et 6). Ces descriptions d’algorithmes classiques, peut-être moins simplifiées qu’à l’habitude, permettent de mesurer rigoureusement le nombre effectif d’appels à la fonction de fitness f . Dans la présentation des résultats et les discussions, nous noterons ces deux climbers FIRST et BEST.

Algorithme 5 : Algorithme du premier améliorant (FIRST).

Données : un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$, un point de départ $x_0 \in \mathcal{X}$.

Résultat : un optimum local x_{opt} et sa fitness F_{opt} .

```

1  $F \leftarrow f(x_0)$ ;
2  $N \leftarrow \mathcal{N}(x_0)$ ;
3 tant que  $N \neq \emptyset$  faire
4   | Sélectionner aléatoirement  $x' \in N$ ;
5   |  $F' \leftarrow f(x')$ ;
6   | si  $F' > F$  alors
7   |   |  $x \leftarrow x'$ ;
8   |   |  $F \leftarrow F'$ ;
9   |   |  $N \leftarrow \mathcal{N}(x)$ 
10  | sinon  $N \leftarrow N \setminus \{x'\}$ 
11 fin
12  $x_{\text{opt}} \leftarrow x$ ;
13  $F_{\text{opt}} \leftarrow F$ ;
14 retourner  $(x_{\text{opt}}, F_{\text{opt}})$ 

```

Algorithme 6 : Algorithme du meilleur améliorant (BEST).

Données : un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$, un point de départ $x_0 \in \mathcal{X}$.

Résultat : un optimum local x_{opt} et sa fitness F_{opt} .

```
1  $x \leftarrow x_0$  ;
2  $F \leftarrow f(x_0)$  ;
3 bloque  $\leftarrow$  faux ;
4 répéter
5    $N \leftarrow \mathcal{N}(x)$  ;
6    $F_b \leftarrow -\infty$  ;
7   tant que  $N \neq \emptyset$  faire
8     Sélectionner aléatoirement  $x' \in N$  ;
9      $F' \leftarrow f(x')$  ;
10    si  $F' > F_b$  alors
11       $x_b \leftarrow x'$  ;
12       $F_b \leftarrow F'$ 
13    fin
14     $N \leftarrow N \setminus \{x'\}$ 
15  fin
16  si  $F_b > F$  alors
17     $x \leftarrow x_b$  ;
18     $F \leftarrow F_b$ 
19  sinon bloque  $\leftarrow$  vrai
20 jusqu'à bloque = vrai ;
21  $x_{\text{opt}} \leftarrow x$  ;
22  $F_{\text{opt}} \leftarrow F$  ;
23 retourner  $(x_{\text{opt}}, F_{\text{opt}})$ 
```

Expérimentation

L'étude empirique a consisté à comparer les optimums locaux atteints par FIRST et BEST sur un large panel de paysages de fitness. Sont utilisés ici :

- 16 paysages NK standards, non neutres, obtenus à partir de 16 paramétrages différents ($N \in \{128, 256, 512, 1024\}$, $K \in \{1, 2, 4, 8\}$) ;
- les 288 paysages obtenus à partir d'instances de problèmes combinatoires présentés section 5.5 : 27 paysages FSP_{ins}, 132 paysages QAP_{swap} et 129 paysages MAXSAT.

Cette comparaison des règles pivot est effectuée à partir de climbers stricts, qui n'acceptent à aucun moment des mouvements neutres. C'est pourquoi ce panel d'instances n'inclut pas de paysages NK neutres. Ces derniers seront considérés dans le prochain chapitre lors de l'évaluation des politiques de mouvements neutres.

La rugosité de chaque paysage a été estimée par échantillonnage. Puisque les climbers évalués ici acceptent uniquement des voisins strictement améliorants, l'évaluation de la rugosité est effectuée via l'indicateur $\rho_k^>$. Pour estimer $\rho_k^>$ par échantillonnage, nous calculons pour toutes les valeurs possibles de k (allant de 1 au diamètre du paysage) un ensemble d'un million de valeurs d'épistasie de signe $\xi^>(x, x', y, y')$ obtenues à partir de k -échelles générées aléatoirement.

En analysant les différentes valeurs de k -rugosité estimées sur les différents paysages, il apparaît que les rugosités locales ($\rho_1^>$) sont suffisantes pour ordonner les différents paysages par niveau de rugosité. En effet, sur l'ensemble des paysages de fitness considérés dans cette étude, augmenter k n'affecte pas le niveau de rugosité relatif entre les paysages, contrairement aux paysages-exemples artificiels présentés dans la section 5.3.3 (figures 18 et 20). La figure 27 illustre ce point sur différents paysages NK. Ainsi et pour plus de clarté, seule la rugosité locale $\rho_1^>$ sera reportée dans les tables de résultats. En outre, dans un contexte d'escalade stricte, c'est la rugosité locale qui peut constituer un éventuel frein à l'escalade des paysages. Au contraire, la rugosité à de plus hautes échelles conditionne davantage l'impact des méthodes perturbatives qui ne sont pas évaluées ici et feront l'objet de travaux futurs. La neutralité des paysages n'est pas examinée dans cette étude et sera estimée dans le prochain chapitre dédié.

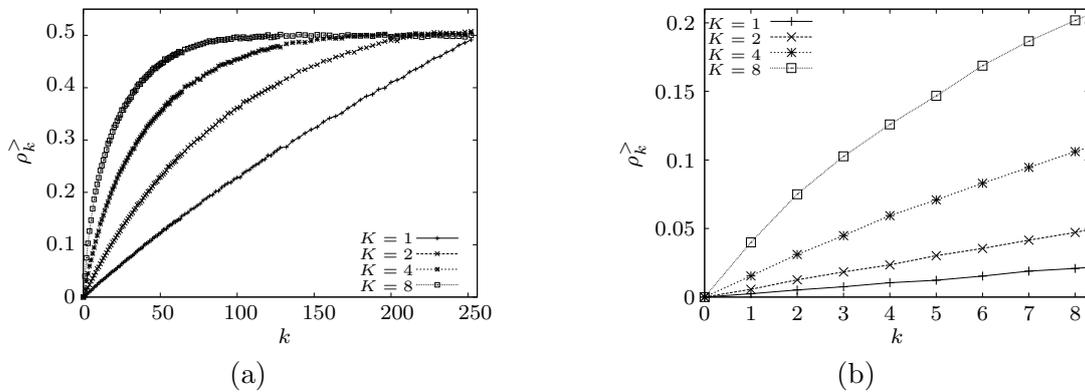


FIGURE 27 – (a) Évolution des k -rugosités estimées sur différents paysages NK (avec $N = 256$). (b) Focus sur les plus petites valeurs de k .

Chacun des deux algorithmes FIRST et BEST ont été exécutés 100 fois sur tous les paysages. Pour chaque paysage, ces climbers débutent leur recherche à partir du même ensemble de 100 points de départ, afin de supprimer l'éventuel biais relatif à l'initialisation des recherches. Le critère d'arrêt est la détection d'un optimum local. On reporte alors la fitness moyenne des 100 optimums trouvés. Dans les tables de résultats présentées ci-après, pour chaque paysage la meilleure moyenne apparaît en gras. Un fond gris indique que la méthode n'est pas statistiquement dominée par l'autre selon le test de Mann-Whitney (gris foncé avec une valeur p inférieure à 0.05, gris clair avec une valeur p inférieure à 0.1). Nous obtenons ainsi trois niveaux de dominance entre les climbers :

- $A \succ_{95\%} B$ signifie que le climber A domine statistiquement le climber B avec $p \leq 0.05$ (relativement au test de Mann-Whitney) sur le paysage ou l'ensemble de paysages considéré ;
- $A \succ_{90\%} B$ signifie que A domine statistiquement B avec $p \leq 0.1$;
- $A \succ_{\text{avg}} B$ signifie que A est plus performant que B en moyenne, mais ne domine pas nécessairement B statistiquement avec $p \leq 0.1$.

Les tables 16 à 19 résument les résultats obtenus par les climbers stricts utilisant respectivement la stratégie du premier (FIRST) et du meilleur (BEST) améliorant, sur différents types de paysages de fitness combinatoires : NK (table 16), QAP_{swap} (table 17), FSP_{ins} (table 18) et MAXSAT (table 19).

Paysages NK

Les résultats des climbers sur les paysages NK, dont la rugosité et la dimension sont paramétrables en fonction de N et K , sont particulièrement informatifs. En étudiant la table 16, nous constatons que la stratégie du meilleur améliorant domine statistiquement celle du premier améliorant pour $K \in \{1, 2\}$, tandis que le premier améliorant apparaît comme plus efficace lorsque K — et par conséquent la rugosité — croît. Une vue rapide des résultats pourrait laisser penser que la taille du paysage (2^N) semble ne pas affecter l’efficacité relative des deux variantes. Cependant, on peut remarquer que le niveau de rugosité des paysages (ρ) n’est pas uniquement déterminé par K mais décroît lorsque N augmente. Par conséquent, l’efficacité relative des climbers est déterminée par la rugosité des paysages, mais aussi par leur taille. La figure 28.a met en évidence la relation entre rugosité, taille et efficacité relative entre FIRST et BEST. Elle indique que plus un paysage NK est rugueux et large, plus la stratégie du premier améliorant tend à être plus efficace qu’un climber basé sur le meilleur améliorant.

Instance	$\rho_1^>$	FIRST	BEST
128.1	0.52%	.7021	.7090
128.2	1.24%	.7021	.7082
128.4	3.12%	.7254	.7260
128.8	7.40%	.7142	.7108
256.1	0.26%	.7021	.7079
256.2	0.61%	.7066	.7094
256.4	1.57%	.7235	.7204
256.8	3.98%	.7166	.7122

Instance	$\rho_1^>$	FIRST	BEST
512.1	0.12%	.6897	.6953
512.2	0.32%	.7135	.7174
512.4	0.80%	.7200	.7193
512.8	2.06%	.7206	.7155
1024.1	0.07%	.6969	.7039
1024.2	0.14%	.7146	.7197
1024.4	0.39%	.7246	.7242
1024.8	1.05%	.7216	.7174

TABLE 16 – Comparaison entre FIRST et BEST sur des paysages NK standards.

Paysages QAP_{swap}

La table 17 résume les résultats obtenus sur les paysages QAP_{swap} qui, contrairement aux paysages NK, sont dérivés d’instances d’un problème combinatoire académique. Puisque les caractéristiques des paysages ne peuvent être réglées en amont, nous avons réalisé l’étude expérimentale sur l’intégralité des paysages QAP_{swap} définis dans la section 5.5, ce qui laisse davantage de possibilités de considérer des paysages diversifiés. Pour résumer les résultats, on dénombre que FIRST $\succ_{95\%}$ BEST sur 32 des 124 paysages, alors que BEST $\succ_{95\%}$ FIRST sur 9 paysages. En particulier, BEST semble plus approprié pour escalader les paysages basés sur les instances de Burkard & Offermann (bur*), et Christofides & Benavent (chr*). FIRST est particulièrement adapté pour escalader les paysages Skorin-Kapov (sko*), Steinberg (ste*) et les plus grandes des instances Taillard (tai*). La figure 28.b tend à respecter les observations effectuées pour les paysages NK, à savoir que

Instance	$\rho_1^>$	FIRST	BEST
bur26a	9.7%	-5 444 341.1	-5 444 564.6
bur26b	9.7%	-3 835 171.4	-3 832 617.9
bur26c	9.9%	-5 452 501.6	-5 446 368.8
bur26d	9.8%	-3 842 352.8	-3 836 063.4
bur26e	9.7%	-5 407 830.3	-5 407 339.0
bur26f	9.7%	-3 801 482.2	-3 799 065.6
bur26g	9.6%	-10 161 339.5	-10 157 156.1
bur26h	9.6%	-7 133 940.0	-7 126 406.7
chr12a	19.9%	-14 595.1	-14 430.5
chr12b	19.4%	-15 512.3	-14 800.7
chr12c	20.0%	-15 903.8	-15 283.3
chr15a	16.4%	-15 019.9	-14 655.5
chr15b	15.9%	-13 455.4	-12 803.9
chr15c	16.6%	-15 703.7	-15 597.1
chr18a	14.2%	-19 067.4	-18 400.7
chr18b	13.5%	-1 813.5	-1 797.4
chr20a	13.1%	-3 305.0	-3 241.8
chr20b	13.3%	-3 325.4	-3 212.6
chr20c	12.9%	-26 596.7	-26 280.5
chr22a	11.9%	-7 027.7	-7 039.8
chr22b	12.0%	-7 127.8	-7 050.0
chr25a	10.6%	-6 025.1	-6 027.0
els19	12.8%	-22 280 466.0	-21 630 921.3
esc16a	14.8%	-70.3	-70.7
esc16b	12.4%	-292.0	-292.0
esc16c	16.1%	-163.1	-161.4
esc16d	15.0%	-17.8	-17.4
esc16e	12.8%	-30.9	-30.7
esc16g	12.7%	-29.0	-28.5
esc16h	14.4%	-996.0	-996.0
esc16i	13.5%	-14.2	-14.1
esc16j	11.1%	-9.3	-9.1
esc32a	9.5%	-160.5	-160.4
esc32b	10.3%	-216.9	-217.1
esc32c	8.9%	-646.6	-645.2
esc32d	8.3%	-214.3	-213.2
esc32e	3.6%	-2.0	-2.0
esc32f	3.5%	-2.0	-2.0
esc32g	3.3%	-6.1	-6.1
esc32h	8.9%	-455.0	-454.0
esc64a	2.8%	-118.4	-118.0
had12	18.9%	-1 676.5	-1 673.7
had14	17.4%	-2 751.4	-2 750.2
had16	16.7%	-3 753.5	-3 754.8
had18	15.3%	-5 425.2	-5 424.2
had20	14.2%	-7 006.5	-7 000.1
kra30a	10.6%	-95 771.1	-95 854.8
kra30b	10.6%	-96 633.4	-96 428.5
kra32	10.0%	-18 974.8	-19 032.8
lipa20a	15.0%	-3 788.7	-3 789.3
lipa20b	16.5%	-31 305.1	-31 137.5
lipa30a	10.9%	-13 447.1	-13 441.4
lipa30b	13.0%	-176 474.9	-176 739.0
lipa40a	8.5%	-32 014.0	-32 014.9
lipa40b	10.9%	-565 732.7	-566 510.7
lipa50a	7.0%	-62 888.4	-62 907.6
lipa50b	9.5%	-1 434 252.1	-1 436 411.0
lipa60a	5.9%	-108 401.4	-108 403.8
lipa60b	8.6%	-3 019 482.0	-3 025 897.5
lipa70a	5.2%	-171 380.6	-171 400.4
lipa70b	7.8%	-5 543 957.8	-5 559 790.9
lipa80a	4.5%	-255 307.6	-255 334.4
lipa80b	7.3%	-9 435 185.1	-9 444 877.8
lipa90a	4.1%	-363 384.1	-363 429.9
lipa90b	6.8%	-15 203 151.5	-15 216 234.1

Instance	$\rho_1^>$	FIRST	BEST
nug12	21.2%	-610.1	-610.6
nug14	18.8%	-1 067.9	-1 062.5
nug15	18.3%	-1 204.8	-1 201.3
nug16a	17.4%	-1 688.2	-1 692.0
nug16b	17.3%	-1 310.3	-1 307.2
nug17	16.4%	-1 809.3	-1 803.5
nug18	15.9%	-2 019.7	-2 018.5
nug20	14.9%	-2 676.9	-2 678.5
nug21	14.3%	-2 547.4	-2 547.9
nug22	13.4%	-3 731.1	-3 722.0
nug24	13.2%	-3 657.9	-3 643.9
nug25	12.9%	-3 884.6	-3 887.5
nug27	12.1%	-5 454.7	-5 453.1
nug28	12.0%	-5 391.9	-5 398.2
nug30	11.6%	-6 368.5	-6 384.8
rou12	22.5%	-248 706.8	-248 356.4
rou15	19.8%	-379 573.7	-378 164.5
rou20	16.4%	-759 465.0	-764 630.1
scr12	19.3%	-33 702.6	-33 710.7
scr15	16.2%	-57 027.9	-56 401.3
scr20	13.4%	-121 454.2	-121 880.8
sko100a	5.6%	-154 795.6	-155 195.2
sko100b	5.6%	-156 675.3	-156 962.4
sko100c	5.7%	-150 997.6	-151 312.0
sko100d	5.7%	-152 286.5	-152 626.8
sko100e	5.7%	-152 173.4	-152 600.8
sko100f	5.7%	-151 733.4	-152 129.3
sko42	9.6%	-16 321.2	-16 370.3
sko49	8.8%	-24 041.9	-24 160.9
sko56	8.0%	-35 369.7	-35 474.5
sko64	7.5%	-49 707.8	-49 783.0
sko72	6.9%	-67 898.9	-67 965.8
sko81	6.4%	-92 874.7	-93 124.4
sko90	6.0%	-117 860.5	-118 112.9
ste36a	8.1%	-10 603.0	-10 639.1
ste36b	7.9%	-18 992.7	-19 078.2
ste36c	7.9%	-8 945 385.7	-9 021 581.7
tail00a	6.4%	-21 811 727.6	-21 843 265.6
tail00b	4.4%	-1 237 857 238.1	-1 249 060 042.6
tail0a	24.6%	-143 711.7	-142 578.5
tail0b	22.7%	-1 339 375.6	-1 319 043.7
tail2a	21.7%	-244 924.4	-245 395.6
tail2b	19.0%	-44 193 333.3	-43 651 098.9
tail5a	19.9%	-406 509.0	-407 274.5
tail5b	15.9%	-52 063 153.5	-52 050 124.0
tail7a	18.4%	-520 972.3	-519 811.3
tai20a	16.6%	-747 454.7	-745 418.7
tai20b	12.9%	-141 012 824.1	-142 677 266.0
tai25a	14.6%	-1 227 561.6	-1 228 539.7
tai25b	11.2%	-391 785 290.8	-405 482 117.5
tai30a	13.0%	-1 907 004.3	-1 911 719.4
tai30b	10.3%	-717 135 475.2	-726 629 805.2
tai35a	11.9%	-2 536 171.5	-2 550 449.9
tai35b	9.4%	-306 198 132.1	-306 229 161.2
tai40a	11.0%	-3 293 480.9	-3 296 135.1
tai40b	8.9%	-699 305 121.9	-703 984 330.1
tai50a	9.6%	-5 175 285.5	-5 185 014.7
tai50b	7.5%	-487 675 806.5	-491 485 627.5
tai60a	8.6%	-7 535 747.8	-7 544 353.2
tai60b	6.3%	-649 061 353.2	-657 167 128.7
tai64c	2.3%	-1 865 318.8	-1 864 589.2
tai80a	7.3%	-14 032 294.4	-14 075 017.0
tai80b	5.3%	-863 181 601.3	-869 008 411.8
tho30	11.6%	-157 233.3	-156 888.5
tho40	9.4%	-251 350.9	-251 944.3
wil50	8.8%	-49 557.6	-49 544.9
wil100	5.7%	-275 844.9	-275 987.0

TABLE 17 – Comparaison entre FIRST et BEST sur des paysages QAP_{swap}.

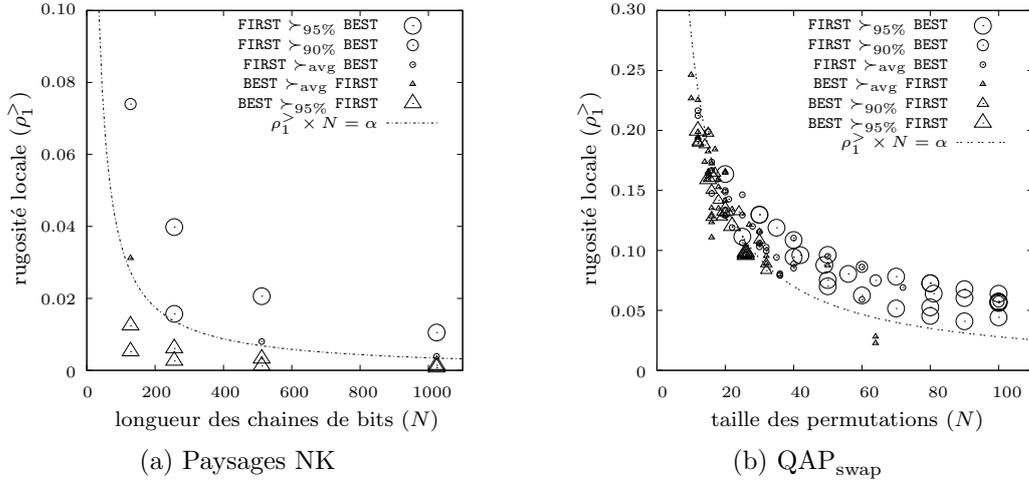


FIGURE 28 – Niveau de dominance du climber FIRST (cercles) ou BEST (triangles) sur les paysages NK et QAP_{swap} en fonction de leur taille et rugosité. La ligne pointillée aide à identifier une corrélation entre rugosité, taille et efficacité de la règle pivot utilisée (ici, $\alpha = 3.5$ pour les paysages NK et $\alpha = 2.8$ pour les paysages QAP_{swap}).

l'efficacité comparée des climbers dépend à la fois de la rugosité et de la taille des paysages. D'ailleurs, on observe qu'il semble exister un indicateur permettant de discriminer la règle pivot la plus adaptée en fonction des caractéristiques du paysage. Que ce soit à partir des expérimentations sur les paysages NK ou QAP_{swap} , un indicateur simple basé sur la taille et la rugosité locale fournit une catégorisation convenable des paysages relative à la capacité d'un climber FIRST ou BEST à les escalader. En particulier, la stratégie du premier améliorant semble plus adaptée pour escalader efficacement les instances les plus difficiles, soit les paysages plus larges et possédant un niveau de rugosité plus important.

Paysages FSP_{ins} (flowshop)

La supériorité de FIRST est particulièrement nette sur les paysages FSP_{ins} , où il domine systématiquement BEST en moyenne, avec $\text{FIRST} \succ_{95\%} \text{BEST}$ sur 23 des 27 paysages (voir table 18). Un point intéressant est que dans la plupart des travaux traitant du flowshop, les algorithmes de recherche locale définis sont justement basés sur la stratégie du premier améliorant — comme [Juan *et al.*, 2013] pour une recherche locale itérée, ou [Taillard, 1990; Ben-Daya et Al-Fawzan, 1998] pour des recherches tabou spécifiques employant la règle du premier améliorant. Pourtant, dans ces publications, l'utilisation de FIRST n'est généralement pas discutée ou justifiée, et résulte peut-être davantage d'aspects pratiques et de problématiques liées aux temps de calcul ; en effet, le voisinage, en $O(n^2)$, est plutôt large, et son évaluation incrémentale n'offre pas de bénéfice particulier.

Instance	$\rho_1^>$	FIRST	BEST
20.5_01.ta001	10.1%	-1 306.7	-1 311.0
20.10_01.ta011	12.0%	-1 629.3	-1 634.3
20.15_01	12.3%	-1 961.4	-1 972.6
20.20_01.ta021	13.2%	-2 374.7	-2 391.9
30.5_01	7.4%	-1 736.8	-1 737.5
30.10_01	10.0%	-2 072.2	-2 084.7
30.15_01	10.6%	-2 510.7	-2 516.8
30.20_01	11.5%	-2 856.6	-2 869.5
50.5_01.ta031	4.9%	-2 743.3	-2 749.9
50.10_01.ta041	7.6%	-3 140.2	-3 147.7
50.15_01	8.4%	-3 535.6	-3 546.7
50.20_01.ta051	9.1%	-4 033.7	-4 056.0
70.5_01	4.1%	-3 794.4	-3 799.3
70.10_01	6.2%	-4 168.0	-4 175.2

Instance	$\rho_1^>$	FIRST	BEST
70.15_01	7.2%	-4 566.6	-4 580.1
70.20_01	7.7%	-4 980.5	-5 004.1
100.5_01.ta061	4.0%	-5 510.0	-5 516.5
100.10_01.ta071	5.2%	-5 864.5	-5 894.9
100.15_01	6.0%	-6 155.3	-6 182.8
100.20_01.ta081	6.6%	-6 560.5	-6 614.2
150.5_01	2.5%	-8 066.3	-8 068.6
150.10_01	4.3%	-8 240.6	-8 284.8
150.15_01	4.8%	-8 676.8	-8 712.2
150.20_01	5.5%	-9 327.7	-9 396.8
200.10_01.ta091	3.6%	-11 006.3	-11 049.2
200.15_01	4.3%	-11 400.0	-11 469.2
200.20_01.ta101	4.7%	-11 650.7	-11 711.4

TABLE 18 – Comparaison entre FIRST et BEST sur des paysages FSP_{ins} (flowshop).

Paysages MAXSAT

Les résultats obtenus sur les paysages MAXSAT (table 19), ne sont pas en parfaite concordance avec les précédents. En effet, les indicateurs n (pour la taille) et $\rho_1^>$ ne permettent pas de consensus clair pour discriminer FIRST et BEST. Cependant, une tendance notable existe pour les paysages dérivés d’instances 3-SAT aléatoires, où FIRST n’est jamais statistiquement dominé. Sur les instances construites à partir du codage SAT d’autres problèmes, les résultats semblent dépendre de caractéristiques spécifiques à chaque type d’instances.

La difficulté de discriminer les paysages MAXSAT avec des indicateurs généraux est vraisemblablement typique de ce problème et particulièrement des instances SAT construites à partir d’instances de problèmes combinatoires. Whitley *et al.* [2013] arrivent à des conclusions similaires aux nôtres, à savoir que la stratégie du premier améliorant produit de meilleurs résultats sur des instances MAXSAT aléatoires uniformes (partie gauche de la table 19), et que le meilleur améliorant produit de meilleurs résultats sur des problèmes industriels (partie droite de la table 19, pour différents types de problèmes). Ils expliquent ce phénomène par le fait que, contrairement aux instances SAT aléatoires dans lesquelles les interactions entre variables sont uniformément distribuées, les problèmes industriels comportent généralement un ensemble de variables critiques qui ont davantage d’effet sur la fonction de fitness et qui doivent être fixées au cours des premières étapes de la recherche, ce que permet la stratégie du meilleur améliorant. Cette hétérogénéité de l’impact des variables sur la fitness amène très probablement les instances à ne pas pouvoir être caractérisées de manière adéquate au moyen d’indicateurs classiques. Des indicateurs alternatifs pourraient représenter cette hétérogénéité dans les paysages en mesurant par exemple la variabilité des interactions épistatiques.

Nombre d’évaluations

Cette étude vise à comparer l’efficacité des climbers selon la qualité des optimums locaux atteints, indépendamment des coûts computationnels. Néanmoins, comme attendu, FIRST requiert systématiquement moins d’évaluations (fitness envisagées au cours de la recherche) que BEST pour atteindre un optimum local. Le ratio moyen $\frac{\#éval(FIRST)}{\#éval(BEST)}$ constaté durant nos expérimentations, en considérant toutes les exécutions, instances et types de paysages, est de l’ordre de 13%. En particulier, ce ratio varie de 1.5% à 19% pour les

Instances 3-SAT aléatoires				
Instance	n	$\rho_1^>$	FIRST	BEST
uf20-01	20	1.81%	-2.39	-2.24
uf50-01	50	0.31%	-5.25	-5.62
uuf50-01	50	0.31%	-6.46	-6.58
uf75-01	75	0.18%	-8.67	-8.90
uuf75-01	75	0.15%	-9.24	-8.83
uf100-01	100	0.09%	-10.77	-11.44
uuf100-01	100	0.08%	-11.34	-11.38
uf125-01	125	0.05%	-12.29	-13.11
uuf125-01	125	0.05%	-14.10	14.78
uf150-01	150	0.03%	-14.76	-15.27
uuf150-01	150	0.03%	-15.42	-15.73
uf175-01	175	0.03%	-18.26	-18.22
uuf175-01	175	0.03%	-19.25	-19.71
uf200-01	200	0.02%	-20.67	-21.94
uuf200-01	200	0.03%	-23.42	24.27
uf225-01	225	0.02%	-21.58	-22.83
uuf225-01	225	0.02%	-24.79	-25.46
uf250-01	250	0.01%	-23.66	24.38
uuf250-01	250	0.01%	-25.67	-26.83
RTL....m429_0	100	0.08%	-9.51	-9.96
BMS....m429_0	100	0.03%	-7.89	-8.11
CBS....m403.b10.0	100	0.07%	-9.50	-9.61
CBS....m403.b30.0	100	0.08%	-9.48	-9.46
CBS....m403.b50.0	100	0.07%	-9.59	-9.81
CBS....m403.b70.0	100	0.07%	-9.98	10.50
CBS....m403.b90.0	100	0.09%	-10.66	-11.02
CBS....m411.b10.0	100	0.08%	-9.42	-9.82
CBS....m411.b30.0	100	0.07%	-10.01	-10.37
CBS....m411.b50.0	100	0.05%	-11.03	-10.87
CBS....m411.b70.0	100	0.06%	-10.69	-10.78
CBS....m411.b90.0	100	0.06%	-10.84	-10.78
CBS....m418.b10.0	100	0.08%	-8.58	-9.00
CBS....m418.b30.0	100	0.09%	-9.83	-9.98
CBS....m418.b50.0	100	0.08%	-10.51	-10.74
CBS....m418.b70.0	100	0.11%	-10.20	-10.12
CBS....m418.b90.0	100	0.08%	-10.34	-10.53
CBS....m423.b10.0	100	0.09%	-9.99	-10.56
CBS....m423.b30.0	100	0.09%	-9.76	-10.00
CBS....m423.b50.0	100	0.09%	-9.87	-9.80
CBS....m423.b70.0	100	0.09%	-8.96	-9.01
CBS....m423.b90.0	100	0.10%	-10.31	-10.14
CBS....m429.b10.0	100	0.08%	-9.83	-9.52
CBS....m429.b30.0	100	0.08%	-9.28	-9.31
CBS....m429.b50.0	100	0.11%	-10.55	-11.27
CBS....m429.b70.0	100	0.09%	-9.54	9.98
CBS....m429.b90.0	100	0.10%	-10.73	-10.87
CBS....m435.b10.0	100	0.09%	-9.21	9.65
CBS....m435.b30.0	100	0.09%	-10.35	-11.28
CBS....m435.b50.0	100	0.08%	-10.07	10.66
CBS....m435.b70.0	100	0.09%	-11.01	-11.42
CBS....m435.b90.0	100	0.09%	-11.17	-10.98
CBS....m441.b10.0	100	0.07%	-9.05	-9.82
CBS....m441.b30.0	100	0.09%	-9.86	-10.24
CBS....m441.b50.0	100	0.09%	-9.80	-9.71
CBS....m441.b70.0	100	0.08%	-10.59	11.08
CBS....m441.b90.0	100	0.07%	-11.53	-12.37
CBS....m449.b10.0	100	0.08%	-9.24	-10.03
CBS....m449.b30.0	100	0.10%	-11.00	-10.87
CBS....m449.b50.0	100	0.06%	-10.30	-10.58
CBS....m449.b70.0	100	0.09%	-11.48	-11.48
CBS....m449.b90.0	100	0.07%	-12.41	-12.70

Transformations SAT d'instances de divers problèmes				
Instance	n	$\rho_1^>$	FIRST	BEST
flat30-1	90	0.09%	-7.08	-5.76
flat50-1	150	0.06%	-12.45	-10.68
flat75-1	225	0.03%	-19.58	-16.88
flat100-1	300	0.02%	-25.81	-21.94
flat125-1	375	0.02%	-32.45	-26.94
flat150-1	450	0.02%	-39.77	-32.46
flat175-1	525	0.01%	-44.08	-37.82
flat200-1	600	0.01%	-51.11	-42.74
sw100-8-lp0-c5-1	500	0.00%	-18.68	-14.00
sw100-8-lp1-c5-1	500	0.00%	-19.19	-14.68
sw100-8-lp2-c5-1	500	0.00%	-22.23	-17.34
sw100-8-lp3-c5-1	500	0.00%	-24.91	-20.27
sw100-8-lp4-c5-1	500	0.00%	-27.98	-23.19
sw100-8-lp5-c5-1	500	0.00%	-29.98	-25.43
sw100-8-lp6-c5-1	500	0.00%	-30.72	-25.69
sw100-8-lp7-c5-1	500	0.00%	-31.25	-26.24
sw100-8-lp8-c5-1	500	0.00%	-31.68	-26.75
sw100-8-p0-c5-1	500	0.00%	-31.79	-27.17
bw_anomaly	48	0.67%	-8.68	-8.21
bw_medium	116	0.15%	-15.49	-14.77
bw_huge	459	0.01%	-37.93	-43.41
bw_large.a	459	0.01%	-37.85	-41.98
bw_large.b	1087	0.00%	-58.97	-78.66
bw_large.c	3016	0.00%	-98.06	-160.70
bw_large.d	6325	0.00%	-133.83	-265.98
logistics.a	828	0.00%	-49.45	-30.20
logistics.b	843	0.01%	-39.48	-22.78
logistics.c	1141	0.00%	-56.22	-33.29
logistics.d	4713	0.00%	-276.23	-399.81
ais6	61	0.15%	-2.23	-1.93
ais8	113	0.04%	-3.40	-2.92
ais10	181	0.01%	-4.34	-3.59
ais12	265	0.00%	-5.25	-4.46
qg1-07	343	0.00%	-37.26	-36.56
qg1-08	512	0.00%	-46.29	-47.76
qg2-07	343	0.00%	-36.69	-38.31
qg2-08	512	0.00%	-47.06	-49.55
qg3-08	512	0.00%	-77.82	-66.39
qg3-09	729	0.00%	-97.25	-82.44
qg4-08	512	0.00%	-70.61	-61.40
qg4-09	729	0.00%	-88.81	-77.00
qg5-09	729	0.00%	-122.38	-113.72
qg5-10	1000	0.00%	-153.18	-139.46
qg5-11	1331	0.00%	-187.56	-168.20
qg5-12	1728	0.00%	-222.36	-202.46
qg5-13	2197	0.00%	-263.75	-239.93
qg6-09	729	0.00%	-105.93	-99.58
qg6-10	1000	0.00%	-130.62	-123.89
qg6-11	1331	0.00%	-159.42	-148.69
qg6-12	1728	0.00%	-191.12	-178.70
qg7-09	729	0.00%	-118.49	-105.67
qg7-10	1000	0.00%	-145.32	-131.54
qg7-11	1331	0.00%	-174.86	-158.97
qg7-12	1728	0.00%	-209.06	-191.58
qg7-13	2197	0.00%	-244.80	-221.46
bmc-ibm-1	9685	0.00%	-676.73	-642.07
bmc-ibm-2	2810	0.02%	-292.17	-278.14
bmc-ibm-3	14930	0.00%	-1 772.67	-1 665.60
bmc-ibm-4	28161	0.00%	-2 968.92	-2 932.36
bmc-ibm-5	9396	0.00%	-1 019.46	-1 036.26
bmc-ibm-6	51639	0.00%	-4 658.90	-4 595.30
bmc-ibm-7	8710	0.00%	-896.42	-864.46
bmc-galileo-8	58074	0.00%	-7 183.37	-7 913.57
bmc-galileo-9	63624	0.00%	-7 824.87	-8 657.53
bmc-ibm-10	59056	0.00%	-5 795.01	-6 119.76
bmc-ibm-11	32109	0.00%	-3 436.25	-3 476.99
bmc-ibm-12	39598	0.00%	-3 968.42	-4 064.75
bmc-ibm-13	13215	0.00%	-1 478.34	-1 418.25

TABLE 19 – Comparaison entre FIRST et BEST sur des paysages MAXSAT.

paysages NK, de 3% à 21% pour les paysages QAP_{swap} , de 7% à 38% pour les paysages FSP_{ins} , et de 0.05% à 33% pour les paysages MAXSAT. Pour tous les types de paysages, ce ratio décroît lorsque la taille du paysage augmente.

Le plus souvent, que ce soit sur des critères de qualité (hauteurs atteintes en moyenne), de rapidité, de robustesse (FIRST est moins déterministe, ce qui laisse davantage de potentiel à une version itérée) et de passage à l'échelle (l'avantage de FIRST sur BEST est plus net sur les paysages plus larges) — soit les quatre critères, évoqués dans le premier chapitre, permettant de juger de l'efficacité d'un algorithme de recherche approché —, un climber basé sur la stratégie du premier améliorant possède un avantage notable sur l'autre version communément utilisée. Si ce résultat lève d'éventuels doutes quant au choix de la règle pivot lors de la conception d'un climber, et plus généralement d'une recherche locale, la suite de ce chapitre apporte des éléments supplémentaires pour comprendre ce phénomène et améliorer l'efficacité des recherches par voisinage.

7.2 Règles pivot alternatives et stratégie du moins bon améliorant

Dans la section précédente, nous comparons les deux règles pivot habituellement envisagées lors de la conception d'algorithmes de recherche locale et de climbers en particulier : premier et meilleur améliorant. Nous montrons empiriquement que le premier améliorant permet de déterminer de meilleurs optimums locaux que le meilleur améliorant sur la plupart des paysages combinatoires, en particulier sur les plus larges ou plus rugueux ; autrement dit, que sur les paysages dits ou considérés comme difficiles, sélectionner systématiquement le plus haut voisin conduit le plus souvent à être piégé dans des optimums locaux de moins bonne qualité.

Intuitivement, puisqu'une stratégie de type meilleur améliorant choisit systématiquement le plus haut voisin, il est possible qu'elle conduise la recherche vers les pics les plus proches, et donc vers les optimums locaux les plus proches. Au contraire, en effectuant plus souvent des pas réduits, une stratégie de type premier améliorant tendrait à conduire progressivement la recherche vers des zones plus hautes du paysage, où les optimums locaux potentiels sont également plus hauts. En extrapolant, on peut alors penser que choisir le moins haut voisin améliorant à chaque pas de la recherche augmentera la probabilité d'atteindre de plus hautes zones du paysage en évitant les petits pics pentus. C'est ainsi que nous considérons dans cette section la stratégie dite du *moins bon améliorant* (algorithme 7 noté WORST), mais aussi d'autres règles pivot intermédiaires qui permettront de déterminer à quel point les stratégies qui favorisent la sélection des moins hauts voisins améliorants peuvent mener à de plus hauts optimums locaux.

Précisons que cette stratégie du moins bon améliorant (*worst improvement* [Basseur et Goëffon, 2013], mais rétrospectivement l'appellation *least improvement* eût été préférable) a déjà été évoquée à quelques reprises dans des articles ou tutoriaux [Stützle et Dumitrescu, 2003; Hoos et Stützle, 2004; Hansen et Mladenović, 2006; Alekseeva *et al.*, 2008]. À notre connaissance, seuls Alekseeva *et al.* [2008], dans une étude sur le problème du p -médiann, indiquent que cette règle pivot peut permettre d'atteindre des optimums locaux

de meilleure qualité que la règle du meilleur améliorant. Les très rares autres allusions à cette stratégie sont en réalité théoriques, pour illustrer la définition de ce qu'est une règle pivot au moyen d'exemples simples. Sans justification à l'appui, Stützle et Dumitrescu [2003] la mentionnent ainsi comme une stratégie *concevable*, mais qu'il est raisonnable de ne pas utiliser en pratique. L'étude que nous avons menée sur la stratégie du moins bon améliorant — et d'autres variantes — présentée dans ce chapitre, est ainsi de notre point de vue particulièrement originale.

Algorithme 7 : Algorithme du moins bon améliorant (WORST).

Données : un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$, un point de départ $x_0 \in \mathcal{X}$.

Résultat : un optimum local x_{opt} et sa fitness F_{opt} .

```

1  $x \leftarrow x_0$ ;
2  $F \leftarrow f(x_0)$ ;
3 bloque  $\leftarrow$  faux;
4 répéter
5    $N \leftarrow \mathcal{N}(x)$ ;
6    $F_w \leftarrow +\infty$ ;
7   tant que  $N \neq \emptyset$  faire
8     Sélectionner aléatoirement  $x' \in N$ ;
9      $F' \leftarrow f(x')$ ;
10    si  $F' < F_w$  et  $F' > F$  alors
11       $x_w \leftarrow x'$ ;
12       $F_w \leftarrow F'$ 
13    fin
14     $N \leftarrow N \setminus \{x'\}$ 
15  fin
16  si  $F_w \neq +\infty$  alors
17     $x \leftarrow x_w$ ;
18     $F \leftarrow F_w$ 
19  sinon bloque  $\leftarrow$  vrai
20 jusqu'à bloque = vrai;
21  $x_{\text{opt}} \leftarrow x$ ;
22  $F_{\text{opt}} \leftarrow F$ ;
23 retourner  $(x_{\text{opt}}, F_{\text{opt}})$ 

```

7.2.1 Du meilleur au moins bon améliorant

Règles pivot basées sur le rang

Soit $I_x = (x_1, x_2, \dots, x_w)$ un uplet constitué de tous les voisins strictement améliorants (et distincts) de $x \in \mathcal{X}$ triés par fitness décroissante, soit $\bigcup_{i=1}^w \{x_i\} = \{x' \in \mathcal{N}(x), f(x') > f(x)\}$ et $\forall i \in \{1, \dots, w-1\}, f(x_i) \geq f(x_{i+1})$.

Étant donné un point x et son uplet associé $I_x = (x_1, x_2, \dots, x_w)$, nous définissons trois règles pivot basées sur le rang des voisins :

- S_i^- sélectionne, à chaque pas de la recherche, le i -ème meilleur voisin améliorant (x_i) ;

- S_i^{\geq} sélectionne aléatoirement l'un des i -èmes meilleurs voisins améliorants (appartenant à $\{x_1, x_2, \dots, x_i\}$);
- S_i^{\leq} sélectionne aléatoirement l'un des $(w - i + 1)$ -èmes moins bons voisins améliorant (appartenant à $\{x_i, x_{i+1}, \dots, x_w\}$).

Soit $q_1 = \lceil w/4 \rceil$, $m = \lceil w/2 \rceil$ et $q_3 = \lceil 3w/4 \rceil$ les indices des (respectivement) premier, deuxième et troisième quartiles de I_x . $b = 1$ et w dénotent les indices correspondant au meilleur et moins bon élément de I_x . Dans la suite, nous considèrerons $S_i^=$, S_i^{\geq} et S_i^{\leq} combinés à cinq valeurs de rang $i \in \{b, q_1, m, q_3, w\}$. Comme indiqué sur la figure 29, certaines stratégies sont équivalentes ($S_b^=$ et S_b^{\geq} , S_b^{\leq} et S_w^{\geq} , $S_w^=$ et S_w^{\leq}). Bien sûr, $S_b^=$ correspond à la stratégie du meilleur améliorant, $S_w^=$ au moins bon améliorant, et S_w^{\geq} au premier améliorant — à la différence près que la définition suppose que le voisinage ait été généré exhaustivement, ce qui peut être évité lors de la conception de S_w^{\geq} . $S_m^=$ peut être dénommé *améliorant médian*.

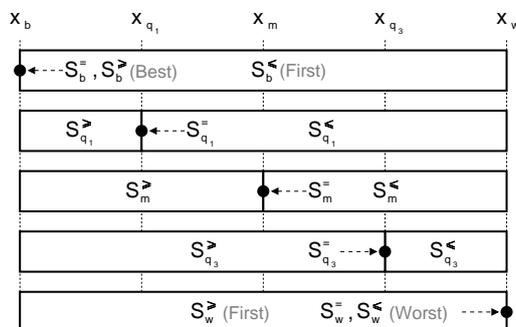


FIGURE 29 – Illustration de différentes règles pivot basées sur le rang. Les voisins améliorants sont classés du meilleur (x_b) au moins bon (x_w) — *plus haut* et *moins haut* en utilisant le vocabulaire réservé aux paysages de fitness. Ce classement détermine l'ensemble des voisins acceptables en fonction de la règle pivot. Par exemple, $S_{q_3}^{\geq}$ consiste à sélectionner aléatoirement un voisin parmi les plus hauts 75% des voisins améliorants.

Au final, nous définissons 12 règles pivot distinctes, en se basant sur le rang de qualité (hauteur) des voisins améliorants, ce qui inclut les stratégies du meilleur, premier et moins bon améliorant.

Expérimentation

L'étude expérimentale présentée ici vise à comparer les climbers utilisant différentes règles pivot basées sur le rang, en mesurant leur habileté respective à atteindre de hauts optimums locaux. Précisons que cette étude se focalise prioritairement sur la performance de la stratégie du moins bon améliorant. Les autres variantes nous aideront à évaluer l'effet de la règle pivot sur l'efficacité des recherches locales.

Nous utiliserons exclusivement dans cette étude les paysages NK standards, ce qui nous permet de régler différents niveaux de rugosité tout en ne considérant aucune neutralité. Pour être le plus précis possible, nous avons étendu le protocole expérimental défini dans

la section précédente³¹, en considérant 10 paysages au lieu d'un pour chaque paramétrage (N, K) différent, et en ajoutant de nouvelles valeurs de K pour inclure, en particulier, des paysages NK plus rugueux. Nous considérons ainsi 28 paramétrages (N, K) , correspondant à toutes les combinaisons de $N \in \{128, 256, 512, 1024\}$ et $K \in \{1, 2, 4, 6, 8, 10, 12\}$, soit au total 280 paysages. 100 points aléatoires, qui serviront de points de départ pour les climbers, ont été générés pour chaque paysage.

La comparaison expérimentale porte sur les 12 climbers définis ci-avant, chacun basé sur une règle pivot S spécifique : $S_b^=$, $S_{q_1}^=$, $S_m^=$, $S_{q_3}^=$, $S_w^=$, $S_{q_1}^>$, $S_m^>$, $S_{q_3}^>$, $S_w^>$, $S_{q_1}^<$, $S_m^<$, $S_{q_3}^<$. Ces 12 climbers ont été exécutés 100 fois — à partir de l'ensemble des points de départ fixés — sur chacun des 280 paysages. Toutes ces variantes étant des climbers stricts, les escalades s'arrêtent lorsqu'un optimum local est atteint.

L'analyse expérimentale s'attardera principalement sur la qualité des optimums locaux atteints par chacune des variantes. Pour chaque triplet (N, K, S) , nous considérons dans un premier temps la fitness moyenne des 1000 optimums locaux résultant des recherches correspondantes. Pour améliorer la lisibilité, nous reportons uniquement sur la table 20 les résultats pour les climbers de type $S_i^=$, ainsi que $S_w^>$ (premier améliorant) pour comparaison. Les résultats des autres variantes intermédiaires ne sont pas reportés dans cette table mais seront discutés ultérieurement. Pour chaque paramétrage (N, K) , la meilleure moyenne est indiquée en gras. Les valeurs sur fond gris sont celles dont les méthodes ne sont pas statistiquement dominées par une autre variante relativement à un test binomial (valeur p inférieure à 0.05). Puisque pour chaque paramétrage (N, K) , les performances des climbers sont évaluées sur l'escalade de 10 paysages différents, un test statistique de comparaison d'échantillons comme celui de Mann-Whitney n'est pas adapté pour cette analyse. Le test binomial effectué ici consiste à comparer, pour chaque paire de climbers, les optimums locaux obtenus selon chaque couple (instance, point de départ). Plus précisément, si \mathfrak{s} est le nombre de succès de la méthode A sur la méthode B en 1000 confrontations, alors A domine statistiquement B avec une valeur p inférieure à 0.05 lorsque $\frac{1}{2^{1000}} \sum_{i=0}^{\mathfrak{s}} \binom{1000}{i} \geq 0.95$, soit $\mathfrak{s} \geq 526$.

Les résultats numériques de la table 20.a permettent d'extraire des conclusions très claires. Tout d'abord, on peut observer que le meilleur améliorant ($S_b^=$) obtient systématiquement les meilleurs résultats moyens sur les instances dites lisses ($K = \{1, 2\}$), tandis que le moins bon améliorant ($S_w^=$) domine les autres variantes sur tous les autres paysages. Cette observation est très nettement confirmée par l'analyse statistique. K n'est cependant pas l'unique facteur discriminant quant à l'efficacité des climbers, puisqu'on peut remarquer que plus N croît, plus les écarts sont marqués. La section 7.1 montrait la supériorité du meilleur améliorant sur le premier améliorant sur les paysages lisses, et la supériorité du premier améliorant sur le meilleur améliorant sur les paysages plus rugueux et plus larges. Nous remarquons plus précisément ici que le moins bon améliorant domine le premier améliorant (colonne $S_w^>$) dès lors que ce dernier est plus efficace que le meilleur améliorant. Pour atteindre de bons optimums locaux et sur la base de ces résultats, il semble ainsi opportun de choisir entre le meilleur et le moins bon améliorant, en fonction de la rugosité du paysage. Cette tendance est confirmée par la figure 30, qui

³¹Cette étude a été assez largement postérieure à la précédente, et nous avons entre temps enrichi notre panel d'instances NK.

montre l'évolution des valeurs de fitness moyennes en fonction de la règle pivot utilisée. Elle indique que la qualité moyenne des optimums locaux atteints est largement corrélée avec le rang, donc la qualité, des voisins sélectionnés ; positivement sur les paysages NK les plus lisses, et négativement sur les plus rugueux (la taille des paysages est ici fixée, avec $N = 1024$).

N.K	S_b^-	$S_{q_1}^-$	S_m^-	$S_{q_3}^-$	S_w^-	S_w^{\geq}
128.1	.6973	.6916	.6877	.6845	.6809	.6896
256.1	.7033	.6986	.6966	.6944	.6918	.6980
512.1	.6937	.6886	.6856	.6828	.6803	.6872
1024.1	.6997	.6954	.6920	.6895	.6867	.6936
128.2	.7169	.7142	.7121	.7107	.7081	.7133
256.2	.7130	.7104	.7080	.7055	.7035	.7080
512.2	.7120	.7097	.7077	.7063	.7041	.7076
1024.2	.7132	.7109	.7092	.7073	.7052	.7088
128.4	.7216	.7214	.7231	.7230	.7250	.7207
256.4	.7211	.7224	.7237	.7247	.7274	.7218
512.4	.7228	.7238	.7255	.7268	.7285	.7232
1024.4	.7232	.7244	.7262	.7275	.7298	.7238
128.6	.7170	.7178	.7194	.7217	.7251	.7187
256.6	.7207	.7217	.7257	.7269	.7313	.7234
512.6	.7214	.7232	.7261	.7291	.7338	.7239
1024.6	.7223	.7243	.7276	.7302	.7353	.7250
128.8	.7124	.7135	.7150	.7166	.7199	.7136
256.8	.7147	.7163	.7201	.7214	.7267	.7179
512.8	.7163	.7190	.7220	.7256	.7306	.7200
1024.8	.7176	.7206	.7239	.7274	.7330	.7215
128.10	.7049	.7056	.7083	.7106	.7115	.7078
256.10	.7082	.7095	.7132	.7158	.7197	.7126
512.10	.7106	.7128	.7163	.7195	.7244	.7143
1024.10	.7121	.7150	.7185	.7223	.7270	.7165
128.12	.6970	.6980	.6996	.7022	.7033	.7009
256.12	.7015	.7026	.7062	.7087	.7129	.7053
512.12	.7045	.7066	.7104	.7133	.7179	.7082
1024.12	.7064	.7089	.7125	.7163	.7209	.7107

(a)

N.K	S_b^-	$S_{q_1}^-$	S_m^-	$S_{q_3}^-$	S_w^-	S_w^{\geq}
128.1	.7103	.7065	.7036	.7022	.6992	.7061
256.1	.7127	.7108	.7083	.7058	.7035	.7092
512.1	.7011	.6970	.6943	.6917	.6895	.6962
1024.1	.7049	.7013	.6981	.6954	.6930	.6998
128.2	.7454	.7419	.7418	.7382	.7386	.7397
256.2	.7316	.7288	.7273	.7257	.7233	.7279
512.2	.7257	.7240	.7219	.7196	.7182	.7220
1024.2	.7233	.7199	.7196	.7175	.7150	.7192
128.4	.7569	.7569	.7595	.7578	.7604	.7544
256.4	.7466	.7502	.7510	.7507	.7544	.7477
512.4	.7413	.7435	.7429	.7448	.7470	.7425
1024.4	.7363	.7366	.7389	.7410	.7439	.7372
128.6	.7569	.7575	.7579	.7589	.7603	.7554
256.6	.7449	.7512	.7524	.7536	.7564	.7507
512.6	.7425	.7420	.7443	.7476	.7528	.7435
1024.6	.7368	.7388	.7407	.7446	.7478	.7394
128.8	.7532	.7521	.7562	.7533	.7560	.7504
256.8	.7433	.7428	.7464	.7483	.7524	.7427
512.8	.7369	.7409	.7407	.7442	.7486	.7410
1024.8	.7308	.7343	.7371	.7395	.7457	.7365
128.10	.7432	.7447	.7480	.7436	.7471	.7448
256.10	.7356	.7344	.7380	.7420	.7453	.7373
512.10	.7317	.7315	.7348	.7377	.7425	.7337
1024.10	.7266	.7307	.7335	.7348	.7401	.7312
128.12	.7370	.7350	.7373	.7389	.7400	.7392
256.12	.7314	.7280	.7320	.7348	.7375	.7300
512.12	.7255	.7261	.7293	.7321	.7360	.7271
1024.12	.7228	.7222	.7264	.7307	.7338	.7238

(b)

TABLE 20 – Comparaison des règles pivot S_i^- . (a) Chaque valeur représente les fitness moyennes des optimums locaux atteints par 1000 escalades réparties sur 10 instances de paysages. (b) Valeurs de fitness moyennes des meilleurs optimums locaux enregistrés pour chaque instance de paysage (soit 10 instances par paramétrage (N, K)). Pour rappel, S_b^- est le meilleur améliorant, S_w^- le moins bon améliorant et S_w^{\geq} , indiqué à titre comparatif, est le premier améliorant.

Sur la table 20.b, on s'intéresse aux meilleurs optimums locaux obtenus par chaque stratégie sur chaque paysage, afin d'offrir un mode de comparaison analogue à celui que l'on appliquerait à des recherches locales itérées. Pour chaque couple (stratégie, instance), nous enregistrons le meilleur optimum local atteint parmi les 100 exécutions. Étant donné un paramétrage (N, K) , les valeurs reportées correspondent aux fitness moyennes des meilleurs optimums locaux atteints par chaque stratégie sur les 10 instances de paysages. Globalement, même avec cette règle de comparaison, les observations de la table 20 restent valables. Nous pouvons ainsi nous attendre à ce que la stratégie du moins bon améliorant reste efficace dans un contexte de recherche locale itérée.

Le graphique de la figure 31 reporte les valeurs de fitness moyennes obtenues par les 12 climbers étudiés (pour $N = 1024$). Les méthodes sont ordonnées suivant le rang moyen des solutions sélectionnées par la règle pivot. Pour chaque niveau de rugosité (K), on observe une évolution monotone ou quasi-monotone des valeurs de fitness finales moyennes. Ceci confirme que l'efficacité des recherches locales est ici directement corrélée (négativement ou positivement) à l'amélioration de fitness réalisée à chaque pas de la recherche. Le tableau

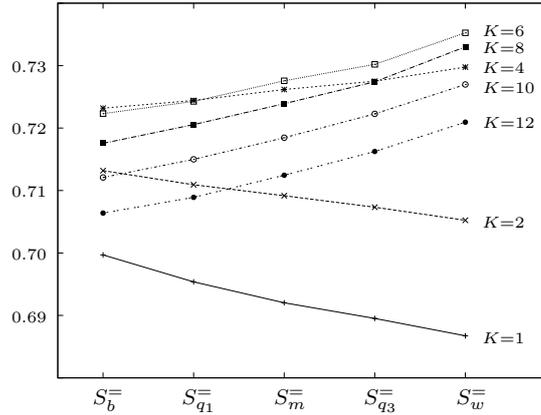


FIGURE 30 – Performances des règles pivot S_i^- en fonction du niveau de rugosité des paysages, pour $N = 1024$. Une tendance similaire est observée pour les autres valeurs de N (voir valeurs sur la table 20).

inclus à cette figure 31 reporte les valeurs numériques pour $K = 12$. Elles permettent de visualiser l'évolution monotone des résultats pour chaque règle pivot générique définie au début de la section ($S^=$, $S^>$, and $S^<$).

Les différents résultats obtenus et reportés dans la table 20 et la figure 31 montrent un net intérêt à réaliser des recherches locales basées sur le moins bon améliorant afin d'aborder des paysages de recherche plus rugueux. Cependant, cette stratégie requiert naturellement davantage d'évaluations et donc, dans un contexte pratique de résolution de problèmes, des temps de calcul bien plus importants. Lorsque l'on compare le meilleur améliorant avec le moins bon améliorant, atteindre un optimum local requiert bien moins de mouvements, puisqu'un pas selon la règle du meilleur améliorant peut améliorer significativement la valeur de fitness, tandis que le moins bon améliorant minimise le bénéfice en termes de fitness à chaque pas de recherche. Quant au premier améliorant, il permet d'éviter la génération systématique du voisinage complet, ce qui réduit drastiquement le nombre d'évaluations nécessaire pour atteindre un optimum local. La section suivante se concentre sur ce paramètre du nombre d'évaluations, et envisage une solution pour garder les avantages du moins bon améliorant en réduisant le nombre d'évaluations des solutions.

7.2.2 Approximer le moins bon améliorant

Dans la section précédente, nous avons souligné que pour escalader des paysages NK plus rugueux, un climber devrait sélectionner les voisins améliorants de plus faible fitness. Cependant, cette recherche s'avère coûteuse pour deux raisons. Tout d'abord, en minimisant les améliorations réalisées à chaque pas, on tend à accroître grandement le nombre de pas nécessaires pour atteindre un optimum local. Ensuite, comme pour le meilleur améliorant, l'intégralité du voisinage doit être générée pour assurer la sélection du voisin améliorant le moins haut — à moins qu'une évaluation incrémentale du voisinage ne soit réalisable.

Ici, nous proposons d'approximer la stratégie du moins bon améliorant en générant le

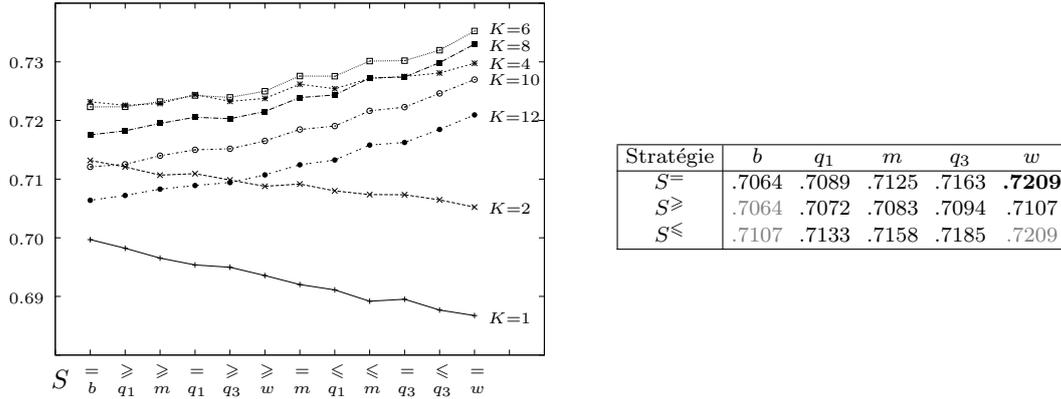


FIGURE 31 – Performance des 12 règles pivot basées sur le rang des voisins, sur 7 ensembles de paysages NK groupés par rugosité (K variant de 1 à 12, avec $N=1024$ uniquement). Les stratégies sont ordonnées selon les valeurs de rang moyennes $\frac{\text{moy_rang_voisins_selectionnes}-1}{\text{nombre_voisins_ameliorants}-1}$ (soit respectivement, de $S_B^=$ à $S_W^=$: 0, 1/8, 1/4, 1/4, 3/8, 1/2, 1/2, 5/8, 3/4, 3/4, 7/8, 1). Le tableau (à droite) contient les valeurs numériques pour $K=12$.

voisinage partiellement. Un tel mécanisme est communément utilisé pour approximer le meilleur améliorant sur des voisinages larges [Mladenović et Hansen, 1997]. Nous introduisons ainsi le *moins bon améliorant partiel* (appelé κ -WORST et noté génériquement \mathcal{W}_κ), qui consiste à stopper la génération du voisinage après que κ voisins améliorants ont été trouvés — à moins que la totalité du voisinage ait été évaluée. À chaque pas de recherche, \mathcal{W}_κ sélectionne le point de plus faible fitness parmi (au plus) κ voisins améliorants (voir algorithme 8). En conséquence, le paramètre κ affecte directement le nombre de points évalués nécessaires pour réaliser un pas de recherche. On peut remarquer que \mathcal{W}_1 équivaut au premier améliorant, tandis que \mathcal{W}_N correspond au moins bon améliorant complet. Notons que la sélection d'un voisin selon la règle \mathcal{W}_κ n'est pas déterministe dès lors que le voisinage comporte plus de κ voisins strictement améliorants.

Pour évaluer l'efficacité d'une telle stratégie, nous avons testé plusieurs versions du moins bon améliorant partiel : \mathcal{W}_2 , \mathcal{W}_4 , \mathcal{W}_8 , \mathcal{W}_{16} , ainsi que \mathcal{W}_1 et \mathcal{W}_N pour comparaison. L'expérimentation consiste à mettre en compétition ces 6 climbers sur l'ensemble de 280 paysages NK utilisé dans la section précédente. Le protocole reste inchangé en ce qui concerne les points de départ, le nombre d'exécutions et le critère d'arrêt. Précisons que les voisins sont toujours générés dans un ordre aléatoire. La table 21 reporte les résultats obtenus par ces climbers sur les paysages dits rugueux ($K \geq 4$), c'est-à-dire ceux où le moins bon améliorant domine l'ensemble des autres stratégies présentées dans la section précédente. Les fitness moyennes des optimums atteints et les dominances statistiques fournies sont calculées de manière analogue à la précédente étude. Nous reportons en outre, pour chaque classe d'instance (paramétrage (N, K)), le nombre moyen d'évaluations nécessaires pour atteindre un optimum local (en milliers).

On rappelle que \mathcal{W}_1 et \mathcal{W}_N dénotent respectivement les $S_w^=$ et S_w^{\geq} de la table 20 (premier et moins bon améliorant) ; on retrouve ainsi les mêmes résultats pour ces méthodes dans les deux tables, montrant que le moins bon améliorant domine le premier améliorant sur tous

Algorithme 8 : Algorithme κ -WORST.

Données : un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$, un point de départ $x_0 \in \mathcal{X}$, un paramètre d'exploration partielle du voisinage κ .

Résultat : un optimum local x_{opt} et sa fitness F_{opt} .

```
1  $x \leftarrow x_0$  ;
2  $F \leftarrow f(x_0)$  ;
3 bloque  $\leftarrow$  faux ;
4 répéter
5    $N \leftarrow \mathcal{N}(x)$  ;
6    $F_w \leftarrow +\infty$  ;
7    $k \leftarrow 0$  ;
8   tant que  $k < \kappa$  et  $N \neq \emptyset$  faire
9     Sélectionner aléatoirement  $x' \in N$  ;
10     $F' \leftarrow f(x')$  ;
11    si  $F' > F$  alors
12       $k \leftarrow k + 1$  ;
13      si  $F' < F_w$  alors
14         $x_w \leftarrow x'$  ;
15         $F_w \leftarrow F'$ 
16      fin
17    fin
18     $N \leftarrow N \setminus \{x'\}$  ;
19  fin
20  si  $k > 0$  alors
21     $x \leftarrow x_w$  ;
22     $F \leftarrow F_w$ 
23  sinon bloque  $\leftarrow$  vrai
24 jusqu'à bloque = vrai ;
25  $x_{\text{opt}} \leftarrow x$  ;
26  $F_{\text{opt}} \leftarrow F$  ;
27 retourner  $(x_{\text{opt}}, F_{\text{opt}})$ 
```

K	N	BEST	FIRST	κ -WORST				WORST
		-	\mathcal{W}_1	\mathcal{W}_2	\mathcal{W}_4	\mathcal{W}_8	\mathcal{W}_{16}	\mathcal{W}_N
4	128	.7216 5k	.7207 1k	.7224 2k	.7237 4k	.7234 8k	.7243 16k	.7250 31k
	256	.7211 19k	.7218 2k	.7233 5k	.7254 10k	.7257 21k	.7262 41k	.7274 136k
		512	.7228 76k	.7232 5k	.7248 11k	.7262 23k	.7278 49k	.7281 97k
	1024		.7232 302k	.7238 12k	.7253 25k	.7270 54k	.7286 112k	.7291 223k
6	128	.7170 4k	.7187 1k	.7214 2k	.7228 5k	.7242 10k	.7240 21k	.7251 46k
	256	.7207 15k	.7234 2k	.7264 5k	.7285 12k	.7301 26k	.7309 56k	.7313 221k
		512	.7214 63k	.7239 5k	.7272 12k	.7298 28k	.7315 63k	.7328 135k
	1024		.7223 251k	.7250 13k	.7280 29k	.7310 65k	.7330 145k	.7343 311k
8	128	.7124 3k	.7136 1k	.7168 2k	.7194 5k	.7197 11k	.7192 24k	.7199 56k
	256	.7147 13k	.7179 2k	.7218 5k	.7243 13k	.7259 30k	.7267 66k	.7267 284k
		512	.7163 53k	.7200 5k	.7242 13k	.7266 31k	.7285 73k	.7297 163k
	1024		.7176 214k	.7215 13k	.7251 31k	.7285 74k	.7306 172k	.7316 382k
10	128	.7049 3k	.7078 1k	.7103 2k	.7122 5k	.7116 11k	.7127 24k	.7115 59k
	256	.7082 11k	.7126 2k	.7159 5k	.7176 13k	.7196 32k	.7201 71k	.7198 325k
		512	.7106 46k	.7143 5k	.7190 13k	.7212 33k	.7232 81k	.7237 183k
	1024		.7121 187k	.7165 14k	.7204 33k	.7235 80k	.7255 189k	.7265 439k
12	128	.6970 2k	.7009 1k	.7021 2k	.7035 5k	.7037 11k	.7028 24k	.7033 59k
	256	.7015 10k	.7053 2k	.7089 5k	.7105 13k	.7124 33k	.7122 74k	.7129 346k
		512	.7045 41k	.7082 5k	.7127 13k	.7151 34k	.7168 85k	.7179 199k
	1024		.7064 166k	.7107 14k	.7150 34k	.7178 84k	.7197 206k	.7206 487k

TABLE 21 – Comparaison des règles pivot du premier améliorant, moins bon améliorant et moins bon améliorant partiel. Sont indiquées les fitness moyennes des optimums locaux atteints depuis 1000 escalades de paysages distribués sur 10 instances différentes pour chaque paramétrage (N, K) .

ces paysages NK où $K \geq 4$. Globalement, selon la figure 32 qui résume graphiquement les résultats de la table 21, on observe que $\mathcal{W}_N \succ \mathcal{W}_{16} \succ \mathcal{W}_8 \succ \mathcal{W}_4 \succ \mathcal{W}_2 \succ \mathcal{W}_1$, avec $A \succ B$ signifiant que le climber A monte en moyenne plus haut que B sur un ensemble particulier de paysages. Cela confirme la tendance générale observée dans la précédente étude sur les règles pivot basées sur le rang, à savoir que lorsque le premier améliorant monte plus haut que le meilleur améliorant, c'est plus généralement que l'efficacité du climber est inversement proportionnelle à la qualité des voisins améliorants sélectionnés à chaque pas de recherche. Néanmoins, sur la plupart des instances, une relativement faible valeur de κ (par rapport à N) est suffisante pour obtenir un moins bon améliorant partiel dont les résultats ne sont pas statistiquement dominés par un moins bon améliorant complet. Il est intéressant de noter que la trajectoire de ces recherches se détermine principalement durant les premiers pas. Le nombre de voisins améliorants diminue en toute logique assez rapidement et hormis pour les plus petites valeurs de κ , les différentes variantes de \mathcal{W}_κ deviennent, en milieu ou en fin d'escalade, équivalentes.

Parallèlement, une autre technique d'approximation a été envisagée, consistant à sélectionner le moins bon améliorant d'un sous-ensemble du voisinage, plutôt que d'un sous-ensemble de voisins améliorants. Cette stratégie revient à évaluer aléatoirement un nombre prédéfini de voisins, puis de sélectionner le moins bon voisin améliorant ; si aucun de ces voisins n'est améliorant, alors on sélectionne le premier améliorant des voisins restants. Les tests, réalisés en choisissant différentes proportions de la taille du voisinage à explorer, ont conduit à des recherches plus longues et pas plus efficaces. À l'inverse de l'algorithme du moins bon améliorant partiel détaillé précédemment, cette stratégie rend l'approximation du moins bon améliorant plus précise dans les premières étapes de la recherche, lorsque le nombre d'améliorants est significatif, mais équivaut plus rapidement à un premier améliorant, lorsque les voisins améliorants se raréfient — là où \mathcal{W}_κ équivaut alors au moins bon améliorant complet. Une hybridation des deux techniques a également été expérimentée, n'offrant pas d'amélioration significative de l'efficacité de notre algorithme du moins bon améliorant partiel, qui a en outre l'avantage de la simplicité et de ne requérir qu'un unique paramètre. Pour davantage de clarté, nous avons choisi de ne pas reporter dans ce manuscrit les résultats numériques de cette étude.

Nombre d'évaluations

L'utilisation de cette stratégie du moins bon améliorant partiel est particulièrement intéressante puisqu'elle requiert nettement moins d'évaluations que la version complète du moins bon améliorant. Dans la plupart des cas, avec une valeur adéquate du paramètre κ , un moins bon améliorant partiel peut statistiquement atteindre des optimums locaux comparables en évaluant bien moins de points qu'un moins bon améliorant complet. Paramétrer κ consiste alors à déterminer le meilleur compromis entre effort (computationnel) et efficacité. Malgré le fait que les stratégies du meilleur et du moins bon améliorant nécessitent toutes deux l'évaluation intégrale du voisinage à chaque pas de recherche, les résultats expérimentaux confirment que le nombre d'évaluations engendrées par une exécution du moins bon améliorant est très sensiblement plus important qu'avec un meilleur améliorant, puisque le nombre de (petits) pas augmente nécessairement. La

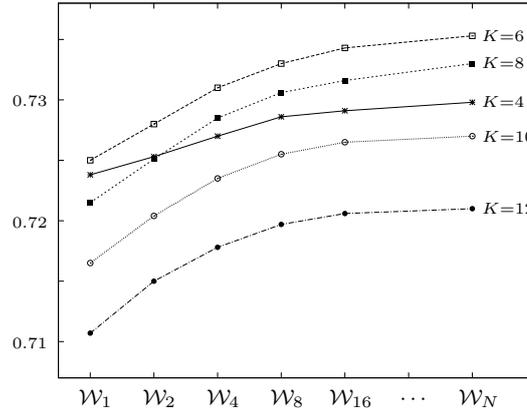


FIGURE 32 – Fitness moyennes atteintes par des climbers de type moins bon améliorant partiel, (\mathcal{W}_κ , avec $\kappa \in \{1, 2, 4, 8, 16, 1024\}$) sur des paysages NK larges et non lisses $N = 1024$ ($K \in \{4, 6, 8, 10, 12\}$).

différence computationnelle s’observe plus particulièrement durant la première partie de la recherche, où les voisins améliorants sont nombreux.

Sur la figure 33.a, nous reportons l’évolution de la fitness moyenne atteinte en fonction du nombre de points évalués sur une instance de paysage 1024_6. Cette figure illustre à quel point le moins bon améliorant est très couteux en comparaison des autres stratégies. On y observe que, même si la vitesse de convergence du moins bon améliorant est faible, elle tend à s’accroître au cours de la recherche³². Une évolution similaire a été observée pour toutes les instances.

La figure 33.b reporte les mêmes résultats, avec un focus sur les 400 000 premières évaluations, suffisantes pour que tous les climbers de type meilleur améliorant et moins bon améliorant partiel atteignent un optimum local. Cette figure permet de visualiser, sur les paysages NK non lisses, la nette supériorité de \mathcal{W}_1 (premier améliorant), \mathcal{W}_2 , \mathcal{W}_4 , \mathcal{W}_8 et \mathcal{W}_{16} , par rapport au meilleur améliorant. De plus, le compromis entre vitesse de convergence et qualité des optimums locaux est aisé à observer. Augmenter κ permet d’atteindre de meilleurs optimums locaux, mais moyennant un nombre d’évaluations plus élevé. En particulier, fixer κ à 2 suffit pour améliorer les résultats obtenus par le premier améliorant ($\kappa = 1$) sans augmenter drastiquement le nombre de points évalués. Lorsque $\kappa = 16$, la vitesse de convergence observée est similaire à celle du meilleur améliorant, mais les recherches sont piégées bien plus tard dans des optimums locaux.

7.2.3 Application à des problèmes combinatoires académiques

Les études empiriques présentées dans ce chapitre et le suivant, qui visent à comparer l’efficacité des climbers sur différents types de paysages et avec un maximum de fiabilité,

³²Pour chaque exécution du moins bon améliorant, la vitesse de convergence augmente durant l’intégralité de la recherche, et spécifiquement durant les derniers pas. Cependant, sur la figure 33.a, le moins bon améliorant semble ralentir après $4 \cdot 10^6$ évaluations en moyenne. Ceci s’explique simplement par le fait que les 100 exécutions atteignent un optimum local entre $4 \cdot 10^6$ et $5 \cdot 10^6$, mais pas simultanément. Ce phénomène ne doit pas être interprété comme un ralentissement de la convergence de chaque climber.

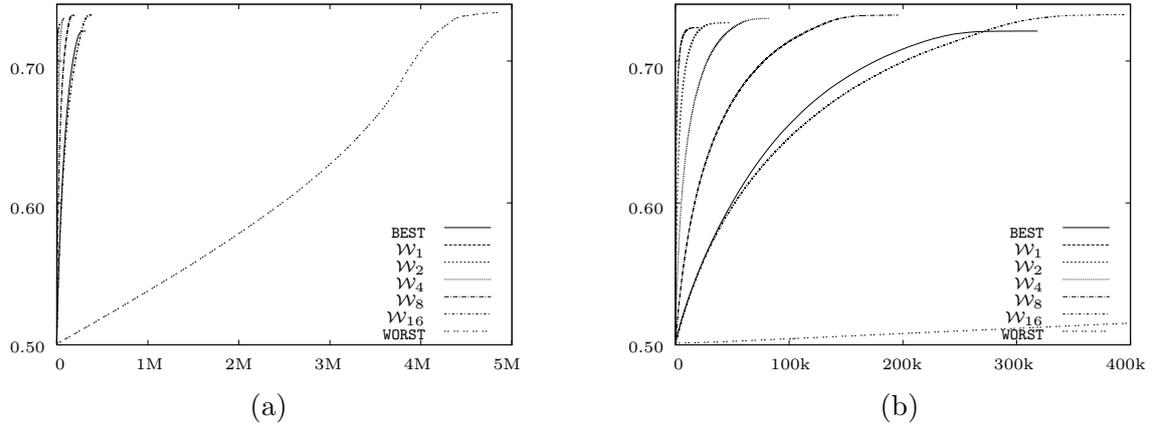


FIGURE 33 – (a) Convergence des climbers étudiés sur un paysage de fitness 1024_6 (nombre d’évaluations en abscisse, fitness moyenne atteinte en ordonnée). (b) Restriction sur les 400 000 premières évaluations.

nécessitent de très longues expérimentations. Au total, ce sont des dizaines de millions d’escalades de paysages qui ont été exécutées sur le cluster du LERIA au cours des deux dernières années. Concernant nos investigations sur la règle du moins bon améliorant, si les résultats obtenus sur les paysages NK ont été publiés [Basseur et Goëffon, 2014], des travaux complémentaires sont en cours, à des niveaux plus ou moins grands de finalisation. Naturellement, ces derniers se mènent au gré du rythme du cluster, dont les ressources sont partagées.

Nous possédons néanmoins quelques résultats préliminaires sur l’efficacité du moins bon améliorant appliqué à l’escalade de paysages dérivés de problèmes combinatoires académiques, que nous pouvons évoquer afin d’appuyer notre propos. Ceux-ci confirment les hypothèses nées de l’étude de ces climbers sur les paysages NK. En particulier, sur le problème du flowshop (paysages FSP_{ins}) où le premier améliorant est nettement la règle pivot la plus efficace parmi les deux variantes traditionnellement utilisées en recherche locale, il apparaît que la stratégie du moins bon améliorant partiel — qui prend particulièrement son sens ici par rapport au moins bon améliorant complet, le voisinage étant en $O(n^2)$ — est plus performante que le premier améliorant sur la plupart des instances, et très sensiblement sur les instances larges. Analysés instance par instance, les résultats sont moins nets que ceux obtenus sur les paysages NK. Une première explication se trouve dans la rugosité des paysages NK standards, plus uniforme et réglée par les paramètres de génération des instances. En outre, les paysages FSP_{ins} comportent un taux de neutralité important et il est pertinent d’associer une politique de sélection des mouvements neutres à ces règles pivot pour mieux évaluer la dynamique de la recherche (voir chapitre suivant). Nous notons cependant, en moyennant l’ensemble des résultats obtenus par les six variantes, que l’efficacité des climbers est également ici inversement proportionnelle à l’importance des pas effectués en termes de variation de fitness (plus petits en moyenne lorsque κ augmente) ; l’observation se vérifie particulièrement pour les paysages les plus difficiles (nombre élevé de jobs et de machines).

Instance	FIRST	\mathcal{W}_2	\mathcal{W}_4	\mathcal{W}_8	\mathcal{W}_{16}
20_5_01.ta001	-1 306.7	-1 304.0	-1 302.1	-1 301.8	-1 305.3
20_10_01.ta011	-1 629.3	-1 630.0	-1 633.3	-1 632.4	-1 629.5
20_15_01	-1 961.4	-1 963.8	-1 964.4	-1 966.4	-1 964.4
20_20_01.ta021	-2 374.7	-2 367.8	-2 361.1	-2 366.2	-2 367.5
30_5_01	-1 736.8	-1 737.2	-1 734.2	-1 738.1	-1 735.3
30_10_01	-2 072.2	-2 069.3	-2 066.7	-2 064.6	-2 063.6
30_15_01	-2 510.7	-2 498.6	-2 495.9	-2 492.7	-2 489.1
30_20_01	-2 856.6	-2 846.8	-2 853.7	-2 845.6	-2 840.7
50_5_01.ta031	-2 743.3	-2 742.8	-2 743.5	-2 741.7	-2 741.2
50_10_01.ta041	-3 140.2	-3 134.0	-3 128.9	-3 130.9	-3 127.3
50_15_01	-3 535.6	-3 524.9	-3 520.6	-3 506.9	-3 506.0
50_20_01.ta051	-4 033.7	-4 028.4	-4 016.5	-4 015.9	-4 010.2
70_5_01	-3 794.4	-3 794.4	-3 793.9	-3 794.6	-3 794.1
70_10_01	-4 168.0	-4 149.6	-4 151.6	-4 151.1	-4 145.4
70_15_01	-4 566.6	-4 556.8	-4 544.7	-4 535.6	-4 526.3
70_20_01	-4 980.5	-4 957.4	-4 952.1	-4 937.7	-4 927.5
100_5_01.ta061	-5 510.0	-5 508.7	-5 510.0	-5 507.3	-5 508.8
100_10_01.ta071	-5 864.5	-5 872.0	-5 853.6	-5 854.0	-5 850.3
100_15_01	-6 155.3	-6 136.1	-6 127.5	-6 113.6	-6 117.7
100_20_01.ta081	-6 560.5	-6 552.8	-6 519.0	-6 520.1	-6 498.7
150_5_01	-8 066.3	-8 064.8	-8 065.3	-8 065.2	-8 065.0
150_10_01	-8 240.6	-8 226.7	-8 213.3	-8 216.1	-8 205.0
150_15_01	-8 676.8	-8 669.4	-8 657.4	-8 654.8	-8 648.4
150_20_01	-9 327.7	-9 288.7	-9 265.2	-9 250.0	-9 237.8
200_10_01.ta091	-11 006.3	-10 990.0	-10 986.3	-10 985.1	-10 980.0
200_15_01	-11 400.0	-11 386.7	-11 365.6	-11 364.5	-11 354.5
200_20_01.ta101	-11 650.7	-11 634.5	-11 594.2	-11 584.8	-11 580.0
Moyenne	-5 180.4	-5 171.7	-5 163.7	-5 160.7	-5 156.3

TABLE 22 – Résultats des climbers de type premier améliorant partiel sur les paysages FSP_{ins} , avec $\kappa \in \{2, 4, 6, 8\}$. Les résultats obtenus par FIRST (voir table 18) sont reportés pour comparaison.

Instance	$ S $	Climber	Meilleur opt.	Moyenne	Temps de calcul
rand200-5	200	WORST	3 925	3 726.0	0.31
		WORST \mathcal{W}_{16}	3 947	3 720.5	0.07
		WORST \mathcal{W}_4	3 908	3 675.6	0.02
		FIRST	3 906	3 655.5	0.01
		BEST	3 747	3 527.5	0.01
p500-100-1	500	WORST	300 710	292 418.2	101.46
		WORST \mathcal{W}_{16}	297 740	287 373.0	1.21
		WORST \mathcal{W}_4	294 682	279 912.7	0.20
		FIRST	289 550	278 692.8	0.05
		BEST	280 768	265 873.8	0.09
p2000-1	2000	WORST	2 366 885	2 358 156.7	11 085.34
		WORST \mathcal{W}_{16}	2 329 199	2 319 762.7	22.60
		WORST \mathcal{W}_4	2 300 701	2 284 389.5	3.13
		FIRST	2 332 360	2 293 806.5	1.08
		BEST	2 197 345	2 163 574.5	2.00

TABLE 23 – Comparaison de climbers sur trois instances du problème de clique partition.

Nous avons également profité de travaux annexes sur la résolution du problème de maximisation *clique partition* pour effectuer une étude similaire de comparaison des stratégies de climbers. L'intérêt est qu'il s'agit d'un problème académique défini sur un autre type d'espace de recherche. Le problème de clique partition consiste à partitionner les sommets d'un graphe complet non orienté $(S, \{s_i, s_j\}_{s_i, s_j \in S})$, dont chaque arête $\{s_i, s_j\}$ est étiquetée par une valeur v_{ij} positive ou négative, en un nombre non fixé de k ensembles disjoints V_1, V_2, \dots, V_k , tels que la somme des valeurs des arêtes conservées (dit *score*) soit maximale.

La comparaison des performances de cinq climbers — moins bon améliorant, moins bon améliorant partiel (deux versions, avec $\kappa = 16$ et $\kappa = 4$), premier améliorant, meilleur améliorant — s'est effectuée sur les instances utilisées par Palubeckis *et al.* [2014]. Chaque climber a été exécuté 100 fois sur chaque instance. Nous reportons sur la table 23, pour trois instances représentatives de tailles diverses, la moyenne des optimums et le meilleur optimum atteint, ainsi que le temps de calcul moyen d'une recherche locale.

On constate ici que la stratégie du moins bon améliorant permet d'atteindre de meilleurs optimums mais au prix de temps de calcul très importants. Les variantes partielles permettent de baisser drastiquement le temps de calcul de l'algorithme WORST tout en atteignant, en moyenne, de meilleures solutions que FIRST et BEST. Il est intéressant de noter que sur ce problème, les algorithmes de référence emploient une recherche tabou [De Amorim *et al.*, 1992; Brusco et Köhn, 2009; Palubeckis *et al.*, 2014] utilisant la règle pivot du meilleur améliorant, alors que cette règle conduit clairement vers des optimums moins bons. Nous avons alors conçu notre propre algorithme basé sur la recherche tabou mais utilisant une règle du moins bon améliorant partiel; ceci nous a permis d'égaliser ou d'améliorer l'intégralité des meilleures solutions connues pour les instances disponibles dans la littérature sur ce problème, en des temps de calcul réduits :

- meilleurs scores égalés sur les 7 instances de Charon et Hudry [2006], et sur les 6 instances de Brusco et Köhn [2009];
- meilleurs scores battus (17 instances) ou égalés (18) sur les 35 nouvelles instances aléatoires de Palubeckis *et al.* [2014].

L’algorithme développé utilise la stratégie du moins bon améliorant partiel ($\kappa = 4$) mais hybridée à une recherche tabou. Sur les instances les plus difficiles où de nouvelles bornes ont été découvertes, l’algorithme est plus performant que sa variante employant la stratégie du meilleur améliorant, tout fonctionnement égal par ailleurs. L’algorithme en question et les résultats ne seront pas détaillés dans ce manuscrit car ils sortent de notre étude sur les paysages de fitness et les mécanismes d’intensification, mais ils permettent de montrer à quel point un changement de stratégie concernant la règle pivot peut permettre d’améliorer l’efficacité des métaheuristiques plus sophistiquées. Il en est de même pour les politiques de mouvements neutres, comme nous le verrons dans le prochain chapitre.

7.2.4 Discussion

Ces travaux sur les règles pivot n’ont pas pour unique vocation de déterminer, en fonction d’indicateurs rendant compte des propriétés d’un paysage (ou d’une zone de paysage), les stratégies d’escalade les plus efficaces en vue d’une sélection de type *portfolio* [Leyton-Brown *et al.*, 2003]; il s’agit cependant déjà d’une application intéressante. Ils permettent en outre, de par la clarté des résultats expérimentaux, de mieux comprendre la dynamique des méthodes de recherche locale. Ainsi, si la stratégie du premier améliorant domine celle du meilleur améliorant sur bon nombre d’instances, ce n’est pas par incorporation d’un mécanisme aléatoire — bien que l’intérêt d’une recherche non déterministe dans un contexte itéré ne se discute pas —, mais bien parce que les pas les plus améliorants peuvent pénaliser la recherche en l’attirant vers les optimums locaux les plus proches alors qu’un oracle les contournerait. Ce constat découle principalement des résultats très réguliers et monotones obtenus par l’ensemble des règles pivots intermédiaires introduites dans cette section et observés sur les paysages NK, dont les caractéristiques sont paramétrables. Il apparaît que l’efficacité des climbers est en effet directement corrélée à la qualité des voisins améliorants choisis à chaque itération. En particulier, sur les paysages NK dits rugueux, où le premier améliorant domine le meilleur améliorant, l’efficacité des climbers est inversement proportionnelle à l’amélioration de fitness des voisins améliorants sélectionnés, avec ainsi un avantage final à la stratégie du moins bon améliorant.

En complément des premiers résultats d’Ochoa *et al.* [2010], qui montrent que la taille des bassins d’attraction des optimums locaux est généralement, sur des problèmes réguliers, proportionnelle à leur qualité, on peut imaginer qu’une pente plus ardue mènera vers un optimum local plus proche et moins bon, tandis que les pentes menant aux meilleurs optimums locaux seraient moins raides. Lorsque la rugosité est suffisamment élevée, il existe de nombreux optimums locaux proches de la trajectoire menant aux plus hauts optimums. On imagine alors que les pentes les plus hautes mènent plus probablement vers ces optimums de moins bonne qualité.

Gardons à l’esprit qu’en revanche, des paysages très lisses ou au contraire extrêmement rugueux, conduisent à des effets inverses. Concernant les paysages lisses, le nombre réduit d’optimums locaux diminue la nécessité de chercher à les contourner. De plus, avec une épistasie réduite, il est vraisemblable que les pentes successivement les plus ardues mènent conjointement à de plus hauts optimums locaux. Les paysages très rugueux, en revanche, sont peu structurés; c’est le cas par exemple d’une relation de voisinage qui n’établit pas suffisamment les proximités sémantiques des points (solutions), à supposer qu’il en existe.

Dans ce contexte, d’ailleurs, l’utilité d’y appliquer un algorithme de recherche locale ne semble pas réellement pertinente. Sur ces paysages, la proportion de points étant des optimums locaux est bien plus importante. Cela signifie que les recherches locales strictes y sont très courtes ; quelques pas peuvent suffire avant de rencontrer un optimum local. Logiquement, la probabilité d’atteindre de plus hauts optimums est accrue en maximisant, à chaque itération, la fitness du prochain point visité — celui-ci ayant une certaine chance d’être le dernier.

Contrairement à Ochoa *et al.* [2010] qui ont pu vérifier cet aspect de petits paysages NK ($N = 16$), il ne fut pas envisageable dans le cadre de notre étude sur les paysages larges de générer des paysages NK extrêmement rugueux (jusqu’à $K = N - 1$). En effet, on rappelle qu’une matrice de fitness-contributions d’une instance NK possède $2^{K+1} \times N$ valeurs et ces instances deviennent impossibles à construire³³ pour de plus grandes valeurs de K . Nous avons néanmoins simulé le comportement des climbers sur des paysages à fitness aléatoires en générant simplement, à chaque pas, N valeurs de fitness aléatoires, puis en remplaçant la fitness courante par le maximum (**BEST**), un majorant aléatoire (**FIRST**) et le plus petit majorant (**WORST**) jusqu’à ce que les N fitness générées soient inférieures à la fitness courante, symbolisant ainsi la présence d’un optimum local. Bien sûr, il ne s’agit pas exactement de la génération d’un paysage de fitness aléatoire, puisque le paysage est localement régénéré à chaque pas, mais cela nous permet d’approcher le comportement des climbers sur des paysages NK tels que $K = N - 1$, quelle que soit la valeur de N . Pour que les fitness des optimums locaux atteints soient du même ordre de grandeur que celles des paysages NK (à rugosité maximale), les fitness sont établies à partir de la moyenne de N nombres aléatoires compris dans $[0, 1[$. Avec des fitness prenant leur valeur uniformément dans l’intervalle $[0, 1[$, les optimums observent des fitness tous très proches de 1. La table 24 reporte la moyenne des optimums atteints sur des paysages à fitness aléatoires (1000 exécutions par climber et par valeur de N), simulés selon la méthode décrite ci-dessus. Sur ces paysages extrêmement rugueux, où il n’y a plus aucune corrélation entre la relation de voisinage et la fonction de fitness, on peut vérifier que l’efficacité comparée des climbers s’est à nouveau inversée par rapport aux paysages rugueux mais structurés. On retrouve alors la hiérarchie des règles pivot correspondant aux paysages dits lisses : **BEST** atteint de plus hautes valeurs que **FIRST**, qui lui-même domine **WORST**.

N	128	256	512	1024
BEST	.5746	.5568	.5426	.5319
FIRST	.5707	.5537	.5406	.5306
WORST	.5669	.5508	.5390	.5294

TABLE 24 – Fitness moyennes obtenues par escalades de paysages à fitness aléatoires simulés.

7.3 Vers d’autres règles pivot

Alors que le chapitre 6 nous confortait dans l’intuition que les meilleurs optimums locaux sont accessibles par des climbers depuis la plupart des points de départ, les études

³³Dans des travaux en cours, nous développons un modèle alternatif des paysages NK, permettant une représentation polynomiale des instances. Manukyan *et al.* [2014] ont récemment proposé le modèle des paysages NM qui s’inscrit dans ce sens.

comparatives menées dans ce chapitre montrent que la règle pivot est extrêmement déterminante dans la qualité des optimums atteints. Comme nous l’avons évoqué, un climber est en réalité une marche aléatoire naviguant dans un graphe de transition contraint par la règle pivot. Une règle plus stochastique et donc moins restrictive comme celle du premier améliorant laisse davantage de possibilités, mais ne suit aucune stratégie de recherche particulière. Nous avons vu qu’en fonction des propriétés des paysages, soit le meilleur améliorant, soit le moins bon améliorant, permettait de maximiser l’espérance des fitness atteintes (par rapport au premier améliorant) ; ce qui laisse à penser que des climbers plus efficaces, utilisant des stratégies de recherche plus fines, peuvent être définis. En théorie, il existe une règle pivot d’efficacité maximale. Celle-ci consiste à construire un graphe de transition contraint au nombre d’arcs minimal et tel que le taux de couverture de l’optimum global soit maximal. Mais une telle construction implique la connaissance de l’optimum global et ne peut être effectuée en temps ou en espace polynomial. En pratique, le graphe de transition contraint est défini — via la règle pivot — en intension, et nous cherchons le meilleur compromis entre nombre d’arcs minimum et taux de couverture des plus hauts optimums locaux maximum. En effet, dans un contexte de stratégie intensificatrice, éviter des optimums locaux ne peut s’obtenir qu’en supprimant des arcs du graphe de transition contraint. Or définir en intension des règles supprimant ces arcs entraîne inévitablement la réduction des taux de couverture de l’ensemble des optimums locaux, y compris les plus hauts. En revanche, cela peut augmenter la taille de leur bassin d’attraction et ainsi la probabilité d’être atteints.

Nous avons décrit dans ce chapitre un certain nombre de règles pivot définies de manière à sélectionner un voisin en fonction de la fitness d’un sous-ensemble de voisins du point courant. On dira que ces règles pivot sont de niveau 1, c’est-à-dire qu’elles n’utilisent que l’information relative au voisinage courant. Naturellement, on peut définir des règles pivot de niveau 2 ou supérieur, en utilisant l’information des voisins des voisins, ou d’autres points de l’espace. Ces points non voisins peuvent uniquement servir d’information afin de respecter la condition de se déplacer de point en point voisin à partir d’une structure de voisinage fixée, mais dans un autre contexte pourraient aussi être sélectionnés directement ; ce qui revient dans ce cas à modifier le voisinage et s’apparente davantage à une *recherche à voisinage variable* [Mladenović et Hansen, 1997]. Dans notre cas, une règle pivot conduit nécessairement à sélectionner un point du voisinage courant ; et en se réduisant aux climbers, les règles pivot définies interdisent systématiquement tout mouvement détériorant.

Verel *et al.* [2004] définissent *HC2*, une stratégie d’escalade sélectionnant le voisin de plus haute fitness, ou dont l’un des voisins est de plus haute fitness en envisageant l’ensemble des points à une distance de 2 au maximum. *HC2* est plus efficace que *BEST*, mais ne respecte pas la restriction des mouvements améliorants de voisin en voisin. Dans des travaux en cours, nous avons défini un nouvel ensemble de règles pivot de niveau 2. Parmi elles, la règle du maximum expansion (*ME*) sélectionne à chaque pas le voisin améliorant maximisant le nombre de voisins améliorants. La table 25 reporte les résultats d’une première série d’expérimentations sur les paysages *NK*, avec le processus expérimental utilisé dans la section 7.2.1. Les deux dernières colonnes récapitulent l’amélioration moyenne des fitness atteintes et le nombre d’évaluations nécessaires supplémentaires par rapport à *FIRST*. On y remarque que *ME* permet d’atteindre de bien meilleurs optimums que les règles pivot de niveau 1 étudiées dans ce chapitre. Pourtant, *ME* respecte notre définition de la règle pivot

K	4				6				8			
N	128	256	512	1024	128	256	512	1024	128	512	215	1024
BEST	.7216	.7211	.7228	.7232	.7170	.7207	.7214	.7223	.7124	.7147	.7163	.7176
FIRST	.7207	.7218	.7232	.7238	.7187	.7234	.7239	.7250	.7136	.7179	.7200	.7215
WORST \mathcal{W}_2	.7224	.7233	.7248	.7253	.7214	.7264	.7272	.7280	.7168	.7218	.7242	.7251
WORST	.7250	.7274	.7285	.7298	.7251	.7313	.7338	.7353	.7199	.7267	.7306	.7330
Max. expansion	.7464	.7478	.7481	.7490	.7468	.7444	.7503	.7509	.7415	.7444	.7455	.7464

K	10				12				Δ fitness	nb. éval.
N	128	256	512	1024	128	256	512	1024		
BEST	.7049	.7082	.7106	.7121	.6970	.7015	.7045	.7064	− .0027	× 9.7
FIRST	.7078	.7126	.7143	.7165	.7009	.7053	.7082	.7107	0 (référence)	0 (référence)
WORST \mathcal{W}_2	.7103	.7159	.7190	.7204	.7021	.7089	.7127	.7150	+ .0031	× 2.3
WORST	.7115	.7198	.7243	.7270	.7033	.7129	.7179	.7210	+ .0077	× 213.0
Max. expansion	.7356	.7376	.7386	.7396	.7267	.7300	.7316	.7321	+ .0252	× 6 682.6

TABLE 25 – Comparaison d’un climber basé sur la règle du maximum expansion par rapport aux règles pivot à un niveau.

intensificatrice et tout chemin emprunté par un climber ME est un chemin empruntable par FIRST. Naturellement, le frein principal à l’utilisation de cette stratégie en pratique est l’effort computationnel qu’elle nécessite, bien supérieur encore à celui du moins bon améliorant. Une perspective est d’élaborer des stratégies basées sur le maximum expansion mais plus rapides, comme nous l’avons fait pour la règle du moins bon améliorant. Il faut préciser que cette règle a déjà été utilisée par Alekseeva *et al.* [2008] sous l’appellation *freedom*, dans un article que nous avons découvert récemment et qui de plus est le seul à notre connaissance à considérer expérimentalement la règle du moins bon améliorant. Comme nous l’avons déjà mentionné, les auteurs ont observé sur le problème du p -médián des conclusions similaires aux nôtres sur les paysages NK rugueux en termes d’efficacité et de cout relatifs entre ces stratégies. Un aspect intéressant des travaux de Alekseeva *et al.* [2008] est qu’ils ont pu mesurer l’erreur moyenne à l’optimum de ces climbers sur le problème du p -médián. Leurs résultats indiquaient une erreur induite par l’utilisation de la règle *freedom* deux à trois fois plus faible qu’en employant FIRST ou WORST, et même environ dix fois plus faible que l’erreur générée par BEST.

Même peu avantageuses en pratique du fait de leur plus grande complexité algorithmique, ces règles pivot qui permettent d’atteindre de plus hauts optimums locaux nous aident à mieux comprendre la dynamique des recherches locales dans les paysages de recherche. L’information importante est que même une vision extrêmement locale du paysage à chaque pas — ces règles pivot sont de niveau 1 ou 2 — suffit à emprunter au final une trajectoire évitant les mauvais optimums locaux sans même en avoir détecté aucun. Nous avons choisi dans ce chapitre — comme dans le suivant — de conserver un maximum de résultats expérimentaux car il nous semble que le propos aurait été amoindri avec uniquement un bilan faisant état de nos études empiriques. Dans leur globalité, ces comparaisons entre climbers nous confirment que la stratégie d’intensification d’une métaheuristique doit être pensée avec soin, et qu’elle est clairement déterminante dans la capacité d’atteindre de bonnes solutions approchées. Ces études constituent une première étape vers le développement de stratégies de recherche plus efficaces et, dans le sens où elles pourraient utiliser les propriétés structurelles des paysages, intelligentes. Il sera notamment intéressant de déterminer si la corrélation (positive ou négative) entre fitness des points

sélectionnés et des optimums atteints est directe ou indirecte. Par exemple, les voisins améliorants de plus petite fitness ont probablement plus de voisins améliorants que ceux de plus haute fitness. La règle du moins bon améliorant n'est-elle pas dans ce cas une approximation de la règle du maximum expansion ? Est-ce uniquement le critère d'expansion et non de plus petite variation de fitness qui permet d'éviter certains optimums locaux ? Ces aspects seront débattus dans des travaux futurs.

Chapitre 8

Politiques de mouvements neutres et neutralité artificielle

Ce chapitre vise à déterminer l'apport relatif des différentes politiques d'application des mouvements neutres pour escalader des paysages de fitness comportant de la neutralité. Poursuivant notre propos sur les techniques d'escalade, indépendant de tout mécanisme perturbatif, nous ne cherchons pas ici à déterminer les stratégies les plus appropriées pour sortir des plateaux; tout comme le but du chapitre précédent était d'analyser les mécanismes conduisant aux meilleurs optimums locaux, et non de déterminer les moyens de sortir de ces optimums locaux. Nous estimons ici le bien-fondé de considérer les mouvements neutres tout au long de la recherche au même titre que tout autre voisin améliorant, puis sur cette base d'incorporer artificiellement de la neutralité aux paysages afin de moins contraindre les possibilités de mouvement, et de rendre les escalades plus efficaces.

8.1 Comparaison des politiques de mouvements neutres

Trois différentes stratégies de gestion de la neutralité ont été abordées dans la section 5.6 (climber basique, basique avec perturbations neutres, stochastique), menant à cinq variantes d'escalade lorsque combinées aux deux règles pivot premier et meilleur améliorant. Les climbers basiques **FIRST** et **BEST** ne considèrent à aucun moment l'acceptation de mouvements neutres au cours de la recherche, et leur comparaison sur des paysages variés a fait l'objet de la section 7.1. Comme ils ne peuvent être comparés équitablement avec les variantes autorisant les mouvements neutres, cette comparaison des stratégies d'escalade a été séparée en deux études distinctes. Dans cette section, nous nous intéresserons ainsi aux trois autres variantes : le climber stochastique — un climber de type premier améliorant acceptant à chaque itération un voisin aléatoire de fitness supérieure ou égale — et deux climbers basiques (premier et meilleur améliorant) combinés avec un mécanisme de perturbation neutre. Dans ce cas, des mouvements neutres sont appliqués uniquement à partir d'optimums locaux. Ces trois algorithmes seront respectivement appelés **STOCH** (algorithme 9), **FIRST+NP** et **BEST+NP** (voir algorithme 10 pour les deux variantes). Dans cette section, nous comparons les stratégies dites classiques et n'incluons pas la stratégie du moins bon améliorant, dont l'incorporation au sein d'un climber stochastique sera discutée

Algorithme 9 : Algorithme STOCH.

Données : un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$, un point de départ $x_0 \in \mathcal{X}$, un nombre d'accès à des valeurs de fitness à ne pas dépasser (T).

Résultat : un optimum approché x_{opt} et sa fitness F_{opt} .

```
1  $F \leftarrow f(x_0); t \leftarrow 1;$ 
2  $N \leftarrow \mathcal{N}(x_0);$ 
3 tant que  $N \neq \emptyset$  et  $t < T$  faire
4   | Sélectionner aléatoirement  $x' \in N;$ 
5   |  $F' \leftarrow f(x'); t \leftarrow t + 1;$ 
6   | si  $F' \geq F$  alors
7   |   |  $x \leftarrow x';$ 
8   |   |  $F \leftarrow F';$ 
9   |   |  $N \leftarrow \mathcal{N}(x)$ 
10  | sinon  $N \leftarrow N \setminus \{x'\}$ 
11 fin
12  $x_{\text{opt}} \leftarrow x;$ 
13  $F_{\text{opt}} \leftarrow F;$ 
14 retourner  $(x_{\text{opt}}, F_{\text{opt}})$ 
```

dans la section 8.4.

Les paysages NK utilisés dans le chapitre précédent ne possèdent pas de neutralité. Sur ces paysages non neutres, les climbers **STOCH** et **FIRST+NP** ont un comportement exactement identique à **FIRST**; de même, **BEST+NP** et **BEST** sont équivalents s'ils sont exécutés sur des paysages de fitness ne comportant aucune transition neutre — ou un nombre infime. Les expérimentations seront réalisées sur des variantes neutres des paysages NK utilisés lors de la section 7.1, ainsi que sur l'intégralité des paysages QAP_{swap} , FSP_{ins} et **MAXSAT**. Ce qui représente, au total, 480 paysages :

- **NK_p** : 64 paysages NK résultant des 16 paysages NK utilisés dans la section 7.1 et modifiés chacun au moyen de quatre paramétrages différents ($p \in \{0.50, 0.80, 0.95, 0.99\}$).
- **NK_q** : 64 paysages NK résultant des mêmes 16 paysages NK et modifiés chacun au moyen de quatre paramétrages ($q \in \{2, 3, 5, 10\}$).
- **NK_r** : 64 paysages NK résultant des mêmes 16 paysages NK et modifiés en tronquant la fonction de fitness au moyen de quatre paramétrages ($r \in \{10, 100, 1000, 10000\}$).
- Les 27 paysages FSP_{ins} , 132 paysages QAP_{swap} et 129 paysages **MAXSAT** introduits section 5.5.

Comme ce chapitre se focalise sur les mouvements neutres au sein de paysages neutres, nous reporterons dans la partie expérimentale uniquement l'indicateur de neutralité ν , présenté section 5.3.2, définition 8. Pour chaque paysage, ν est évalué par échantillonnage à partir d'un million de paires de points voisins aléatoires. Afin de mieux apprécier la répartition de la neutralité dans les paysages de fitness et particulièrement autour des optimums locaux, nous comparons table 26 l'indicateur de neutralité par échantillonnage ν calculé sur les paysages NK neutres, à un indicateur de neutralité aux optimums locaux noté ν_{opt} , qui s'obtient en déterminant la moyenne des degrés de neutralité ι des optimums locaux atteints à partir de 10 000 exécutions d'un climber strict **FIRST** sur chacun des

Algorithme 10 : Algorithmes FIRST+NP et BEST+NP.

Données : un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$, un point de départ $x_0 \in \mathcal{X}$, un nombre d'accès à des valeurs de fitness à ne pas dépasser (T).

Résultat : un optimum approché x_{opt} et sa fitness F_{opt} .

```
1  $x \leftarrow x_0$  ;
2  $F \leftarrow f(x)$  ;  $t \leftarrow 1$  ;
3 bloque  $\leftarrow$  faux ;
4 répéter
5    $N \leftarrow \mathcal{N}(x)$  ;
6    $F_b \leftarrow -\infty$  ;
7   tant que  $N \neq \emptyset$  et  $t < T$  [et  $F_b \leq F$ ]† faire
8     Sélectionner aléatoirement  $x' \in N$  ;
9      $F' \leftarrow f(x')$  ;  $t \leftarrow t + 1$  ;
10    si  $F' > F_b$  [et  $F' \leq F$ ]† alors
11       $x_b \leftarrow x'$  ;
12       $F_b \leftarrow F'$ 
13    fin
14     $N \leftarrow N \setminus \{x'\}$ 
15  fin
16  si  $F_b \geq F$  alors
17     $x \leftarrow x_b$  ;
18     $F \leftarrow F_b$ 
19  sinon bloque  $\leftarrow$  vrai
20 jusqu'à bloque = vrai ou  $t = T$  ;
21  $x_{\text{opt}} \leftarrow x$  ;
22  $F_{\text{opt}} \leftarrow F$  ;
23 retourner  $(x_{\text{opt}}, F_{\text{opt}})$ 
```

[†] Algorithme FIRST+NP uniquement.

paysages. On remarque que sur ce type de paysages, les degrés de neutralité sont plus faibles aux optimums, mais d'un ordre de grandeur plus ou moins important en fonction des paramètres des paysages. En outre, seuls les modèles NKp et NKr peuvent produire des paysages avec une forte neutralité. Même avec son paramétrage entraînant le plus de neutralité ($q = 2$), un paysage NKq aléatoire a peu de chance de compter un taux de neutralité supérieur à 0.5. Et lorsque K croît, les voisins neutres se raréfient, en particulier autour des optimums locaux. Le modèle NKp permet en revanche de générer des paysages très neutres, à condition que p soit proche de 1. Dans ce cas, la plupart des fitness-contributions sont nulles, impliquant des fitness de points voisins souvent inchangées. Or comme nous le verrons par la suite, ces instances NKp très neutres n'offrent pas de grande difficulté de résolution puisque les fitness des points sont uniquement déterminées par quelques fitness-contributions non nulles. Enfin, la neutralité des paysages NKr dépend logiquement de N , alors que celle des paysages NKp et NKq ne varie pas avec N . En effet, augmenter N proportionnellement réduit la variation de fitness entre deux voisins — ce qui est induit par l'équation 25 (chapitre 5).

La figure 34 représente la répartition des différents degrés de neutralité au sein des paysages NK. Cela nous permet tout d'abord d'observer qu'au sein d'un paysage NK, les points ont des degrés de neutralité très peu variés. De plus, un fort taux de neutralité n'implique pas l'existence de zones totalement plates. Concernant les paysages NKp, il n'y a presque aucun point ayant un degré de neutralité maximal — même avec $p = 0.99$. En revanche, les paysages NKr (introduits dans [Basseur et Goëffon, 2013]), sur des paramètres spécifiques, peuvent permettre de générer des paysages comportant d'immenses plateaux.

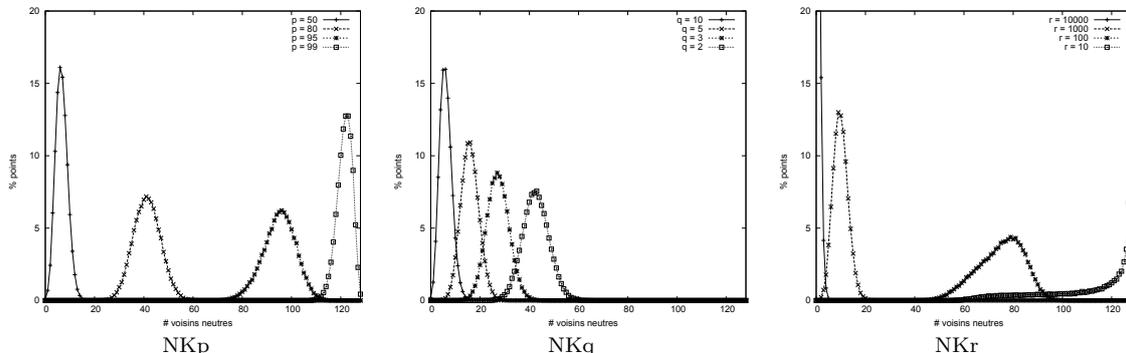


FIGURE 34 – Distribution de 10 000 points aléatoires en fonction de leur nombre de voisins neutres ($N \times \iota$) pour 12 instances de paysages NK neutres ($N = 128$ et $K = 2$).

Les taux de neutralité estimés ν des autres types de paysages considérés sont disponibles dans les tableaux de résultats de la section suivante. Ils sont très variables en fonction des instances et des problèmes, et seront discutés au cours de l'étude comparative des climbers.

NK _p					
<i>N</i>	<i>K</i>	<i>p</i>	ν	ν_{opt}	
128	1	.50	11.2%	6.8%	
		.80	44.6%	35.1%	
		.95	81.8%	75.2%	
		.99	94.9%	92.6%	
	2	.50	5.6%	2.4%	
		.80	30.9%	16.8%	
		.95	73.6%	56.8%	
		.99	94.4%	89.2%	
	4	.50	1.2%	0.3%	
		.80	14.5%	3.4%	
		.95	60.5%	29.1%	
		.99	89.7%	73.5%	
8	.50	0.1%	0.0%		
	.80	3.4%	0.3%		
	.95	41.3%	6.7%		
	.99	83.4%	44.3%		

<i>N</i>	<i>K</i>	<i>p</i>	ν	ν_{opt}	
256	1	.50	11.9%	7.4%	
		.80	45.09%	34.6%	
		.95	81.46%	74.2%	
		.99	96.11%	94.2%	
	2	.50	5.4%	2.3%	
		.80	30.8%	16.8%	
		.95	74.9%	58.8%	
		.99	94.2%	88.9%	
	4	.50	1.3%	0.3%	
		.80	15.2%	3.7%	
		.95	60.9%	29.2%	
		.99	90.8%	75.7%	
8	.50	0.1%	0.0%		
	.80	3.6%	0.2%		
	.95	41.3%	6.7%		
	.99	83.7%	44.7%		

<i>N</i>	<i>K</i>	<i>p</i>	ν	ν_{opt}	
512	1	.50	12.2%	7.6%	
		.80	45.3%	34.9%	
		.95	81.7%	74.8%	
		.99	96.3%	94.6%	
	2	.50	5.4%	2.2%	
		.80	31.2%	16.7%	
		.95	74.3%	58.0%	
		.99	94.6%	89.6%	
	4	.50	1.2%	0.3%	
		.80	15.2%	3.8%	
		.95	61.2%	29.5%	
		.99	90.2%	74.5%	
8	.50	0.1%	0.0%		
	.80	3.5%	0.2%		
	.95	41.4%	6.6%		
	.99	83.6%	44.6%		

<i>N</i>	<i>K</i>	<i>p</i>	ν	ν_{opt}	
1024	1	.50	12.0%	7.8%	
		.80	44.7%	34.2%	
		.95	81.7%	74.5%	
		.99	96.1%	94.2%	
	2	.50	5.6%	2.3%	
		.80	31.5%	16.8%	
		.95	74.3%	58.2%	
		.99	94.2%	88.8%	
	4	.50	1.2%	0.3%	
		.80	15.1%	3.7%	
		.95	61.2%	29.7%	
		.99	90.7%	75.6%	
8	.50	0.1%	0.0%		
	.80	3.6%	0.3%		
	.95	41.3%	6.5%		
	.99	83.6%	44.4%		

TABLE 26 – Taux de neutralité estimés des paysages NK neutres : globaux (ν) et aux optimums locaux (ν_{opt}).

Expérimentation

Nous réutiliserons le même protocole expérimental que dans la section 7.1 — et la même représentation des résultats sous forme de table — à l’exception du critère d’arrêt des recherches. Ici, seul un optimum local strict — n’ayant aucun voisin neutre — stopperait la recherche sous la condition qu’aucun mouvement n’est possible. Or un tel cas de figure peut être rare voire inexistant au sein de paysages neutres. Dans ce cas, nous prévoyons également un nombre maximal de points envisagés, autrement dit d’évaluations de fitness, à ne pas dépasser. Ce paramètre d’horizon T a été fixé dans chaque cas de manière à permettre la convergence de la recherche, avec $T = 10 \times |\mathcal{N}_{\text{flip}}|^2$ pour les paysages à base de chaînes de bits (NK, MAXSAT), et $T = 500 \times |\mathcal{N}_{\{\text{ins}, \text{swap}\}}|$ pour les paysages à base de permutations (QAP_{swap}, FSP_{ins}). Ainsi, pour chaque paysage, le nombre d’évaluations permises est toujours cubique en fonction de la taille de l’instance, puisque la taille du voisinage est linéaire pour NK et MAXSAT (respectivement N et n), et quadratique pour Flowshop et QAP (respectivement $(n-1)^2$ et $\binom{n}{2}$). Ces paramètres ont été réglés de manière à ce qu’aucune amélioration significative des résultats ne puisse être constatée en permettant davantage d’évaluations.

Précisons enfin que, pour un paysage donné, les graines aléatoires et les ensembles de points de départ sont les mêmes pour l’ensemble des climbers considérés dans cette section et dans la section 7.1 (FIRST, BEST, STOCH, FIRST+NP, BEST+NP).

Paysages NK neutres

La table 27.a reporte les résultats obtenus sur les paysages NKp. STOCH est statistiquement dominé sur seulement 8 paysages, contre 14 pour FIRST+NP et 12 pour BEST+NP (considérant un seuil de valeur p de 0.05). BEST+NP tend à être plus performant que les deux variantes de type premier améliorant sur les paysages comportant de faibles taux de neutralité (voir colonne ν) et une faible rugosité ($K \in \{1, 2\}$). Les variantes utilisant la règle du premier améliorant, et en particulier STOCH, atteignent des zones globalement plus hautes sur des paysages plus neutres et rugueux.

Lorsque le paramètre de neutralité p est fixé à 0.99, les trois variantes obtiennent des résultats sensiblement identiques. En fait, il apparaît que ces instances NK probabilistes sont relativement simples à résoudre, puisque une petite proportion seulement de fitness-contributions sont non nulles. Les paysages associés sont en revanche extrêmement plats, et les variantes basiques FIRST et BEST — dont les résultats sur les paysages NK neutres ne sont pas reportés dans ce manuscrit — restent généralement bloqués très tôt. Par exemple, les fitness des optimums atteints par BEST sur ces paysages sont en moyenne 46% inférieurs que BEST+NP sur l’ensemble des paysages d’instances NK probabilistes générées avec $p = 0.99$. Même en se focalisant sur les instances générées avec un paramètre p plus faible, on peut remarquer que globalement, les différentes méthodes sont moins discriminables qu’en utilisant d’autres paysages neutres. De plus, il faut une assez grande proportion de fitness-contributions nulles pour obtenir un taux de neutralité non négligeable du paysage, ce qui réduit l’épistasie et fait perdre la spécificité des paysages NK. Pourtant, le modèle NKp reste utilisé pour l’étude de paysages de fitness neutres [Smith et Smith, 2001; Smith

Instance	ν	STOCH	F.+NP	B.+NP
128.1.p50	11.2%	.4320	.4317	.4366
128.1.p80	44.6%	.2241	.2239	.2253
128.1.p95	81.8%	.0651	.0651	.0653
128.1.p99	94.9%	.0146	.0146	.0146
128.2.p50	5.6%	.4754	.4763	.4806
128.2.p80	30.9%	.2876	.2840	.2868
128.2.p95	73.6%	.1130	.1131	.1138
128.2.p99	94.4%	.0305	.0305	.0306
128.4.p50	1.2%	.5257	.5250	.5233
128.4.p80	14.5%	.3380	.3367	.3363
128.4.p95	60.5%	.1780	.1774	.1769
128.4.p99	89.7%	.0788	.0785	.0789
128.8.p50	0.1%	.5233	.5223	.5161
128.8.p80	3.4%	.3367	.3346	.3317
128.8.p95	41.3%	.1922	.1901	.1896
128.8.p99	83.4%	.1138	.1123	.1129
256.1.p50	11.9%	.4237	.4235	.4283
256.1.p80	45.1%	.2208	.2210	.2222
256.1.p95	81.5%	.0655	.0654	.0655
256.1.p99	96.1%	.0139	.0139	.0139
256.2.p50	5.4%	.4795	.4788	.4850
256.2.p80	30.8%	.2888	.2893	.2915
256.2.p95	74.9%	.1169	.1170	.1176
256.2.p99	94.2%	.0303	.0303	.0304
256.4.p50	1.2%	.5241	.5268	.5230
256.4.p80	15.2%	.3386	.3381	.3364
256.4.p95	60.9%	.1800	.1775	.1801
256.4.p99	90.8%	.0791	.0790	.0788
256.8.p50	0.1%	.5251	.5261	.5208
256.8.p80	3.6%	.3409	.3364	.3342
256.8.p95	41.3%	.1931	.1901	.1918
256.8.p99	83.7%	.1155	.1150	.1144
512.1.p50	12.2%	.4195	.4195	.4238
512.1.p80	45.3%	.2183	.2184	.2199
512.1.p95	81.7%	.0674	.0674	.0676
512.1.p99	96.3%	.0142	.0142	.0142
512.2.p50	5.4%	.4853	.4855	.4901
512.2.p80	31.2%	.2912	.2907	.2934
512.2.p95	74.3%	.1184	.1187	.1192
512.2.p99	94.6%	.0317	.0317	.0317
512.4.p50	1.2%	.5228	.5231	.5216
512.4.p80	15.2%	.3346	.3357	.3341
512.4.p95	61.2%	.1782	.1767	.1783
512.4.p99	90.2%	.0785	.0780	.0781
512.8.p50	0.1%	.5325	.5327	.5260
512.8.p80	3.5%	.3419	.3432	.3383
512.8.p95	41.4%	.1936	.1929	.1935
512.8.p99	83.6%	.1158	.1162	.1147
1024.1.p50	12.0%	.4270	.4272	.4311
1024.1.p80	44.7%	.2231	.2231	.2246
1024.1.p95	81.7%	.0683	.0683	.0685
1024.1.p99	96.1%	.0147	.0147	.0147
1024.2.p50	5.6%	.4861	.4864	.4904
1024.2.p80	31.5%	.2915	.2916	.2937
1024.2.p95	74.3%	.1189	.1188	.1194
1024.2.p99	94.2%	.0319	.0319	.0320
1024.4.p50	1.2%	.5269	.5252	.5258
1024.4.p80	15.1%	.3378	.3357	.3361
1024.4.p95	61.2%	.1787	.1779	.1796
1024.4.p99	90.7%	.0784	.0783	.0783
1024.8.p50	0.1%	.5317	.5323	.5270
1024.8.p80	3.6%	.3449	.3444	.3397
1024.8.p95	41.3%	.1959	.1943	.1943
1024.8.p99	83.6%	.1163	.1163	.1160

(a) NKp

Instance	ν	STOCH	F.+NP	B.+NP
128.1.q10	6.3%	.7294	.7283	.7304
128.1.q5	15.2%	.7484	.7478	.7507
128.1.q3	24.4%	.7902	.7909	.7950
128.1.q2	41.1%	.8594	.8594	.8594
128.2.q10	4.8%	.7297	.7300	.7345
128.2.q5	12.4%	.7687	.7664	.7683
128.2.q3	21.3%	.8186	.8175	.8176
128.2.q2	33.3%	.8914	.8898	.8904
128.4.q10	2.9%	.7557	.7553	.7567
128.4.q5	9.4%	.7922	.7893	.7912
128.4.q3	16.2%	.8396	.8376	.8372
128.4.q2	26.2%	.9309	.9273	.9265
128.8.q10	1.8%	.7417	.7383	.7351
128.8.q5	6.9%	.7734	.7693	.7652
128.8.q3	11.8%	.8199	.8145	.8126
128.8.q2	19.2%	.9020	.8941	.8965
256.1.q10	6.6%	.7271	.7267	.7323
256.1.q5	15.3%	.7518	.7516	.7557
256.1.q3	25.0%	.7938	.7938	.7948
256.1.q2	39.9%	.8672	.8670	.8668
256.2.q10	4.7%	.7318	.7307	.7340
256.2.q5	12.2%	.7610	.7608	.7640
256.2.q3	21.3%	.8183	.8159	.8193
256.2.q2	33.3%	.8907	.8879	.8886
256.4.q10	3.0%	.7503	.7498	.7490
256.4.q5	9.4%	.7865	.7838	.7819
256.4.q3	16.2%	.8436	.8407	.8429
256.4.q2	26.2%	.9363	.9297	.9288
256.8.q10	1.8%	.7425	.7402	.7369
256.8.q5	6.9%	.7762	.7741	.7710
256.8.q3	11.8%	.8255	.8196	.8200
256.8.q2	19.3%	.9098	.9039	.9018
512.1.q10	6.7%	.7126	.7122	.7170
512.1.q5	15.2%	.7408	.7402	.7430
512.1.q3	26.1%	.7823	.7824	.7835
512.1.q2	40.2%	.8368	.8354	.8355
512.2.q10	4.7%	.7399	.7385	.7422
512.2.q5	12.4%	.7722	.7704	.7724
512.2.q3	21.0%	.8187	.8175	.8196
512.2.q2	33.5%	.9060	.9023	.9021
512.4.q10	3.0%	.7494	.7461	.7468
512.4.q5	9.4%	.7862	.7824	.7812
512.4.q3	16.1%	.8398	.8356	.8351
512.4.q2	26.1%	.9348	.9291	.9286
512.8.q10	1.8%	.7457	.7454	.7413
512.8.q5	6.8%	.7804	.7760	.7765
512.8.q3	11.8%	.8320	.8260	.8248
512.8.q2	19.3%	.9132	.9074	.9087
1024.1.q10	6.6%	.7203	.7195	.7256
1024.1.q5	15.2%	.7514	.7509	.7550
1024.1.q3	25.9%	.7955	.7953	.7971
1024.1.q2	40.5%	.8416	.8411	.8414
1024.2.q10	4.7%	.7434	.7422	.7462
1024.2.q5	12.3%	.7768	.7763	.7783
1024.2.q3	21.2%	.8259	.8253	.8260
1024.2.q2	33.7%	.9067	.9036	.9045
1024.4.q10	3.0%	.7540	.7511	.7501
1024.4.q5	9.4%	.7897	.7850	.7858
1024.4.q3	16.2%	.8469	.8424	.8432
1024.4.q2	26.2%	.9348	.9286	.9294
1024.8.q10	1.8%	.7479	.7467	.7428
1024.8.q5	6.9%	.7830	.7789	.7772
1024.8.q3	11.9%	.8350	.8299	.8292
1024.8.q2	19.3%	.9191	.9125	.9121

(b) NKq

Instance	ν	STOCH	F.+NP	B.+NP
128.1.r10000	0.8%	.7023	.7017	.7087
128.1.r1000	9.4%	.7082	.7057	.7103
128.1.r100	65.6%	.7127	.7131	.7125
128.1.r10	93.7%	.5250	.5000	.5000
128.2.r10000	0.7%	.7061	.7036	.7088
128.2.r1000	7.7%	.7155	.7108	.7132
128.2.r100	58.5%	.7313	.7299	.7306
128.2.r10	93.1%	.5310	.5020	.5020
128.4.r10000	0.6%	.7279	.7256	.7264
128.4.r1000	5.9%	.7362	.7304	.7341
128.4.r100	49.5%	.7750	.7608	.7625
128.4.r10	91.1%	.6000	.5340	.5330
128.8.r10000	0.4%	.7145	.7156	.7120
128.8.r10000	4.3%	.7216	.7176	.7193
128.8.r100	39.0%	.7510	.7421	.7353
128.8.r10	88.2%	.6000	.5560	.5560
256.1.r10000	1.8%	.7046	.7039	.7091
256.1.r1000	17.9%	.7134	.7095	.7117
256.1.r100	82.2%	.6874	.6848	.6842
256.1.r10	95.9%	.5000	.5000	.5000
256.2.r10000	1.6%	.7075	.7062	.7104
256.2.r1000	15.1%	.7211	.7157	.7168
256.2.r100	78.4%	.7150	.7124	.7117
256.2.r10	95.2%	.5000	.5000	.5000
256.4.r10000	1.2%	.7243	.7232	.7222
256.4.r1000	11.7%	.7410	.7312	.7311
256.4.r100	72.1%	.7720	.7575	.7599
256.4.r10	93.7%	.5000	.5000	.5000
256.8.r10000	0.9%	.7216	.7177	.7135
256.8.r1000	8.6%	.7282	.7224	.7227
256.8.r100	63.0%	.7714	.7575	.7583
256.8.r10	91.6%	.5010	.5000	.5000
512.1.r10000	4.0%	.6932	.6923	.6966
512.1.r1000	34.5%	.7054	.7016	.7023
512.1.r100	91.2%	.6296	.6021	.6017
512.1.r10	97.1%	.5000	.5000	.5000
512.2.r10000	3.1%	.7170	.7149	.7189
512.2.r1000	29.0%	.7398	.7298	.7301
512.2.r100	89.2%	.6667	.6443	.6430
512.2.r10	97.5%	.5000	.5000	.5000
512.4.r10000	2.4%	.7240	.7228	.7222
512.4.r1000	22.7%	.7543	.7400	.7383
512.4.r100	86.0%	.6926	.6784	.6794
512.4.r10	95.6%	.5000	.5000	.5000
512.8.r10000	1.7%	.7228	.7200	.7192
512.8.r1000	16.9%	.7429	.7310	.7304
512.8.r100	81.1%	.7066	.7115	.7129
512.8.r10	94.0%	.5000	.5000	.5000
1024.1.r10000	7.5%	.7031	.6999	.7059
1024.1.r1000	58.1%	.7135	.7115	.7114
1024.1.r100	95.6%	.5857	.5498	.5492
1024.1.r10	97.9%	.5000	.5000	.5000
1024.2.r10000	6.2%	.7228	.7189	.7233
1024.2.r1000	50.8%	.7498	.7418	.7418
1024.2.r100	94.7%	.6043	.5657	.5652
1024.2.r10	97.5%	.5000	.5000	.5000
1024.4.r10000	4.7%	.7328	.7280	.7276
1024.4.r1000	41.8%	.7748	.7592	.7589
1024.4.r100	93.0%	.6158	.5832	.5851
1024.4.r10	96.9%	.5000	.5000	.5000
1024.8.r10000	3.4%	.7277	.7233	.7229
1024.8.r1000	32.3%	.7617	.7469	.7481
1024.8.r100	90.5%	.6208	.6014	.6016
1024.8.r10	95.8%	.5000	.5000	.5000

(c) NKr

TABLE 27 – Comparaisons des climbers non stricts sur des paysages NK neutres.

et al., 2002; Verel *et al.*, 2011]³⁴.

Dans l’optique de comparer les politiques de sélection des mouvements neutres, les expérimentations sur les paysages NKq amènent des conclusions plus pertinentes. La table 27.b montre des différences très significatives entre les trois stratégies. Les climbers stochastiques atteignent de plus hauts points sur la plupart des instances, en particulier sur les paysages plus rugueux ou neutres (K élevé, q faible). Ce résultat est particulièrement intéressant car il nous semble que les politiques basiques, qui considèrent les mouvements améliorants prioritairement sur les mouvements neutres, sont plus traditionnellement employées au sein des métaheuristiques. C’est surtout la comparaison entre les climbers STOCH et FIRST+NP, deux variantes du premier améliorant, qui permet de montrer l’intérêt d’accepter des voisins neutres tout au long de la recherche, puisqu’il s’agit de la seule différence de conception entre les deux méthodes. Sur les paysages NKq, STOCH obtient de meilleurs résultats en moyenne que FIRST+NP sur tous les paysages à l’exception de trois, ce qui illustre clairement la pertinence de cette politique de gestion de la neutralité. Cependant, une stratégie de type meilleur améliorant, combinée avec des perturbations neutres, reste efficace sur des paysages lisses, plus particulièrement ceux possédant un faible taux de neutralité. Nous observons que la taille du paysage, induite par le paramètre N , n’influence pas la tendance générale des résultats, bien que les différences entre les stratégies aient tendance à être plus significatives sur les espaces de recherche de plus grande taille.

La plupart des conclusions mises en évidence avec les paysages NKq restent valides pour les paysages NKr, comme on peut le constater en analysant la table 27.c. Les climbers stochastiques dominent les autres variantes sur la plupart des instances dont le taux de neutralité est significatif ($\nu \geq 10\%$). Comme sur les paysages NKq, considérer des mouvements neutres même durant les premières étapes de la recherche, alors que ce n’est pas indispensable pour grimper, permet au final d’atteindre des points plus hauts. En effet, sur la plupart des instances, STOCH obtient de meilleurs résultats en moyenne que FIRST+NP. Précisons que fixer r à 10 mène à des paysages extrêmement plats, et une différence significative entre les climbers sur ces paysages n’est observable sur les instances *_*_r10 que pour $N = 128$, où STOCH domine largement FIRST+NP et BEST+NP. Pour de plus grandes valeurs de N , ces paysages plats comportent de gigantesques plateaux, et déterminer des points de sortie (*portails*) nécessitent naturellement l’usage de méthodes spécifiques [Marmion *et al.*, 2011b].

Paysages générés à partir d’instances de problèmes combinatoires

Les résultats obtenus sur les paysages QAP_{swap} sont présentés dans la table 28. Seuls 77 des 132 paysages sont référencés ici, car aucune neutralité n’a été observée sur les 55 paysages restants³⁵. La supériorité de STOCH reste notable sur ce type de paysage, puisqu’il n’est statistiquement dominé que sur 6 instances, et obtient de meilleurs résultats moyens sur la majorité des paysages (44 sur 77). FIRST+NP et BEST+NP sont dominés statistiquement sur respectivement 14 et 23 instances.

³⁴Malgré les apparences, ces publications n’ont aucun auteur en commun.

³⁵Sur les paysages non neutres, FIRST \leftrightarrow STOCH \leftrightarrow FIRST+NP et BEST \leftrightarrow BEST+NP.

Instance	ν	STOCH	FIRST+NP	BEST+NP
bur26a	1.0%	-5 444 414.6	-5 444 336.1	-5 444 520.5
bur26b	2.0%	-3 834 719.0	-3 835 096.8	-3 832 615.9
bur26c	1.1%	-5 450 892.5	-5 452 501.4	-5 446 368.8
bur26d	2.1%	-3 840 939.5	-3 842 342.5	-3 836 004.4
bur26e	1.1%	-5 405 722.2	-5 407 828.6	-5 407 339.0
bur26f	2.3%	-3 798 790.7	-3 800 974.0	-3 799 057.1
bur26g	1.0%	-10 160 063.1	-10 161 134.7	-10 157 156.1
bur26h	2.0%	-7 135 315.6	-7 133 939.9	-7 126 390.1
chr18b	2.5%	-1 703.4	-1 721.2	-1 726.9
chr20a	0.6%	-3 212.7	-3 231.7	-3 169.9
chr20b	0.4%	-3 315.7	-3 286.4	-3 190.4
chr20c	0.1%	-26 552.4	-26 531.9	-26 232.0
chr22a	0.3%	-6 974.8	-6 995.7	-7 019.0
chr22b	0.2%	-7 085.6	-7 098.4	-7 037.9
chr25a	0.3%	-6 059.5	-5 981.3	-6 011.1
els19	0.4%	-22 069 953.1	-22 280 065.1	-21 630 921.3
esc16a	30.1%	-68.1	-68.1	-68.1
esc16b	38.9%	-292.0	-292.0	-292.0
esc32a	12.0%	-145.2	-148.2	-147.6
esc32b	17.7%	-190.4	-192.2	-193.6
esc32d	32.6%	-204.0	-204.9	-205.2
esc32h	25.7%	-448.2	-448.7	-448.6
esc64a	54.7%	-116.0	-116.0	-116.0
had12	3.9%	-1 671.4	-1 670.0	-1 670.1
had14	3.4%	-2 744.5	-2 746.3	-2 746.1
had16	3.1%	-3 747.1	-3 749.3	-3 751.2
had18	2.6%	-5 413.0	-5 416.5	-5 414.8
had20	2.4%	-6 993.6	-7 000.2	-6 991.5
kra30a	3.0%	-94 667.6	-95 086.9	-94 895.9
kra30b	3.0%	-95 624.8	-96 005.7	-95 922.5
kra32	3.1%	-18 831.7	-18 839.7	-18 881.9
lipa20a	2.7%	-3 780.1	-3 784.8	-3 783.2
lipa30a	2.0%	-13 439.1	-13 442.6	-13 436.3
lipa40a	1.3%	-32 002.4	-32 006.2	-32 009.0
lipa50a	1.1%	-62 888.7	-62 879.7	-62 897.4
lipa60a	0.8%	-108 378.3	-108 387.6	-108 396.0
lipa70a	0.7%	-171 365.7	-171 372.6	-171 385.9
lipa80a	0.6%	-255 300.8	-255 292.8	-255 307.1
lipa90a	0.6%	-363 365.9	-363 370.2	-363 418.8

Instance	ν	STOCH	FIRST+NP	BEST+NP
nug12	3.3%	-606.7	-604.1	-604.4
nug14	2.7%	-1 059.3	-1 061.6	-1 059.9
nug15	2.6%	-1 200.3	-1 201.1	-1 194.9
nug16a	2.2%	-1 681.1	-1 678.3	-1 684.9
nug16b	2.3%	-1 301.1	-1 303.6	-1 302.4
nug17	2.2%	-1 795.8	-1 804.9	-1 796.3
nug18	2.1%	-2 011.7	-2 011.0	-2 010.3
nug20	1.9%	-2 666.8	-2 666.2	-2 667.0
nug21	1.9%	-2 531.3	-2 537.1	-2 542.8
nug22	1.4%	-3 717.0	-3 720.5	-3 714.2
nug24	1.7%	-3 638.9	-3 640.7	-3 639.4
nug25	1.8%	-3 866.1	-3 869.6	-3 872.5
nug27	1.4%	-5 427.7	-5 442.7	-5 438.4
nug28	1.5%	-5 370.8	-5 379.5	-5 391.2
nug30	1.4%	-6 347.2	-6 353.7	-6 361.7
scr12	0.3%	-33 644.8	-33 597.0	-33 635.9
scr15	0.2%	-57 154.3	-56 970.4	-56 397.8
sko100a	0.4%	-154 586.1	-154 712.5	-155 045.0
sko100b	0.4%	-156 489.7	-156 598.3	-156 934.7
sko100c	0.5%	-150 833.5	-150 940.9	-151 165.2
sko100d	0.4%	-152 215.8	-152 210.9	-152 571.9
sko100e	0.4%	-152 097.7	-152 102.5	-152 500.2
sko100f	0.5%	-151 587.4	-151 674.6	-152 070.2
sko42	1.0%	-16 302.6	-16 303.5	-16 346.6
sko49	0.9%	-23 994.3	-24 023.5	-24 109.1
sko56	0.7%	-35 343.8	-35 339.0	-35 410.8
sko64	0.7%	-49 563.8	-49 639.6	-49 719.1
sko72	0.6%	-67 715.3	-67 847.3	-67 915.3
sko81	0.5%	-92 739.7	-92 769.6	-92 954.0
sko90	0.5%	-117 689.4	-117 810.5	-117 995.2
ste36a	0.6%	-10 570.4	-10 578.3	-10 616.4
ste36b	0.3%	-18 992.6	-18 969.8	-19 038.5
ste36c	0.2%	-8 961 529.0	-8 945 385.7	-9 021 250.5
taif4c	67.1%	-1 866 610.7	-1 865 318.8	-1 864 589.2
tho30	0.0%	-157 150.6	-157 184.6	-156 876.8
tho40	0.0%	-251 436.9	-251 297.3	-251 890.0
wil50	0.8%	-49 530.3	-49 528.6	-49 528.3
wil100	0.4%	-275 609.0	-275 780.8	-275 881.2

TABLE 28 – Comparaison de climbers non stricts sur des paysages QAP_{swap}.

Instance	ν	STOCH	FIRST+NP	BEST+NP
20_5_01.ta001	21.5%	-1 278.0	-1 280.6	-1 281.4
20_10_01.ta011	9.9%	-1 609.2	-1 613.4	-1 611.2
20_15_01	8.9%	-1 925.1	-1 932.0	-1 931.0
20_20_01.ta021	5.7%	-2 351.0	-2 355.7	-2 371.0
30_5_01	24.7%	-1 726.0	-1 727.1	-1 726.4
30_10_01	11.8%	-2 014.3	-2 028.5	-2 029.6
30_15_01	8.7%	-2 449.6	-2 461.5	-2 461.4
30_20_01	5.3%	-2 825.4	-2 830.29	-2 840.9
50_5_01.ta031	33.0%	-2 724.4	-2 734.8	-2 732.8
50_10_01.ta041	13.8%	-3 038.8	-3 077.4	-3 078.0
50_15_01	8.9%	-3 442.7	-3 467.1	-3 472.5
50_20_01.ta051	7.1%	-3 958.1	-3 973.9	-3 982.1
70_5_01	33.8%	-3 792.0	-3 792.0	-3 792.0

Instance	ν	STOCH	FIRST+NP	BEST+NP
70_10_01	15.6%	-4 059.8	-4 088.3	-4 089.8
70_15_01	10.2%	-4 457.2	-4 471.7	-4 475.0
70_20_01	8.0%	-4 835.7	-4 873.4	-4 889.4
100_5_01.ta061	32.1%	-5 493.0	-5 493.9	-5 493.6
100_10_01.ta071	17.7%	-5 780.1	-5 813.1	-5 820.2
100_15_01	11.2%	-5 974.9	-6 037.9	-6 048.8
100_20_01.ta081	9.0%	-6 329.3	-6 416.6	-6 429.0
150_5_01	35.6%	-8 064.0	-8 064.1	-8 064.0
150_10_01	17.8%	-8 115.7	-8 166.8	-8 174.6
150_15_01	12.6%	-8 543.1	-8 623.1	-8 626.6
150_20_01	9.6%	-9 024.9	-9 158.3	-9 184.8
200_10_01.ta091	20.5%	-10 875.4	-10 917.8	-10 926.4
200_15_01	13.2%	-11 146.1	-11 249.0	-11 288.8
200_20_01.ta101	10.7%	-11 337.6	-11 460.5	-11 478.6

TABLE 29 – Comparaison de climbers non stricts sur des paysages FSP_{ins}.

La table 29 indique une efficacité très nettement supérieure du climber stochastique pour escalader les paysages FSP_{ins} , et donc aborder les instances du flowshop. Sur l'intégralité de ces paysages, la meilleure fitness moyenne est obtenue par **STOCH**, qui en outre domine statistiquement à la fois **FIRST+NP** et **BEST+NP** sur 22 des 27 instances. 2 des 5 instances restantes (70_5_01 et 150_5_01) sont parmi les plus simples, et **STOCH** parvient systématiquement à la solution optimale³⁶ — tout comme sur les instances 20_5_01_ta001, 30_5_01, et 100_5_01_ta061. L'intérêt d'accepter des voisins neutres y compris avant d'atteindre un optimum local a été souligné par les expérimentations précédentes, et se confirme clairement ici.

Rappelons que nous ne nous intéressons pas ici à la résolution du problème ni à la compétition d'algorithmes dédiés au flowshop. C'est pourquoi nous n'indiquons pas les bornes optimales connues, qui sont sur la plupart de ces instances quasi-inatteignables en pratique par de simples climbers ; même si, en extrapolant les observations du chapitre 6 elles devraient l'être davantage en théorie.

Sur les paysages MAXSAT (table 30), les différences entre les climbers sont clairement visibles ; en particulier pour les instances 3-SAT aléatoires (à gauche), où **STOCH** domine statistiquement **FIRST+NP** et **BEST+NP** sur quasiment toutes les instances utilisées. Ces paysages contiennent un fort taux de voisins neutres ($\nu \geq 20\%$) qui doivent clairement être considérés au même titre que des voisins améliorants afin d'atteindre de meilleurs optimums locaux.

Les résultats obtenus à partir d'instances dérivées de problèmes combinatoires (table 30, à droite), sont également très favorables à **STOCH**. Rappelons que le climber strict utilisant la stratégie du meilleur améliorant obtient de meilleurs résultats que la variante de type premier améliorant sur la plupart de ces instances (voir la table 19 du chapitre précédent). En considérant les mouvements neutres, le climber stochastique, basé sur la stratégie du premier améliorant, atteint des points de meilleure fitness en moyenne sur 55 des 68 paysages, et est statistiquement dominé sur seulement 5 paysages par **BEST+NP**, le climber de type meilleur améliorant utilisant des perturbations neutres.

Il est également intéressant d'analyser ces résultats en considérant les spécificités de certaines classes de problèmes en particulier. Les paysages sw^* sont dérivés d'instances SAT encodées à partir d'instances de coloration de graphes, ces derniers étant générés suivant un paramètre de morphing qui détermine leur caractère petit-monde [Gent *et al.*, 1999]. Ces graphes résultent d'un morphing entre un graphe aléatoire et un graphe de type anneau régulier dont chaque sommet est connecté à ses 8 plus proches voisins. Les 10 instances utilisées ici sont ordonnées par niveau de structure, $sw100-8-lp0-c5$ étant obtenue à partir d'un graphe aléatoire, et $sw100-8-p0-c5$ à partir d'un graphe régulier. Gent *et al.* [1999] prétendaient qu'un haut niveau structurel dans les graphes mène à des instances SAT plus difficiles à résoudre par des algorithmes de recherche locale. Cette observation est certes confirmée par les expérimentations sur **FIRST**, **BEST**, **FIRST+NP** et **BEST+NP**, par le fait que le nombre de clauses non satisfaites augmente globalement avec le niveau de structure. Au contraire, **STOCH** est efficace sur toutes ces instances sw^* et particulièrement sur les plus structurées où l'optimum global est presque systématiquement atteint — 97

³⁶Dans des travaux non publiés, Vaessens en 1995 et 1996 a démontré pour ces instances l'optimalité de certaines solutions.

Instances 3-SAT aléatoires						Transformations SAT d'instances de divers problèmes					
Instance	n	ν	STOCH	F+NP	B+NP	Instance	n	ν	STOCH	F+NP	B+NP
uf20-01	20	20.93%	0.00	-0.09	-0.14	flat30-1	90	6.68%	-0.82	-0.98	-0.82
uf50-01	50	22.53%	-0.93	-1.13	-1.33	flat50-1	150	5.55%	-2.35	-2.47	-2.50
uuf50-01	50	24.08%	-1.70	-1.69	-1.84	flat75-1	225	5.47%	-4.35	-4.35	-4.49
uf75-01	75	23.67%	-1.05	-1.61	-1.88	flat100-1	300	5.37%	-5.34	-5.53	-5.63
uuf75-01	75	23.69%	-1.73	-2.13	-2.27	flat125-1	375	5.05%	-6.22	-6.86	-6.61
uf100-01	100	23.23%	-1.43	-2.28	-2.18	flat150-1	450	5.11%	-7.75	-8.09	-8.03
uuf100-01	100	23.84%	-2.45	-2.96	-2.92	flat175-1	525	5.34%	-8.82	9.36	-8.77
uf125-01	125	23.46%	-1.51	-2.10	-2.22	flat200-1	600	5.23%	-9.35	-9.50	-9.37
uuf125-01	125	23.77%	-2.60	-3.20	-3.26	sw100-8-lp0-c5-1	500	0.46%	-1.27	-1.52	-1.40
uf150-01	150	23.12%	-1.16	-1.90	-2.17	sw100-8-lp1-c5-1	500	0.35%	-0.67	-0.81	-0.57
uuf150-01	150	23.47%	-2.22	-3.01	-3.03	sw100-8-lp2-c5-1	500	0.31%	-0.66	-0.92	-0.89
uf175-01	175	22.91%	-2.29	-3.71	-3.31	sw100-8-lp3-c5-1	500	0.25%	-1.62	-1.65	-1.69
uuf175-01	175	23.27%	-3.29	-4.03	-4.28	sw100-8-lp4-c5-1	500	0.22%	-1.32	-2.21	-2.14
uf200-01	200	23.15%	-2.42	-3.78	-3.95	sw100-8-lp5-c5-1	500	0.21%	-1.11	-3.04	-3.14
uuf200-01	200	23.67%	-4.36	-5.62	-5.81	sw100-8-lp6-c5-1	500	0.20%	-1.24	-3.92	-3.74
uf225-01	225	23.11%	-1.40	-3.08	-2.77	sw100-8-lp7-c5-1	500	0.20%	-1.14	-4.59	-4.47
uuf225-01	225	23.37%	-2.94	-4.71	-4.89	sw100-8-lp8-c5-1	500	0.20%	-0.04	-5.12	-5.01
uf250-01	250	22.92%	-0.74	-2.71	-2.64	sw100-8-p0-c5-1	500	0.19%	-0.06	-5.28	-5.22
uuf250-01	250	23.46%	-3.09	-4.85	-4.98	bw_anomaly	48	9.59%	-2.17	-2.59	-3.06
RTI.....m429_0	100	23.32%	-0.85	-1.46	-1.48	bw_medium	116	5.18%	-4.88	-5.80	-6.29
BMS.....m429_0	100	33.46%	-0.98	-1.24	-1.16	bw_huge	459	1.48%	-12.48	-13.37	-14.58
CBS.....m403.b10_0	100	24.17%	-0.59	-1.39	-1.27	bw_large.a	459	2.44%	-8.88	-10.14	-10.52
CBS.....m403.b30_0	100	24.18%	-1.05	-1.52	-1.55	bw_large.b	1087	1.19%	-14.21	-16.98	-16.53
CBS.....m403.b50_0	100	24.20%	-1.35	-2.04	-2.07	bw_large.c	3016	0.41%	-25.45	-27.62	-27.12
CBS.....m403.b70_0	100	24.27%	-1.58	-1.90	-2.01	bw_large.d	6325	0.17%	-36.50	-39.65	-38.38
CBS.....m403.b90_0	100	25.10%	-1.97	-2.57	-2.58	logistics.a	828	1.94%	-20.56	21.22	-19.40
CBS.....m411.b10_0	100	23.48%	-0.62	-0.99	-1.39	logistics.b	843	2.55%	-15.71	16.21	-15.12
CBS.....m411.b30_0	100	24.42%	-1.18	-1.76	-1.73	logistics.c	1141	1.71%	-22.23	-23.50	-21.62
CBS.....m411.b50_0	100	24.71%	-1.18	-1.86	-1.86	logistics.d	4713	9.04%	-98.64	-102.36	-103.06
CBS.....m411.b70_0	100	24.71%	-1.27	-2.01	-2.06	ais6	61	1.57%	-1.19	1.25	-1.19
CBS.....m411.b90_0	100	24.32%	-1.45	-2.32	-2.19	ais8	113	0.57%	-1.86	-1.88	-1.74
CBS.....m418.b10_0	100	23.41%	-0.61	-0.99	-0.90	ais10	181	0.24%	-2.38	-2.40	-2.15
CBS.....m418.b30_0	100	23.25%	-1.00	-1.51	-1.78	ais12	265	0.10%	-2.85	-2.87	-2.50
CBS.....m418.b50_0	100	23.37%	-1.46	-2.00	-1.96	qg1-07	343	0.00%	-22.27	-23.46	-23.69
CBS.....m418.b70_0	100	23.37%	-1.23	-1.67	-1.74	qg1-08	512	0.00%	-27.96	-28.33	-29.09
CBS.....m418.b90_0	100	23.71%	-1.68	-2.05	-2.05	qg2-07	343	0.00%	-21.84	-22.81	-22.71
CBS.....m423.b10_0	100	23.05%	-0.32	-0.79	-0.89	qg2-08	512	0.00%	-26.93	-28.34	-28.53
CBS.....m423.b30_0	100	22.94%	-0.87	-1.55	-1.60	qg3-08	512	0.01%	-30.08	-30.77	-32.27
CBS.....m423.b50_0	100	23.32%	-1.42	-2.05	-1.85	qg3-09	729	0.00%	-38.76	-38.05	-41.16
CBS.....m423.b70_0	100	22.73%	-1.27	-1.56	-1.80	qg4-08	512	0.02%	-34.49	-41.56	-34.07
CBS.....m423.b90_0	100	23.32%	-1.48	-1.85	-1.94	qg4-09	729	0.01%	-45.08	-53.21	-43.97
CBS.....m429.b10_0	100	22.84%	-0.62	-1.25	-1.16	qg5-09	729	0.01%	-86.75	-91.07	-87.78
CBS.....m429.b30_0	100	22.71%	-0.65	-1.41	-1.37	qg5-10	1000	0.00%	-108.67	-111.06	109.40
CBS.....m429.b50_0	100	23.20%	-1.74	-2.21	-2.23	qg5-11	1331	0.00%	-133.00	-138.96	-135.01
CBS.....m429.b70_0	100	23.16%	-0.59	-1.41	-1.49	qg5-12	1728	0.00%	-160.64	-162.52	-159.54
CBS.....m429.b90_0	100	23.16%	-1.63	-2.36	-2.41	qg5-13	2197	0.00%	-189.69	-196.17	-189.85
CBS.....m435.b10_0	100	22.90%	-0.06	-0.62	-0.56	qg6-09	729	0.05%	-69.96	-74.52	-73.28
CBS.....m435.b30_0	100	23.05%	-1.21	-1.69	-1.84	qg6-10	1000	0.02%	-90.04	-93.06	-92.45
CBS.....m435.b50_0	100	22.51%	-1.54	-1.88	-1.94	qg6-11	1331	0.01%	-108.42	-113.06	-111.61
CBS.....m435.b70_0	100	23.46%	-1.29	-1.96	-2.05	qg6-12	1728	0.00%	-132.13	-134.07	-135.78
CBS.....m435.b90_0	100	22.90%	-1.95	-2.28	-2.38	qg7-09	729	0.00%	-74.32	-76.96	-77.73
CBS.....m441.b10_0	100	22.33%	-0.70	-1.28	-1.16	qg7-10	1000	0.00%	-92.39	-96.42	-95.95
CBS.....m441.b30_0	100	22.62%	-0.67	-0.98	-1.13	qg7-11	1331	0.00%	-112.39	-117.34	-116.39
CBS.....m441.b50_0	100	22.71%	-1.31	-1.90	-1.76	qg7-12	1728	0.00%	-134.27	-139.56	-138.85
CBS.....m441.b70_0	100	22.07%	-1.80	-2.36	-2.18	qg7-13	2197	0.00%	-159.68	-165.91	-163.85
CBS.....m441.b90_0	100	23.27%	-2.10	-2.81	-2.56	bmc-ibm-1	9685	10.22%	-212.93	-226.78	-236.29
CBS.....m449.b10_0	100	22.11%	-0.38	-0.83	-0.80	bmc-ibm-2	2810	12.84%	-95.04	-97.83	-109.27
CBS.....m449.b30_0	100	22.68%	-1.40	-1.74	-1.83	bmc-ibm-3	14930	12.64%	-538.19	-559.67	-577.21
CBS.....m449.b50_0	100	22.09%	-1.41	-1.71	-1.74	bmc-ibm-4	28161	11.77%	-898.87	-942.16	-969.38
CBS.....m449.b70_0	100	22.65%	-1.06	-1.75	-1.61	bmc-ibm-5	9396	14.08%	-208.92	-216.21	-270.44
CBS.....m449.b90_0	100	23.16%	-2.00	-2.61	-2.88	bmc-ibm-6	51639	10.35%	-1 370.69	-1 448.51	-1 498.72
						bmc-ibm-7	8710	11.17%	-305.40	-323.16	-332.70
						bmc-galileo-8	58074	13.68%	-1 392.82	-1 371.32	-2 638.74
						bmc-galileo-9	63624	13.62%	-1 496.00	-1 843.45	-2 554.83
						bmc-ibm-10	59056	10.69%	-1 287.22	-1 494.22	-1 395.31
						bmc-ibm-11	32109	13.90%	-1 135.26	-1 181.41	-1 232.00
						bmc-ibm-12	39598	12.70%	-1 299.49	-1 354.53	-1 371.22
						bmc-ibm-13	13215	14.69%	-399.06	-419.25	-446.89

TABLE 30 – Comparaison de climbers non stricts sur des paysages MAXSAT.

fois sur 100 pour sw100-8-lp8-c5-1, et 98 fois sur 100 pour sw100-8-p0-c5-1 au cours de nos tests —, ce qui semble contredire leur affirmation. Plus généralement, on constate que considérer les mouvements neutres tout au long de la recherche, outre le fait de s’avérer plus efficace dans bon nombre de cas, entraîne une dynamique de recherche très différente.

Une seconde observation peut être faite sur la classe de paysages qg^* , obtenus à partir de transformations SAT de problèmes de carrés latins [Zhang et Stickel, 2000]. L’estimation de la neutralité de ces paysages fournit de très faibles taux, alors qu’utiliser les transitions neutres améliore nettement la fitness des points atteints (voir à titre comparatif les résultats des climbers stricts sur ces instances, table 19). Le taux de neutralité estimé nul ou quasi-nul indique *a priori* une absence de neutralité, donc des résultats identiques pour les climbers stricts et non stricts. Cette contradiction nous laisse penser que la neutralité observée durant la recherche est, pour ces instances, sous-estimée par l’indicateur ν . Pourtant, l’étude comparative sur les paysages NK neutres entre la neutralité globale des paysages et celle estimée autour des optimums locaux (voir tableau 26) montrait que l’indicateur ν surestimait, sur ce type de paysages, la neutralité sur les plus hautes zones. Des tests additionnels et plus larges sur l’estimation de la neutralité en fonction des valeurs de fitness —on pourrait dire, en fonction de l’altitude— montrent en effet que la neutralité existe sur ces paysages qg^* , mais uniquement dans des hautes zones. Ces dernières, rares, sont non détectées par un échantillonnage mais sont naturellement largement explorées par les climbers. La figure 35 reporte l’évolution de la neutralité pour quatre paysages 0-1 représentatifs : NK_1024_2_q2, NK_1024_8_r1000, MAXSAT_CBS...m423_b50_0 et MAXSAT_qg5-10. Des résultats similaires sont obtenus à partir d’instances de même type. Pour réaliser cette étude, la neutralité a été estimée à partir de points appartenant à différents intervalles de fitness, obtenus à partir de nombreux itinéraires de recherche fournis par le climber STOCH. Une étude similaire a été menée concernant la rugosité, mais aucune évolution notable n’a été observée en discriminant les intervalles de points par fitness. Sur la figure, on remarque bien, sur le paysage MAXSAT_qg5-10, la neutralité présente uniquement en altitude et qui explique le gain de performance des climbers non stricts. Comme indiqué dans la table 26, la neutralité dans les paysages NK neutres décroît plus ou moins en fonction des paramètres du paysage. Les paysages définis à partir d’instances 3-SAT aléatoires ont une propriété similaire.

Nombre d’évaluations

Les résultats présentés dans cette section se focalisaient exclusivement sur la qualité (hauteur) des points atteints après un nombre d’évaluations fixé à une valeur suffisamment importante pour laisser aux climbers l’opportunité de converger. On a pu constater, à partir de ces résultats, que STOCH est globalement plus performant que les autres variantes de climber. Naturellement, dans un contexte de résolution de problèmes, la vitesse de convergence est également un aspect déterminant pour comparer l’efficacité des recherches locales, et particulièrement des climbers qui peuvent être utilisés comme des mécanismes d’intensification rapide au sein de méthodes plus sophistiquées. Nous avons ainsi mesuré si le fait d’accepter des voisins neutres dès le début de la recherche, ce qui requiert d’explorer en partie plus de voisinages, implique une durée d’escalade plus importante, et donc des temps de calcul accrus.

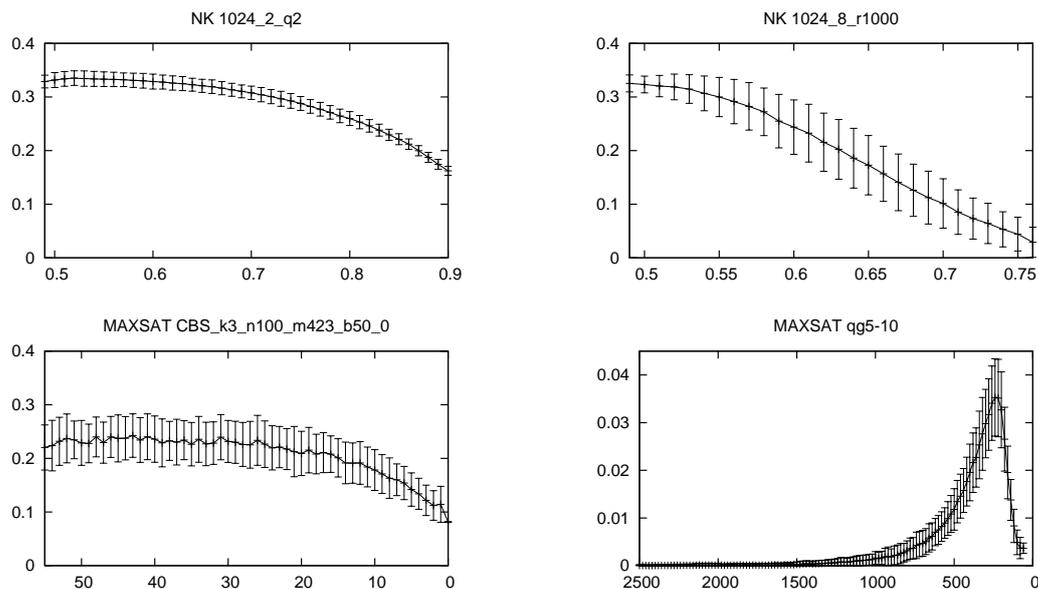


FIGURE 35 – Évolution du degré de neutralité constaté (en ordonnée) en fonction de la fitness des points (en abscisse), pour 4 paysages 0-1 représentatifs (NKq, NKr, 3-SAT aléatoire, MAXSAT basé sur un problème). Pour chaque paysage, les degrés de neutralité des points ont été collectés sur le chemin de 100 escalades **STOCH**.

La figure 36 reporte l'évolution de la fitness moyenne atteinte au cours de la recherche par les trois climbers sur six paysages représentatifs de notre panel utilisé au cours de ce chapitre. Clairement, ces tests relèvent que **STOCH** atteint bien plus rapidement de hauts points du paysage, signifiant que relaxer le critère de mouvement laisse beaucoup plus d'opportunités d'atteindre des zones prometteuses. Intuitivement, réaliser des mouvements neutres augmente la probabilité de s'échapper des petits bassins d'attraction et ainsi d'être attiré par des bassins plus larges, dans lesquels l'escalade est facilitée.

Précisons que cette comparaison computationnelle s'est faite avec le plus grand souci d'équité entre les climbers. C'est pourquoi, dans les algorithmes présentés au début du chapitre (algorithmes 9 et 10), les accès aux fitness *a priori* non connues — représentés par l'usage de la fonction f et dénombrés au moyen de la variable t — se font le plus parcimonieusement possible ; ce qui, comme nous l'avons précisé, peut compliquer légèrement la lecture de ces algorithmes au demeurant très simples.

8.2 Influence des climbers sur l'efficacité d'une recherche locale itérée.

Bien que l'effort computationnel des phases d'intensification des métaheuristiques soit largement dominant, on peut constater que l'effort conceptuel de ces dernières est traditionnellement porté sur les stratégies de diversification. Nous avons ainsi souhaité nous focaliser sur les stratégies de sélection des voisins afin de rendre les phases d'intensification

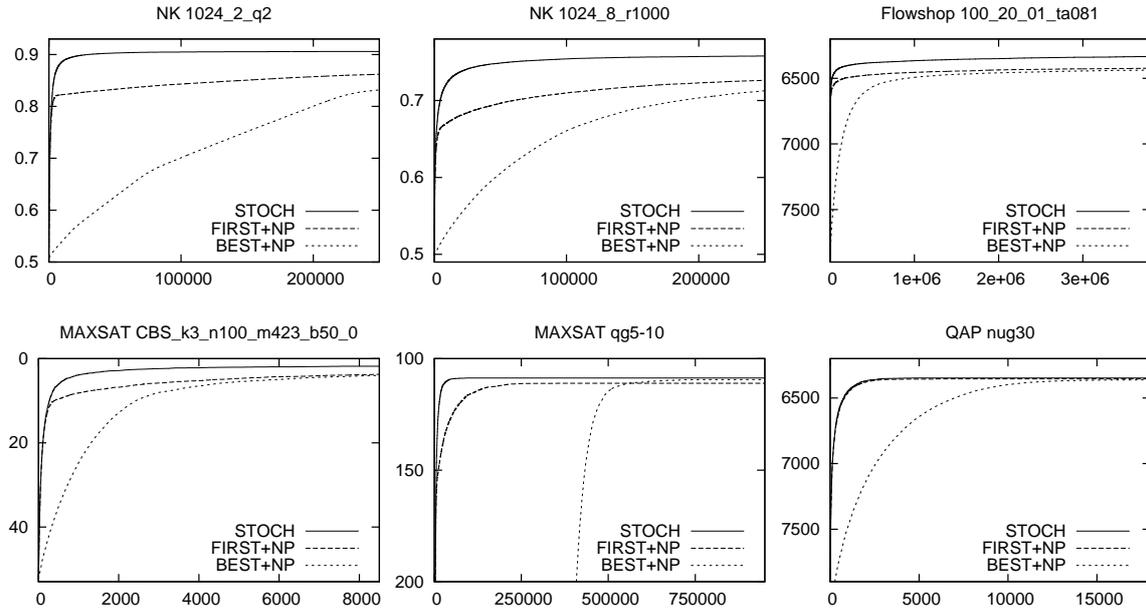


FIGURE 36 – Comparaison des fitness moyennes (en ordonnée) atteintes par 100 exécutions de STOCH, FIRST+NP et BEST+NP, en fonction du nombre de points évalués (en abscisse).

plus efficaces. Bien sûr, il reste à vérifier si le fait de trouver de meilleurs optimums locaux grâce à des techniques purement intensificatrices permet d'améliorer la performance de métaheuristiques plus sophistiquées qui englobent intensification et diversification de la recherche. Nous avons ainsi effectué une large étude expérimentale sur les recherches locales itérées, en comparant les résultats obtenus par 55 recherches locales itérées sur l'ensemble des paysages utilisés dans cette partie. Ces 55 algorithmes — plus précisément séparés en 22 recherches locales strictes et 33 recherches locales non strictes — résultent de toutes les combinaisons envisageables entre les 5 climbers classiques (FIRST, BEST, STOCH, FIRST+NP, BEST+NP) et 11 modes de diversification basés sur trois règles de perturbations : *restart* aléatoire, marche aléatoire depuis le point retourné par le dernier climber, marche aléatoire depuis la meilleure solution rencontrée jusqu'à présent. Les 11 modes de diversifications découlent du paramétrage des marches aléatoires, dont l'importance en nombre de pas est préalablement fixée en paramètre (valeurs testées : 5, 10, 15, 20, 30).

Les résultats de nos tests confirment que l'avantage de performance relative d'un climber est préservé après son intégration à une recherche itérée. Sur la table 31, on remarque que le principal vecteur de performance de ces recherches locales itérées n'est pas la méthode de diversification, mais bien le climber utilisé lors des phases de perturbation. Sur cette table, on ne reporte que les résultats numériques obtenus sur un seul paysage, mais l'intégralité de nos tests montre que la recherche locale itérée la plus performante utilise systématiquement le climber reconnu en moyenne comme le plus performant d'après nos études dédiées.

Détecter le mécanisme de perturbation le plus efficace en fonction des propriétés de

Perturbation	Restart	Marche aléatoire (depuis dernier)					Marche aléatoire (depuis meilleur)				
		Nombre de pas	5	10	15	20	30	5	10	15	20
FIRST	.7200	.7647	.7664	.7655	.7573	.7368	.7573	.7384	.7311	.7273	.7253
BEST	.7104	.7595	.7595	.7578	.7442	.7222	.7539	.7348	.7229	.7186	.7146
STOCH	.7718	.7812	.7823	.7808	.7772	.7741	.7814	.7760	.7738	.7740	.7739
FIRST+NP	.7540	.7690	.7733	.7749	.7721	.7649	.7761	.7692	.7628	.7594	.7570
BEST+NP	.7528	.7626	.7694	.7729	.7715	.7651	.7719	.7735	.7645	.7604	.7566

TABLE 31 – Performance comparée sur le paysage NK neutre 128.8_R100 de 55 variantes de recherche locale itérée, combinaisons de 5 mécanismes d’intensification (climbers utilisés lors des précédents chapitres) et 11 modes de perturbation. Les valeurs reportées sont la moyenne des meilleures fitness atteintes par chacune des 100 exécutions de recherche locale itérée. Une exécution consiste en l’alternance de hill-climblings (chaque climber neutre se termine après $N \times \sqrt{N}$ évaluations sans mouvement strictement améliorant) et de perturbations, et stoppe après un total de $1000 \times N \times \sqrt{N}$ évaluations.

l’instance, et en particulier des indices de rugosité non locale, est un autre objectif de cette expérimentation. L’étude des résultats à cette fin n’est pas achevée et est ainsi à inclure en perspective.

8.3 Une neutralité artificielle pour des recherches locales plus efficaces

8.3.1 Neutralité artificielle et lissage de paysage

L’efficacité comparée des climbers basiques (FIRST, BEST) sur les paysages NK non neutres, et d’un climber stochastique (STOCH) sur les paysages NK arrondis, est particulièrement riche d’enseignements. Rappelons que l’unique différence entre les paysages NK originaux et arrondis (NK_r) vient de la précision des valeurs de fitness. Puisque $\forall x \in \mathcal{X}, F_{\text{NK}_r}(x, r) \leq F_{\text{NK}}(x) < F_{\text{NK}_r}(x, r) + \frac{1}{r}$, il est possible de comparer directement les résultats des paysages NK et NK_r pour chaque paramétrage (N, K) . Ainsi, en comparant les valeurs reportées dans les tables 16 et 27.c, on observe que quelques valeurs de r permettent d’atteindre des solutions de plus haute fitness, malgré un arrondi de fitness défavorable (valeurs tronquées). En soit, ce résultat n’est pas particulièrement surprenant puisque effectuer des mouvements neutres sur un paysage arrondi revient à accepter des mouvements détériorants au sens des fitness originelles (voir représentation unidimensionnelle figure 37) et évite d’être piégé prématurément dans des optimums locaux. Seulement, le point intéressant est que nous n’utilisons pas ici explicitement de stratégie de recherche perturbative telle que le recuit simulé (dont le fonctionnement est proche mais qui emploie explicitement des mouvements détériorants, et dont le paramétrage peut s’avérer complexe), la recherche tabou ou la recherche locale itérée. À la lumière de ces éléments, il nous paraît évident qu’une discrétisation appropriée des fonctions d’évaluation peut aider à la résolution de problèmes d’optimisation au moyen de stratégies de recherche par voisinage.

La table 32 et la figure 38 reportent les valeurs de nos expérimentations précédentes concernant la performance des climbers basiques se basant sur les fitness originelles (paysages NK standards), contre STOCH appliqué aux paysages NK arrondis correspondants

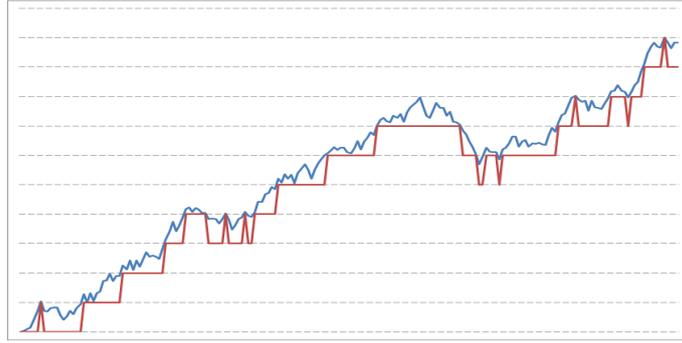


FIGURE 37 – Arrondir la fonction de fitness augmente la neutralité et diminue le nombre d’optimums locaux.

(paysages NK_r , avec $r = 1000$ et $r = 100$). On peut remarquer que les valeurs de r les plus appropriées, parmi celles testées, sont celles qui génèrent ici des paysages de fitness dont le taux de neutralité est compris entre 15% et 75% (voir la table 27.c pour plus de détails). Cela signifie que, pour chaque paramétrage (N, K), un arrondi approprié de la fonction de fitness permet des escalades plus efficaces du moment que la neutralité incorporée est exploitée. Ajouter de la neutralité fait nécessairement décroître les valeurs ρ_k^{\leq} et crée des plateaux qui augmentent le nombre moyen de mouvements nécessaires pour atteindre un optimum local. Régler r consiste alors à déterminer un compromis entre baisser la rugosité et conserver un taux de neutralité raisonnable.

Fitness	F_{NK}	F_{NK_r}	
Arrondi	–	$r = 1000$	$r = 100$
Climber	*	STOCH	STOCH
128.1	.7090	.7082	.7127
128.2	.7082	.7155	.7313
128.4	.7260	.7362	.7750
128.8	.7142	.7216	.7510
256.1	.7079	.7134	.6874
256.2	.7094	.7211	.7150
256.4	.7235	.7410	.7720
256.8	.7166	.7282	.7714

Fitness	F_{NK}	F_{NK_r}	
Arrondi	–	$r = 1000$	$r = 100$
Climber	*	STOCH	STOCH
512.1	.6953	.7054	.6296
512.2	.7174	.7398	.6667
512.4	.7200	.7543	.6926
512.8	.7206	.7429	.7066
1024.1	.7039	.7135	.5857
1024.2	.7197	.7498	.6043
1024.4	.7246	.7748	.6158
1024.8	.7216	.7617	.6208

TABLE 32 – Ajouter artificiellement de la neutralité à des paysages non neutres permet d’atteindre de plus hauts points : comparaison entre le climber basique le plus efficace sur les paysages NK utilisant la fonction de fitness d’origine (F_{NK}), et un climber stochastique sur les paysages correspondants dont les fitness sont tronquées (fonction F_{NK_r}). La colonne * correspond à $\max(\text{FIRST}, \text{BEST})$. Les résultats de STOCH reportés sont ceux de la table 27, pour deux niveaux d’arrondi ($r = \{1000, 100\}$).

Pour aller plus loin dans notre étude, et mesurer plus précisément les valeurs d’arrondi et de neutralité les plus pertinentes en fonction des paysages, nous avons recherché, pour chacune des 16 classes de paysages NK considérées ici (combinaisons de $N \in \{128, 256, 512, 1024\}$ et $K \in \{1, 2, 4, 8\}$), les valeurs de r générant les paysages les plus *faciles* à escalader pour un climber stochastique. On rappelle que des valeurs trop faibles de r impliquent des paysages très plats, sur lesquels un climber s’apparente à une marche aléatoire, tandis

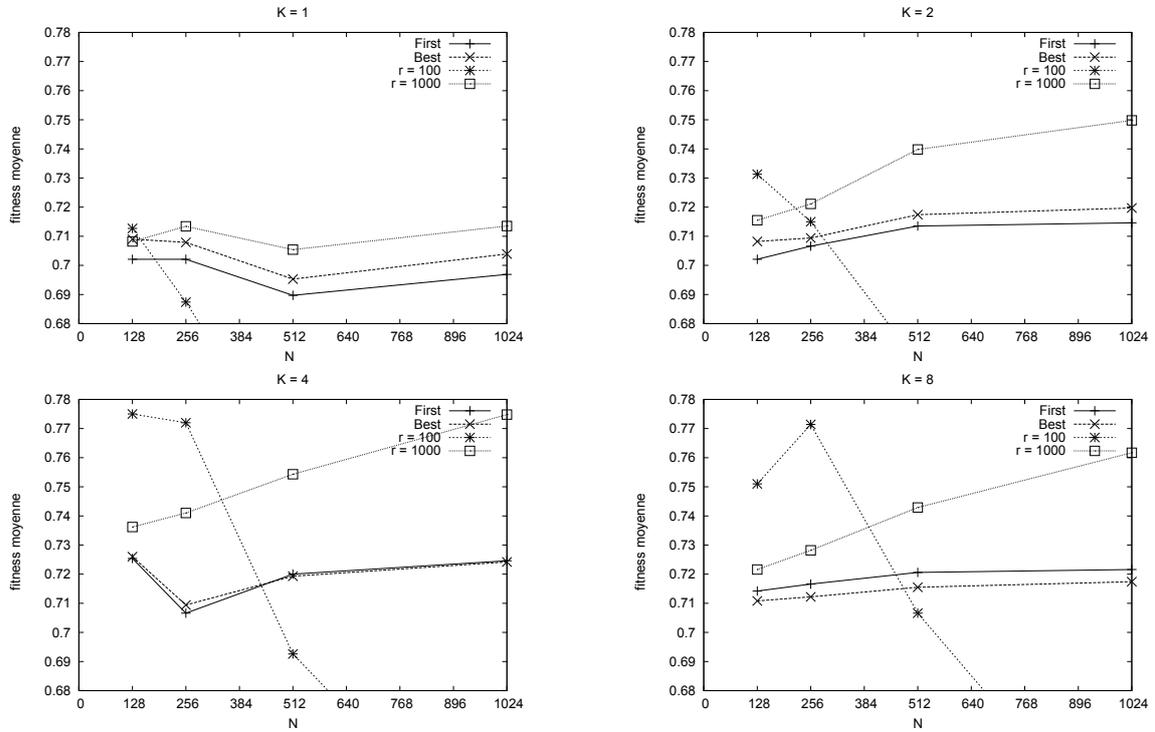


FIGURE 38 – Comparaison entre les climbers basiques sur des paysages non neutres, et un climber stochastique sur les mêmes paysages arrondis.

N_K	Arrondis dominants	Fitness moyennes
128.1	[206, 285]	.7227
128.2	[94, 148]	.7395
128.4	[54, 81]	.7837
128.8	[40, 64]	.7685
256.1	[321, 350]	.7206
256.2	[212, 264]	.7410
256.4	[117, 152]	.7843
256.8	[84, 112]	.7755

N_K	Arrondis dominants	Fitness moyennes
512.1	[731, 824]	.7069
512.2	[454, 523]	.7484
512.4	[234, 304]	.7790
512.8	[173, 206]	.7810
1024.1	[1371, 1562]	.7157
1024.2	[959, 1178]	.7507
1024.4	[449, 570]	.7844
1024.8	[331, 403]	.7836

TABLE 33 – Paysages NKr les plus adaptés à une escalade par climber stochastique pour chaque classe de paysage NK (intervalles de valeurs de r). Les fitness atteintes par des climbers sur les paysages NK non arrondis sont rappelées table 32 (colonne F_{NK}).

que des valeurs de r trop élevées ne gomment pas suffisamment les optimums locaux et bloquent les climbers plus rapidement.

La table 33 indique les intervalles de valeurs d'arrondi r à partir desquels les climbers exécutés sur les paysages NKr correspondants ne sont pas dominés statistiquement par ceux exécutés sur des paysages NKr générés à partir d'autres valeurs de r . Contrairement aux escalades de paysages NKr, nous mémorisons ici le meilleur point atteint via la fonction NK d'origine (fitness non arrondies). En effet, puisque nous comparons ici l'influence de l'ajout de neutralité artificielle sur la capacité à escalader des paysages originellement NK, et non l'influence de la politique de mouvement pour escalader des paysages neutres, les comparaisons doivent s'effectuer au moyen de la même mesure de fitness. En analysant les résultats, on constate que les valeurs optimales de r — qui ont été déterminées par dichotomie au moyen d'un algorithme que nous ne détaillerons pas — dépendent de N et K . Cependant, nous avons vu (table 26) que les taux de neutralité des paysages NKr dépendaient également de N et K , mais aussi que ces taux de neutralité baissaient au cours de la recherche. Nous avons alors établi une mesure de neutralité moyenne observée au cours de la recherche, afin de déterminer au moyen d'un indicateur commun si l'efficacité des climbers dépendait uniquement de la neutralité du paysage. Cette *neutralité observée* (définition 14) fait décroître l'influence des mouvements neutres observés durant de plus longues périodes de stagnation afin de mieux rendre compte de la neutralité globalement rencontrée sur l'ensemble des phases de la recherche. Si les valeurs effectives des neutralités observées ne seront pas nécessairement d'une pertinence capitale, celles-ci étant dépendantes de la formule qui reste perfectible, l'intérêt résidera dans la comparaison de ces valeurs sur différentes classes de paysages.

Définition 14 (Neutralité observée) Soit \mathcal{P} un paysage de fitness dont la taille du voisinage³⁷ est N , et C le parcours d'un climber stochastique défini par la suite des points évalués et des mouvements effectués. Soit p le nombre de mouvements strictement améliorants de C . Soient l_i et n_i ($i \in \{1, \dots, p\}$) respectivement le nombre d'évaluations et le nombre de mouvements neutres effectués entre les mouvements améliorants $i - 1$ et i (avant le premier mouvement améliorant pour $i = 1$). La neutralité observée par C sur \mathcal{P} se définit par :

$$\tilde{\nu}(C, \mathcal{P}) = \frac{\sum_{i=1}^p \frac{n_i}{l_i} \log_N l_i}{\sum_{i=1}^p \log_N l_i} \quad (37)$$

La figure 39 représente l'efficacité des climbers sur chacune des 16 classes de paysages NKr en fonction de la neutralité observée des paysages NKr. La partie droite indique la neutralité observée des paysages NKr à arrondis dominants. Malgré des valeurs de r très diverses en fonction de N et K , cette représentation indique des neutralités artificielles optimales très proches (entre 0.2 et 0.3 selon l'indicateur de neutralité observée défini ci-dessus).

³⁷Si tous les points de l'espace de recherche n'ont pas un nombre de voisins identique, N représente la taille moyenne ou estimée du voisinage d'un point.

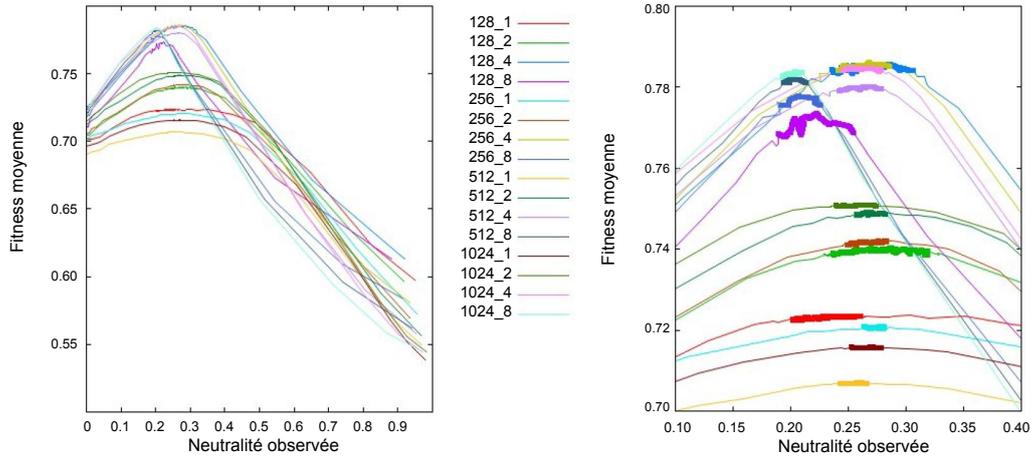


FIGURE 39 – Fitness moyennes atteintes par des climbers stochastiques en fonction des neutralités observées des paysages NKr. Sur la partie droite, les intervalles correspondants à des ensembles de fitness dominants sont épaissis.

8.3.2 Paysages dynamiques à neutralité contrôlée

Puisque la neutralité est variable au cours de la recherche pour chaque paysage NKr fixe, nous avons mis au point un climber sélectionnant de manière adaptative l'arrondi r s'approchant du niveau de neutralité souhaité. Cette stratégie consiste à escalader un paysage dynamique à neutralité artificielle. Étant donné un indice de neutralité de référence ν_{ref} et un ensemble de valeurs d'arrondi entières $\{r_1, \dots, r_n\}$ ³⁸, l'algorithme considère à chaque itération la valeur d'arrondi r_i de meilleure utilité, de manière à conserver tout au long de la recherche un niveau de neutralité artificielle maximisant l'efficacité de l'escalade. Les neutralités estimées de chaque arrondi r_i sont mises à jour à chaque itération selon une moyenne incrémentale renforçant les d dernières itérations. Soit, après l'itération t et pour tout r_i :

$$\nu_{\text{est}}[r_i] \leftarrow \begin{cases} \alpha \times \nu_{\text{est}}[r_i] + 1 - \alpha, & \text{si } f(x', r_i) = f(x, r_i) \\ \alpha \times \nu_{\text{est}}[r_i], & \text{sinon} \end{cases} \quad (38)$$

$f(x, r_i)$ correspond à la fitness arrondie³⁹ de x au moyen du paramètre r_i (soit $F_{\text{NKr}}(x, r)$ pour les paysages NK). $\alpha = 1 - \frac{1}{\min(d, t)}$ équilibre le poids de la dernière évaluation dans l'estimation des r_i .

³⁸Le fait d'utiliser plusieurs valeurs d'arrondi au cours de la recherche peut entraîner une escalade anormalement chaotique. Des arrondis très proches auraient ainsi une efficacité similaire, mais utilisés alternativement, en raison du décalage de leurs paliers de fitness, ils peuvent avoir pour effet de générer une trajectoire descendante. Pour éviter ce phénomène, il faut que les facteurs premiers des r_i soient tous identiques. Le plus grand ensemble respectant cette condition est celui des puissances de 2 ; c'est pourquoi l'ensemble des valeurs d'arrondi que nous avons utilisé dans toutes nos expérimentations est l'ensemble $\{2^i, 1 \leq i \leq 15\}$. La fitness arrondie au moyen du paramètre d'arrondi maximal doit équivaloir à la fonction de fitness non arrondie. Il est cependant possible d'ajouter la fonction de fitness non arrondie à l'ensemble de fitness utilisé ici.

³⁹Le mécanisme d'arrondi présenté ici n'est pertinent que si les fitness prennent leurs valeurs dans l'intervalle $[0, 1]$. Dans le cas contraire, il suffit de procéder à une normalisation.

L'arrondi sélectionné à chaque itération est celui correspondant à une neutralité estimée la plus proche (en termes de rapport) de la neutralité de référence désirée, soit plus précisément $\operatorname{argmin}_{r_i} \frac{\max(\nu_{\text{est}}(r_i), \nu_{\text{ref}})}{\min(\nu_{\text{est}}(r_i), \nu_{\text{ref}})}$. La seule exception est que si des arrondis supérieurs ont des neutralités estimées quasiment identiques au regard d'un seuil θ , ceux-ci seront préférés, afin de faciliter le franchissement des paliers. On précise que nécessairement, à tout moment de la recherche, $(r_i < r_j) \Rightarrow (\nu_{\text{est}}(r_i) \geq \nu_{\text{est}}(r_j))$. À l'itération $t = 0$, les valeurs des neutralités estimées ne sont pas définies et on sélectionne alors un voisin aléatoirement.

La table 34 reporte les résultats obtenus par ce climber stochastique sur paysages à arrondi adaptatif (avec paramètre ν_{ref} fixé à 0.25 et $d = 1000$), noté STOCH_ν , que l'on peut comparer aux performances d'un climber stochastique classique STOCH sur paysages à arrondi fixe (table 33). Ces fitness moyennes sont, comme à l'habitude, calculées à partir des fitness atteintes par 100 escalades (limitées à $10 \times N^2$ évaluations) ; les différentes stratégies comparées emploient les mêmes ensembles de 100 points de départ. Sur ces paysages que l'on pourrait qualifier de dynamiques, les escalades ont été facilitées puisque les climbers atteignent en moyenne des points bien plus hauts qu'avec les meilleurs paramétrages d'arrondi fixe, parfois très largement comme sur le paysage NK_{128_8} .

N_K	STOCH	STOCH $_\nu$	N_K	STOCH	STOCH $_\nu$
128.1	.7227	.7245	512.1	.7069	.7066
128.2	.7395	.7422	512.2	.7484	.7500
128.4	.7837	.7947	512.4	.7790	.7819
128.8	.7685	.7919	512.8	.7810	.7852
256.1	.7206	.7217	1024.1	.7157	.7160
256.2	.7410	.7441	1024.2	.7507	.7509
256.4	.7843	.7910	1024.4	.7844	.7840
256.8	.7755	.7813	1024.8	.7836	.7835

TABLE 34 – Efficacité du climber stochastique utilisant une neutralité de référence fixe et un arrondi de la fonction de fitness adaptatif (STOCH). La colonne STOCH reporte les fitness moyennes atteintes par un climber stochastique utilisant une valeur dominante d'arrondi fixe (voir table 33).

8.3.3 Recherche itérée simulée par l'escalade de paysages dynamiques

L'ensemble de ces résultats sur les climbers stochastiques et la neutralité artificielle nous a conduit à définir une stratégie de recherche plus générale dont l'intensification et la diversification ne sont pas explicitées par la politique de mouvement mais par l'arrondi de la fonction de fitness. La stratégie d'escalade définie ci-avant s'applique à des paysages virtuellement dynamiques au regard de la fonction de fitness arrondie choisie à chaque itération ; le point central étant de conserver un niveau de neutralité permettant d'atteindre de plus hautes fitness. Quelques ajustements suffisent pour concevoir une méthode de recherche locale originale, efficace et autonome.

Tout d'abord, conserver un certain taux de neutralité empêche d'intensifier la recherche pour déterminer des optimums locaux (au sens de la fonction de fitness non arrondie), ce qui pourrait être fait, par exemple, en fin de recherche en supprimant la neutralité artificielle. Seulement, le principe du climber stochastique fait que cette fin de recherche est simplement dictée par un nombre d'itérations donné en paramètre et qui peut être

difficile à établir. Afin de pallier ces deux limitations, nous proposons d’associer un *potentiel d’amélioration* μ_{est} à chaque valeur d’arrondi r_i , mis à jour de la même manière que ν_{est} à la différence que l’on estime ici non pas les voisins neutres, mais les voisins strictement améliorants. Cela permet de détecter les stagnations de la recherche au regard de chaque valeur d’arrondi. Plus précisément, le critère de sélection du r_i reste le même que précédemment, en fonction des neutralités estimées $\nu_{\text{est}}[r_i]$, à la seule exception que les r_i dits *stagnants* (c’est-à-dire tels que $\mu_{\text{est}}[r_i]$ sont inférieurs à un seuil très faible ϵ à une itération $t > d$) ne peuvent pas être choisis. Mécaniquement, cela conduit l’escalade à converger vers un optimum local ; le paramètre du nombre d’itérations disparaît également, remplacé par le seuil ϵ bien moins déterminant.

Enfin, pour simuler une perturbation de l’optimum local, il suffit de réinitialiser t à 0. Cela a pour effet immédiat de considérer pendant une itération un paysage plat, puis le mécanisme de neutralité artificielle adaptative permet de retrouver, en quelques itérations suffisantes pour diversifier la recherche, le comportement précédent. Le fonctionnement général de cette stratégie autonome d’escalade de paysages dynamiques, notée STOCH_ν^* , est résumé par l’algorithme 11. Le critère d’arrêt spécifie un nombre d’évaluations cible atteint pour plus d’uniformité avec les précédents algorithmes, mais un nombre de diversifications préfixé reste une possibilité.

La table 35 met en parallèle, à titre indicatif, les fitness moyennes obtenues par STOCH_ν^* (avec justement 200 diversifications par exécution) avec celles atteintes par des recherches locales itérées classiques exécutées sur les paysages NK originaux (non arrondis). La colonne ILS reporte pour chaque instance la moyenne obtenue par la *meilleure* des 22 variantes de recherches locales itérées strictes évoquées dans la section 8.2. Ces recherches itérées ont été effectuées dans le cadre d’une autre étude ; le nombre d’itérations fixé à $100 \times N^{3/2}$ pour un nombre de diversifications moyen de plusieurs milliers à chaque exécution, ainsi que le report de la meilleure moyenne parmi 22 algorithmes, apportent en théorie un net avantage protocolaire à ILS. STOCH_ν est le climber stochastique adaptatif n’utilisant pas de mécanisme perturbatif, dont les résultats sont reportés depuis la table 34.

Comme nous l’avons mentionné précédemment, il est évident qu’un tel mécanisme apparente davantage le comportement du climber dans le paysage originel à celui d’un recuit simulé (pour la phase ascendante) et donc à une recherche itérée (pour la réinitialisation des neutralités estimées). Cependant, cette abstraction de la recherche locale en escalade de paysages de fitness nous a conduits à transférer le mécanisme perturbatif non pas au critère de sélection, mais à la fonction d’évaluation. L’idée intéressante selon nous est que le modèle d’escalade reste ici simple et uniforme, et respecte nos conclusions sur la pertinence des mécanismes intensificateurs. Nous savons que la diversification de la recherche est un aspect crucial de l’efficacité des métaheuristiques mais celle-ci est délicate à paramétrer, et l’effet réel de chaque mécanisme perturbatif difficile à mesurer. Ici, la diversification n’est pas explicitée par notre modèle d’abstraction, mais est simplement induite par l’incorporation d’une neutralité artificielle qui modifie dynamiquement le paysage.

Algorithme 11 : Algorithme STOCH*.

Données : un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$, un point de départ $x_0 \in \mathcal{X}$, un ensemble de n valeurs d'arrondi de fitness $\mathcal{R} = \{r_1, \dots, r_n\}$ (avec $r_i > r_{i+1}$), une instanciation des paramètres $d, \nu_{\text{ref}}, \theta, \epsilon$, et un nombre d'accès à des valeurs de fitness (horizon T).

Résultat : un optimum approché x_{opt} et sa fitness F_{opt}

```
1  $x \leftarrow x_0$ ;
2  $F \leftarrow f(x)$ ;
3  $t \leftarrow 1$ ; //  $t$  comptabilise le nombre total d'itérations (accès à  $f$ )
4  $t \leftarrow 0$ ; //  $t$  comptabilise le nombre d'itérations depuis la dernière perturbation
5  $F_{\text{opt}} \leftarrow F$ ;
6  $x_{\text{opt}} \leftarrow x$ ;
7 répéter
8   Sélectionner aléatoirement  $x' \in \mathcal{N}(x)$ ;
9    $F' \leftarrow f(x')$ ;  $t \leftarrow t + 1$ ;  $t \leftarrow t + 1$ ;
10  // Mise à jour des  $\nu_{\text{est}}[r_i]$  et  $\mu_{\text{est}}[r_i]$ 
11   $\alpha \leftarrow 1 - \frac{1}{\min(d, t)}$ ;
12  pour tout  $r_i \in \mathcal{R}$  faire
13     $\nu_{\text{est}}[r_i] \leftarrow \alpha \times \nu_{\text{est}}[r_i]$ ; // nécessairement 0 si  $\nu_{\text{est}}[r_i]$  est non défini
14     $\mu_{\text{est}}[r_i] \leftarrow \alpha \times \mu_{\text{est}}[r_i]$ ; // nécessairement 0 si  $\mu_{\text{est}}[r_i]$  est non défini
15    si  $\lfloor r_i \times F' \rfloor = \lfloor r_i \times F \rfloor$  alors  $\nu_{\text{est}}[r_i] \leftarrow \nu_{\text{est}}[r_i] + 1 - \alpha$ 
16    sinon si  $\lfloor r_i \times F' \rfloor > \lfloor r_i \times F \rfloor$  alors  $\mu_{\text{est}}[r_i] \leftarrow \mu_{\text{est}}[r_i] + 1 - \alpha$ 
17  finpour
18  // Choix de l'arrondi
19   $\mathcal{R}' \leftarrow \{r_i \in \mathcal{R}, \nu_{\text{est}}[r_i] > 0\}$ ;
20  si  $t > d$  alors  $\mathcal{R}' \leftarrow \{r_i \in \mathcal{R}', \mu_{\text{est}}[r_i] \geq \epsilon\}$ ;
21  si  $\mathcal{R}' = \emptyset$  alors
22    |  $t \leftarrow 0$  // Diversification
23  sinon
24    |  $R \leftarrow \operatorname{argmin}_{r_i \in \mathcal{R}'} \frac{\max(\nu_{\text{est}}[r_i], \nu_{\text{ref}})}{\min(\nu_{\text{est}}[r_i], \nu_{\text{ref}})}$ ;
25    |  $\mathcal{R}' \leftarrow \{r_i \in \mathcal{R}', \nu_{\text{est}}[R] - \theta \leq \nu_{\text{est}}[r_i] \leq \nu_{\text{est}}[R]\}$ ;
26    |  $R \leftarrow \operatorname{argmin}_{r_i \in \mathcal{R}'} \nu_{\text{est}}[r_i]$ ;
27    | // Politique de mouvement
28    | si  $\lfloor R \times F' \rfloor \geq \lfloor R \times F \rfloor$  alors
29    | |  $x \leftarrow x'$ ;
30    | |  $F \leftarrow F'$ ;
31    | | si  $F > F_{\text{opt}}$  alors
32    | | |  $x_{\text{opt}} \leftarrow x$ ;
33    | | |  $F_{\text{opt}} \leftarrow F$ 
34    | | fin
35    | fin
36  fin
37 jusqu'à  $t = T$ ;
38 retourner  $(x_{\text{opt}}, F_{\text{opt}})$ 
```

Instance	STOCH _v	STOCH _v *	ILS
128_1	.7245	.7245	.7245
128_2	.7422	.7424	.7414
128_4	.7947	.7959	.7921
128_8	.7919	.8050	.7807
256_1	.7217	.7221	.7221
256_2	.7441	.7448	.7443
256_4	.7910	.7940	.7882
256_8	.7813	.7987	.7815

Instance	STOCH _v	STOCH _v *	ILS
512_1	.7066	.7088	.7088
512_2	.7500	.7520	.7502
512_4	.7819	.7888	.7769
512_8	.7852	.7920	.7837
1024_1	.7160	.7174	.7174
1024_2	.7509	.7532	.7518
1024_4	.7840	.7910	.7778
1024_8	.7835	.7893	.7817

TABLE 35 – Efficacité de la recherche stochastique autonome STOCH_v* sur des paysages NK (critère d’arrêt : 200 diversifications par exécution), en comparaison avec des recherches locales itérées classiques (ILS).

8.4 Climber stochastique et règle du moins bon améliorant

Dans ce chapitre, nous avons comparé les deux politiques de neutralité envisageables par les algorithmes d’escalade, selon qu’ils considèrent prioritairement les mouvements strictement améliorants (basique avec perturbation neutre) ou non (stochastique). L’intérêt d’utiliser cette seconde règle a pu être attesté, au point qu’associée à l’incorporation artificielle de la neutralité au paysage, elle permettait de définir des recherches locales plus efficaces. Telle qu’utilisée traditionnellement, sous son appellation climber stochastique, netcrawling ou autre, cette stratégie d’escalade emploie logiquement la règle du premier améliorant (voir section 5.6). Après avoir attesté de la supériorité des climbers basiques utilisant soit la règle du moins bon améliorant, soit celle du meilleur améliorant (en fonction des paysages), nous avons défini une stratégie permettant de combiner la politique de mouvements neutres du climber stochastique avec ces autres règles pivot. L’idée est de réaliser la même proportion de mouvements neutres qu’un climber stochastique tel que défini précédemment. Il suffit pour cela, lors de l’évaluation du voisinage — qui doit être effectuée dans un ordre aléatoire —, de sélectionner le premier voisin neutre rencontré si et seulement si aucun mouvement strictement améliorant n’a encore été évalué. Sinon, le voisin est sélectionné en fonction de la stratégie décrite par la règle pivot. L’algorithme 12 illustre un climber utilisant la règle du moins bon améliorant et une gestion dite stochastique de la neutralité, que l’on appellera STOCH-W (STOCH-W_κ pour ses variantes partielles). Une version similaire a été définie via la règle du meilleur améliorant.

La comparaison des résultats obtenus par les différents climbers sur les paysages FSP_{ins} (tables 18, 22 et 29) nous a permis de hiérarchiser clairement les différentes stratégies. Si WORST reste trop couteux pour présenter un intérêt sur ce type de paysage, il apparaît que parmi les climbers basiques, les variantes partielles de WORST dominent globalement FIRST, qui lui-même est bien plus efficace que BEST. Seulement, sur ces paysages présentant des taux de neutralité importants, un climber stochastique (STOCH, basé sur la règle du premier améliorant) reste bien plus efficace. Nous avons alors comparé les performances de STOCH avec des variantes partielles de STOCH-W, dont les résultats sont présentés table 36. On y remarque que tous les climbers STOCH-W_κ dominent STOCH, signifiant que les climbers stochastiques peuvent être combinés à d’autres règles pivot que celle du premier améliorant, pour davantage d’efficacité. En outre, cela renforce les observations précédentes qui tendent à montrer que la règle pivot d’une recherche locale reste influente sur des

Algorithme 12 : Variante de WORST avec gestion stochastique de la neutralité (STOCH-W).

Données : un paysage de fitness $(\mathcal{X}, \mathcal{N}, f)$, un point de départ $x_0 \in \mathcal{X}$, un nombre d'accès à des valeurs de fitness à ne pas dépasser (T).

Résultat : un optimum approché x_{opt} et sa fitness F_{opt} .

```

1  $x \leftarrow x_0$ ;
2  $F \leftarrow f(x_0)$ ;  $t \leftarrow 1$ ;
3 bloque  $\leftarrow$  faux;
4 répéter
5    $N \leftarrow \mathcal{N}(x)$ ;
6    $F_w \leftarrow +\infty$ ;
7   tant que  $N \neq \emptyset$  faire
8     Sélectionner aléatoirement  $x' \in N$ ;
9      $F' \leftarrow f(x')$ ;  $t \leftarrow t + 1$ ;
10    si  $F' = F$  et  $F_w \neq \infty$  alors
11       $x_w \leftarrow x'$ ;
12       $F_w \leftarrow F'$ ;
13       $N \leftarrow \emptyset$ 
14    sinon
15      si  $F' < F_w$  et  $F' > F$  alors
16         $x_w \leftarrow x'$ ;
17         $F_w \leftarrow F'$ 
18      fin
19       $N \leftarrow N \setminus \{x'\}$ 
20    fin
21  fin
22  si  $F_w \neq \infty$  alors
23     $x \leftarrow x_w$ ;
24     $F \leftarrow F_w$ 
25  sinon bloque  $\leftarrow$  vrai
26 jusqu'à bloque = vrai ou  $t = T$ ;
27  $x_{\text{opt}} \leftarrow x$ ;
28  $F_{\text{opt}} \leftarrow F$ ;
29 retourner  $(x_{\text{opt}}, F_{\text{opt}})$ 

```

stratégies de sélection non basiques (comme une recherche locale itérée ou simplement, ici, une gestion stochastique de la neutralité).

Instance	FIRST	\mathcal{W}_{16}	STOCH	STOCH- \mathcal{W}_2	STOCH- \mathcal{W}_4	STOCH- \mathcal{W}_8	STOCH- \mathcal{W}_{16}
20_5_01.ta001	-1 306.7	-1 305.3	-1 278.0	-1 278.0	-1 278.0	-1 278.0	-1 278.0
20_10_01.ta011	-1 629.3	-1 629.5	-1 609.2	-1 611.1	-1 608.8	-1 610.2	-1 609.1
20_15_01	-1 961.4	-1 964.4	-1 925.1	-1 926.4	-1 926.3	-1 927.2	-1 924.0
20_20_01.ta021	-2 374.7	-2 367.5	-2 351.0	-2 348.0	-2 349.8	-2 348.9	-2 346.0
30_5_01	-1 736.8	-1 735.3	-1 726.0	-1 726.0	-1 726.0	-1 726.0	-1 726.0
30_10_01	-2 072.2	-2 063.6	-2 014.3	-2 007.4	-2 006.9	-2 005.5	-2 005.7
30_15_01	-2 510.7	-2 489.1	-2 449.6	-2 443.9	-2 442.2	-2 440.5	-2 437.6
30_20_01	-2 856.6	-2 840.7	-2 825.4	-2 819.8	-2 814.7	-2 810.6	-2 812.5
50_5_01.ta031	-2 743.3	-2 741.2	-2 724.4	-2 724.0	-2 724.0	-2 724.0	-2 724.0
50_10_01.ta041	-3 140.2	-3 127.3	-3 038.8	-3 034.0	-3 037.1	-3 033.5	-3 034.6
50_15_01	-3 535.6	-3 506.0	-3 442.7	-3 439.5	-3 435.4	-3 429.6	-3 428.8
50_20_01.ta051	-4 033.7	-4 010.2	-3 958.1	-3 953.3	-3 948.2	-3 945.4	-3 941.9
70_5_01	-3 794.4	-3 794.1	-3 792.0	-3 792.0	-3 792.0	-3 792.0	-3 792.0
70_10_01	-4 168.0	-4 145.4	-4 059.8	-4 050.6	-4 051.6	-4 050.0	-4 049.3
70_15_01	-4 566.6	-4 526.3	-4 457.2	-4 438.0	-4 434.2	-4 425.7	-4 434.8
70_20_01	-4 980.5	-4 927.5	-4 835.7	-4 823.0	-4 810.1	-4 810.5	-4 803.6
100_5_01.ta061	-5 510.0	-5 508.8	-5 493.0	-5 493.0	-5 493.0	-5 493.0	-5 493.0
100_10_01.ta071	-5 864.5	-5 850.3	-5 780.1	-5 773.7	-5 773.2	-5 773.8	-5 774.3
100_15_01	-6 155.3	-6 117.7	-5 974.9	-5 952.7	-5 952.9	-5 951.0	-5 951.2
100_20_01.ta081	-6 560.5	-6 498.7	-6 329.3	-6 298	-6 300.6	-6 294.4	-6 294.1
150_5_01	-8 066.3	-8 065.0	-8 064.0	-8 064.0	-8 064.0	-8 064.0	-8 064.0
150_10_01	-8 240.6	-8 205.0	-8 115.7	-8 107.4	-8 106.6	-8 108.0	-8 108.0
150_15_01	-8 676.8	-8 648.4	-8 543.1	-8 493.8	-8 490.7	-8 491.0	-8 494.2
150_20_01	-9 327.7	-9 237.8	-9 024.9	-8 972.7	-8 973.7	-8 966.7	-8 951.6
200_10_01.ta091	-11 006.3	-10 980.0	-10 875.4	-10 874.9	-10 874.4	-10 874.8	-10 874.7
200_15_01	-11 400.0	-11 354.5	-11 146.1	-11 121.4	-11 121.8	-11 117.7	-11 121.8
200_20_01.ta101	-11 650.7	-11 580.0	-11 337.6	-11 307.8	-11 310.9	-11 310.3	-11 311.9
Moyenne	-5 180.4	-5 156.3	-5 080.4	-5 069.4	-5 068.4	-5 066.8	-5 066.2

TABLE 36 – Résultats des climbers de type premier améliorant partiel sur les paysages FSP_{ins} , avec $\kappa \in \{2, 4, 6, 8\}$. Les résultats obtenus par FIRST (voir table 18) et \mathcal{W}_{16} (voir table 22) sont reportés pour comparaison.

Cette étude des politiques de gestion de neutralité nous a dans un premier temps confirmé l'intérêt de sélectionner des mouvements neutres tout au long de la recherche, y compris pour faciliter la découverte rapide de plus hauts optimums locaux. Cependant, le climber stochastique est basé sur la stratégie du premier améliorant ; les résultats du chapitre 7 indiquent que si cette règle pivot reste la plupart du temps plus adaptée que celle du meilleur améliorant, les caractéristiques des paysages de fitness, et donc des paysages de recherche associés à la résolution d'instances de problèmes, rendent préférable l'usage des règles du meilleur (sur les paysages lisses) ou plus souvent du moins bon améliorant (sur les paysages plus rugueux). Les versions partielles de ces règles pivot permettent de ne pas explorer systématiquement le voisinage de manière exhaustive. Nous venons de voir que ces climbers pouvaient être très simplement adaptés afin de prendre en compte la sélection de mouvements neutres tout au long de la recherche, ce qui permet d'améliorer l'efficacité des climbers stochastiques. Nous avons en outre constaté que l'ajout artificiel de neutralité dans les paysages pouvait permettre d'atteindre de meilleurs optimums locaux à partir de simples climbers stochastiques. Enfin, un algorithme certes plus complexe qu'un climber traditionnel mais basé sur ce même principe de mouvements ascendants, qui permet de régler la neutralité de manière autonome jusqu'à diversifier la recherche lorsque celle-ci

stagne, nous conforte dans l'intérêt d'étudier ces algorithmes de recherche locale *a priori* simplistes mais dont le potentiel n'a pas été, de notre point de vue, suffisamment exploité.

Conclusion générale

Obtenir une compréhension accrue des mécanismes évolutifs qui régissent certains algorithmes de résolution de problèmes difficiles représente le fil conducteur de mon activité de recherche. À ce titre, nous espérons que les travaux présentés dans ce manuscrit peuvent apporter une contribution allant dans ce sens. D'autres travaux complémentaires, en bio-informatique ou en optimisation multiobjectif, n'ont pas été développés ici mais n'en constituent pas moins une part importante de mes intérêts scientifiques, que je compte renforcer dans les prochaines années.

La première partie de ce manuscrit illustre un cheminement d'idées menant de la définition d'un modèle d'algorithme évolutif en les partitionnant dynamiquement ses individus, à l'étude de bandits spécifiques pouvant abstraire la problématique de sélection adaptative des opérateurs. Ces travaux se posent davantage comme des pistes de réflexion pour le développement de concepts algorithmiques autonomes, que comme une nouvelle proposition de métaheuristique, hyperheuristique, ou stratégie de bandit.

L'étude des stratégies d'escalade dans les paysages de fitness (seconde partie) nous a permis d'avoir une vision plus précise de l'effet de certains choix conceptuels dans les algorithmes de recherche locale. Comme nous l'avons souligné dans l'introduction, il nous semble que l'efficacité d'une métaheuristique est largement influencée par ses composants de base, à commencer par la stratégie d'intensification. Les algorithmes basiques de type amélioration itérative, premier améliorant et meilleur améliorant, semblent si simplistes en regard des mécanismes de recherche plus avancés, que les études fondamentales à leur sujet sont rares et ne rendent pas clairement état de leurs effets sur la recherche. Pourtant, nous avons indiqué que le caractère intensificateur de ces algorithmes ne réduisait pas leurs possibilités de trajectoires autant qu'on pourrait l'imaginer de prime abord. L'étude des climbers sur les paysages NK, notamment, a mis en corrélation la qualité des voisins améliorants sélectionnés avec la qualité des optimums atteints. Hormis sur les instances plus simples dont les paysages associés sont moins rugueux, nous avons montré que cette corrélation était négative, donc que la stratégie du premier améliorant permettait généralement d'atteindre des solutions de meilleure qualité, mais par extension qu'une troisième stratégie, de type moins bon améliorant, convergerait en moyenne vers des optimums encore meilleurs. Une étude similaire sur les politiques de sélection des mouvements neutres a indiqué les performances supérieures — en termes de qualité des solutions atteintes mais aussi de rapidité d'exécution — des recherches locales qui acceptent des mouvements neutres à tout moment de la recherche, plutôt qu'uniquement pour sortir des optimums locaux. Nous en avons déduit que cette neutralité pouvait être une aide

plutôt qu'un frein à l'efficacité des recherches locales, et proposons des pistes pour régler dynamiquement une incorporation de neutralité artificielle dans la fonction d'évaluation.

Au sein de mes différents travaux, que ce soit sur les paysages de fitness, les modèles en îles ou les recherches locales ensemblistes pour l'optimisation multiobjectif, j'ai pu apprécier que des modèles d'abstraction de haut niveau pour la résolution de problèmes combinatoires menaient à la conceptualisation de stratégies alternatives simples, efficaces et génériques, concernant les politiques de mouvement et de choix d'opérateurs. Les différentes études réalisées en ce sens, ainsi que l'exercice de rédaction de ce mémoire, m'ont mené à avoir aujourd'hui une meilleure vue des leviers potentiels permettant de résoudre plus efficacement les problèmes combinatoires. Aboutir à ce que des travaux initialement nés d'idées et de concepts bien distincts puissent mener vers une vision enrichie et unifiée de mon champ disciplinaire, marque sans doute l'achèvement d'une étape et en amorce une nouvelle, autour d'un projet de recherche structuré et original.

Tout d'abord, la finalité des travaux sur les paysages de fitness est de comprendre les dynamiques d'escalade efficaces et de les mettre en correspondance avec les stratégies de recherche imposées par les métaheuristiques traditionnelles. De prochaines études à mener incluent l'expérimentation de règles pivot moins évidentes, la mesure de l'effet des mécanismes perturbatifs, ou le morphing de paysages. Jouer sur l'espace de recherche, la relation de voisinage ou la fonction de fitness des paysages pourrait permettre d'adapter dynamiquement leur dimension, neutralité ou rugosité. Parmi les principales pistes à explorer, l'extension du domaine de définition des variables pourrait permettre de débloquent des chemins de recherche ; de la même manière que la modification dynamique de la fonction de fitness est un moyen de contrôler les niveaux de neutralité et de rugosité pour maximiser la capacité exploratoire des stratégies de recherche génériques.

Si le modèle des paysages de fitness nous aide à déterminer *comment* appliquer les opérateurs sur les individus dans les algorithmes évolutionnaires, les modèles avancés de bandits nous indiqueront *quels* opérateurs appliquer. Une première étape à ce propos sera de généraliser notre modèle de bandits à bras interconnectés afin que ce dernier puisse abstraire plus précisément des instances de problèmes. Un bandit multi-bras pourrait alors être défini à partir d'une collection de paysages de fitness — éventuellement infinie —, et les stratégies de résolution de bandits reviendraient alors à déterminer à tout moment et de manière autonome, quel paysage, et donc opérateur, est le plus pertinent. La résolution efficace et autonome des problèmes combinatoires serait alors conditionnée par la capacité à déterminer l'action statistiquement la plus pertinente sur le long terme (1) sur un bandit et (2) sur un paysage de fitness.

Références bibliographiques

- David H. Ackley. A connectionist algorithm for genetic search. *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 121–135. L. Erlbaum Associates Inc., 1985.
- Deepak Agarwal, Bee-Chung Chen, Pradheep Elango. Explore/exploit schemes for web content optimization. *Ninth IEEE International Conference on Data Mining (ICDM)*, pages 1–10. IEEE, 2009.
- Rajeev Agrawal. Sample mean based index policies with $O(\log n)$ regret for the multi-armed bandit problem. *Advances in Applied Probability*, pages 1054–1078, 1995.
- Takuyo Aita, Hidefumi Uchiyama, Tetsuya Inaoka, Motowo Nakajima, Toshio Kokubo, Yuzuru Husimi. Analysis of a local fitness landscape with a model of the rough mt. fuji-type landscape: Application to prolyl endopeptidase and thermolysin. *Biopolymers*, 54(1):64–79, 2000.
- David Aldous, Umesh Vazirani. “Go with the winners” algorithms. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 492–501. IEEE, 1994.
- Ekaterina Alekseeva, Yuri Kochetov, Alexander Plyasunov. Complexity of local search for the p -median problem. *European Journal of Operational Research*, 191(3):736–752, 2008.
- Lourdes Araujo, Juan Julian Merelo, Antonio Mora, Carlos Cotta. Genotypic differences and migration policies in an island model. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation (GECCO)*, pages 1331–1338. ACM, 2009.
- Peter Auer, Nicolo Cesa-Bianchi, Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.
- Helio J.C. Barbosa, Asla Medeiros Sá. On adaptive operator probabilities in real coded genetic algorithms. *Workshop on Advances and Trends in Artificial Intelligence for Problem Solving (SCCC)*, 2000.
- Lionel Barnett. Ruggedness and neutrality – the NK p family of fitness landscapes. *Alive VI: Sixth International Conference on Artificial Life*, pages 18–27. MIT Press, 1998.
- Lionel Barnett. Netcrawling – Optimal evolutionary search with neutral networks. *Congress on Evolutionary Computation (CEC)*, volume 1, pages 30–37. IEEE, 2001.
- Matthieu Basseur, Adrien Goëffon. Hill-climbing strategies on various landscapes: an empirical comparison. *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference (GECCO)*, pages 479–486. ACM, 2013.
- Matthieu Basseur, Adrien Goëffon. On the efficiency of worst improvement for climbing NK landscapes. *Proceeding of the sixteenth annual conference on Genetic and evolutionary computation conference (GECCO)*. ACM, 2014.
- Matthieu Basseur, Adrien Goëffon, Arnaud Liefoghe, Sébastien Verel. On set-based local search for multiobjective combinatorial optimization. *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference (GECCO)*, pages 471–478. ACM, 2013.
- Matthieu Basseur, Adrien Goëffon, Frédéric Lardeux, Frédéric Saubion, Vincent Vigneron. On the attainability of NK landscapes global optima. *Proceeding of the seventh Annual Symposium on Combinatorial Search (SOCS)*, 2014.
- William Bateson, Gregor Mendel. *Mendel’s principles of heredity*. Putnam’s, 1909.

- Roberto Battiti, Giampietro Tecchioli. The reactive tabu search. *ORSA journal on computing*, 6(2):126–140, 1994.
- Mohamed Ben-Daya, Mohammed A. Al-Fawzan. A tabu search approach for the flow shop scheduling problem. *European Journal of Operational Research*, 109(1):88–95, 1998.
- Hans-Georg Beyer, Hans-Paul Schwefel. Evolution strategies: A comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.
- Leonora Bianchi, Marco Dorigo, Luca Maria Gambardella, Walter J. Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. 8(2):239–287, 2009.
- Ilhem Boussaïd, Julien Lepagnot, Patrick Siarry. A survey on optimization metaheuristics. *Information Sciences*, 237:82–117, 2013.
- Alex Van Breedam. Comparing descent heuristics and metaheuristics for the vehicle routing problem. *Computers & Operations Research*, 28(4):289–315, 2001.
- Michael J. Brusco, Hans-Friedrich Köhn. Clustering qualitative data based on binary equivalence relations: neighborhood search heuristics for the clique partitioning problem. *Psychometrika*, 74(4):685–703, 2009.
- Rainer E. Burkard, Stefan E. Karisch, Franz Rendl. QAPLIB: A quadratic assignment problem library. *Journal of Global Optimization*, 10(4):391–403, 1997.
- Edmund K. Burke, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan, John R. Woodward. A classification of hyper-heuristic approaches. *Handbook of metaheuristics*, pages 449–468. Springer, 2010.
- Robert Paul Byrne. *Global optimisation in process design*. PhD thesis, University of London, 1997.
- Caner Candan, Adrien Goëffon, Frédéric Lardeux, Frédéric Saubion. A dynamic island model for adaptive operator selection. *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference (GECCO)*, pages 1253–1260. ACM, 2012.
- Caner Candan, Adrien Goëffon, Frédéric Lardeux, Frédéric Saubion. Non stationary operator selection with island models. *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference (GECCO)*, pages 1509–1516. ACM, 2013.
- Erick Cantú-Paz. Migration policies, selection pressure, and parallel evolutionary algorithms. *Journal of Heuristics*, 7(4):311–334, 2001.
- Erick Cantú-Paz, David E. Goldberg. Are multiple runs of genetic algorithms better than one? *Genetic and Evolutionary Computation (GECCO)*, pages 801–812. Springer, 2003.
- Vladimír Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1):41–51, 1985.
- Nicolo Cesa-Bianchi, Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press Cambridge, 2006.
- Irène Charon, Olivier Hudry. Noising methods for a clique partitioning problem. *Discrete Applied Mathematics*, 154(5):754–769, 2006.
- Chihyung Derrick Cheng, Alexander Kosorukoff. Interactive one-max problem allows to compare the performance of interactive and human-based genetic algorithms. *Genetic and Evolutionary Computation (GECCO)*, pages 983–993. Springer, 2004.
- Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.
- Fabien Chhel, Adrien Goëffon, Frédéric Lardeux, Frédéric Saubion. Minimum multiple characterization of biological data using partially defined boolean formulas. *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC)*, pages 1399–1405. ACM, 2012.
- Vasek Chvátal. Hard knapsack problems. *Operations Research*, 28:1402–1411, 1980.
- Jonathan D. Cohen, Samuel M. McClure, J. Yu Angela. Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):933–942, 2007.

- Luis Da Costa, Álvaro Fialho, Marc Schoenauer, Michèle Sebag. Adaptive operator selection with dynamic multi-armed bandits. *Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO)*, pages 913–920. ACM, 2008.
- Fabio Daolio, Sébastien Verel, Gabriela Ochoa, Marco Tomassini. Local optima networks and the performance of iterated local search. *Proceedings of the 14th international conference on Genetic and evolutionary computation conference (GECCO)*, pages 369–376. ACM, 2012.
- Lawrence Davis. Adapting operator probabilities in genetic algorithms. *International Conference on Genetic Algorithms*, pages 61–69, 1989.
- Saul G. De Amorim, Jean-Pierre Barthélemy, Celso C. Ribeiro. Clustering and clique partitioning: simulated annealing and tabu search approaches. *Journal of Classification*, 9(1):17–41, 1992.
- Kenneth A. De Jong. Parameter setting in EAs: a 30 year perspective. *Parameter Setting in Evolutionary Algorithms*, pages 1–18. Springer, 2007.
- Jörg Denzinger, Jordan Kidney. Improving migration by diversity. *Congress on Evolutionary Computation (CEC)*, pages 700–707, 2003.
- Bilel Derbel, Sébastien Verel. DAMS: distributed adaptive metaheuristic selection. *Proceedings of the 13th annual conference on Genetic and evolutionary computation (GECCO)*, pages 1955–1962. ACM, 2011.
- Marco Dorigo. *Optimization, learning and natural algorithms*. PhD thesis, Politecnico di Milano, 1992.
- Marco Dorigo, Mauro Birattari. Ant colony optimization. *Encyclopedia of Machine Learning*, pages 36–39. Springer, 2010.
- Stefan Droste, Thomas Jansen, Ingo Wegener. On the analysis of the $(1+1)$ evolutionary algorithm. *Theoretical Computer Science*, 276(1):51–81, 2002.
- Agoston E. Eiben, Selmar K. Smit. Evolutionary algorithm parameters and methods to tune them. *Autonomous Search*, pages 15–36. Springer, 2012.
- Agoston E. Eiben, James E. Smith. *Introduction to evolutionary computing*. Springer, 2003.
- Agoston E. Eiben, Zbigniew Michalewicz, Marc Schoenauer, James E. Smith. Parameter control in evolutionary algorithms. *Parameter setting in evolutionary algorithms*, pages 19–46. Springer, 2007.
- Taisir Eldos. A new migration model for the distributed genetic algorithms. *Proceedings of the International Conference on Scientific Computing (CSC)*, pages 26–29, 2006.
- Paul Erdős, Alfréd Rényi. On a classical problem of probability theory. *Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, 6:215–220, 1961.
- Álvaro Fialho, Luis Da Costa, Marc Schoenauer, Michèle Sebag. Extreme value based adaptive operator selection. *Parallel Problem Solving from Nature (PPSN X)*, pages 175–184. Springer, 2008.
- Álvaro Fialho, Luis Da Costa, Marc Schoenauer, Michèle Sebag. Dynamic multi-armed bandits and extreme value-based rewards for adaptive operator selection in evolutionary algorithms. *Learning and Intelligent Optimization (LION 3)*, pages 176–190. Springer, 2009.
- Álvaro Fialho, Luis Da Costa, Marc Schoenauer, Michèle Sebag. Analyzing bandit-based adaptive operator selection mechanisms. *Annals of Mathematics and Artificial Intelligence*, 60(1-2):25–64, 2010.
- Aurélien Garivier, Éric Moulines. On upper-confidence bound policies for switching bandit problems. *Algorithmic Learning Theory*, pages 174–188. Springer, 2011.
- Josselin Garnier, Leila Kallel. Efficiency of local search with multiple local optima. *SIAM Journal on Discrete Mathematics*, 15(1):122–141, 2001.
- Ian P. Gent, Holger H. Hoos, Patrick Prosser, Toby Walsh. Morphing: Combining structure and randomness. *Aaai/Iaai*, 99:654–660, 1999.
- Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5), 1986.

- Adrien Goëffon. *Nouvelles heuristiques de voisinage et mémétiques pour le problème Maximum de Parcimonie*. PhD thesis, Université d'Angers, 2006.
- Adrien Goëffon, Frédéric Lardeux. Optimal one-max strategy with dynamic island models. *23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 485–488. IEEE, 2011a.
- Adrien Goëffon, Frédéric Lardeux. Modèles en îles dynamiques. *Septièmes Journées Francophones de Programmation Par Contraintes (JFPC)*, 2011b.
- Adrien Goëffon, Frédéric Lardeux. Autonomous local search algorithms with island representation. *Learning and Intelligent Optimization (LION 6)*, pages 390–395. Springer, 2012.
- Adrien Goëffon, Macha Nikolski, David J. Sherman. An efficient probabilistic population-based descent for the median genome problem. *Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO)*, pages 315–322. ACM, 2008a.
- Adrien Goëffon, Jean-Michel Richer, Jin-Kao Hao. Progressive tree neighborhood applied to the maximum parsimony problem. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(1):136–145, 2008b.
- Adrien Goëffon, Frédéric Lardeux, Frédéric Saubion. Simulating non stationary operators in search algorithms. 2014.
- David E. Goldberg. Probability matching, the magnitude of reinforcement, and classifier system bidding. *Machine Learning*, 5(4):407–425, 1990.
- David E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.
- David E. Goldberg, Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, 1991.
- David E. Goldberg, Kalyanmoy Deb, James H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362, 1991.
- David E. Goldberg, Kalyanmoy Deb, Jeffrey Horn. Massive multimodality, deception, and genetic algorithms. *Parallel Problem Solving from Nature (PPSN II)*, pages 37–45. Elsevier, 1992.
- Ole-Christoffer Granmo, Stian Berg. Solving non-stationary bandit problems by random sampling from sibling kalman filters. *Trends in Applied Intelligent Systems*, pages 199–208. Springer, 2010.
- Steven Gustafson, Edmund K. Burke. The speciating island model: An alternative parallel evolutionary algorithm. *Journal of Parallel and Distributed Computing*, 66(8):1025–1036, 2006.
- Youssef Hamadi, Éric Monfroy, Frédéric Saubion. *Autonomous Search*. Springer, 2011.
- Pierre Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. *Congress on numerical methods in combinatorial optimization*, pages 70–145, 1986.
- Pierre Hansen, Nenad Mladenović. First vs. best improvement: An empirical study. *Discrete Applied Mathematics*, 154(5):802–817, 2006.
- Jin-Kao Hao, Christine Solnon. Méta-heuristiques et intelligence artificielle. *Panorama de l'intelligence artificielle, volume 2 : Algorithmes pour l'I.A.* Cepadues, 2014.
- Jin-Kao Hao, Philippe Galinier, Michel Habib. Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. *Revue d'Intelligence Artificielle*, 13(2):283–324, 1999.
- George Harik, Erick Cantú-Paz, David E. Goldberg, Brad L. Miller. The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Evolutionary Computation*, 7(3):231–253, 1999.
- Ulrich Herkenrath. The N-armed bandit with unimodal structure. *Metrika*, 30(1):195–210, 1983.
- David V. Hinkley. Inference about the change-point in a sequence of random variables. *Biometrika*, 57(1):1–17, 1970.
- Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

- John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- Holger H. Hoos. Automated algorithm configuration and parameter tuning. *Autonomous Search*, pages 37–71. Springer, 2012.
- Holger H. Hoos, Thomas Stützle. *Stochastic Local Search: Foundations & Applications*. Morgan Kaufmann, 2004.
- Holger H. Hoos, Thomas Stützle. SATLIB: An Online Resource for Research on SAT. *SAT 2000*, pages 283–292. IOS Press, 2000.
- Frank Hutter, Holger H. Hoos, Kevin Leyton-Brown, Thomas Stützle. ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306, 2009.
- Thomas Jansen. *Analyzing Evolutionary Algorithms*. Springer, 2013.
- Angel A. Juan, Helena R. Lourenço, Manuel Mateo, Rachel Luo, Quim Castella. Using iterated local search for solving the flow-shop problem: Parallelization, parametrization, and randomization issues. *International Transactions in Operational Research*, 2013.
- Bryant A. Julstrom. What have you done for me lately? Adapting operator probabilities in a steady-state genetic algorithm. *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 81–87. Morgan Kaufmann, 1995.
- Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
- Stuart A. Kauffman, Simon Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128(1):11–45, 1987.
- Stuart A. Kauffman, Edward D. Weinberger. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of theoretical biology*, 141(2):211–245, 1989.
- Frank P. Kelly. Multi-armed bandits with discount factor near one: The Bernoulli case. *The Annals of Statistics*, 9(5):987–1001, 1981.
- Rafal Kicinger, Tomasz Arciszewski, Kenneth De Jong. Evolutionary computation and structural design: A survey of the state-of-the-art. *Computers & Structures*, 83(23):1943–1978, 2005.
- John F.C. Kingman. A simple model for the balance between selection and mutation. *Journal of Applied Probability*, pages 1–12, 1978.
- Scott Kirkpatrick, C. Daniel Gelatt, Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- Dimitris E. Koulouriotis, A. Xanthopoulos. Reinforcement learning and evolutionary algorithms for non-stationary multi-armed bandit problems. *Applied Mathematics and Computation*, 196(2):913–922, 2008.
- Tze Leung Lai, Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.
- Ailsa H. Land, Alison G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28:497–520, 1960.
- John Langford, Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems*, pages 817–824, 2008.
- Frédéric Lardeux, Adrien Goëffon. A dynamic island-based genetic algorithms framework. *Simulated Evolution and Learning (SEAL)*, pages 156–165. Springer, 2010.
- Pedro Larrañaga, Jose A. Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer, 2002.
- Jan Karel Lenstra, Alexander Rinnooy Kan, Peter Brucker. Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1:343–362, 1977.

- Kevin Leyton-Brown, Eugene Nudelman, Galen Andrew, Jim McFadden, Yoav Shoham. A portfolio approach to algorithm selection. *Proceedings of the 18th international joint conference on Artificial intelligence (IJCAI)*, pages 1542–1543. Morgan Kaufmann, 2003.
- Lihong Li, Wei Chu, John Langford, Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.
- Yun-En Liu, Travis Mandel, Emma Brunskill, Zoran Popovic. Trading off scientific knowledge and user learning with multi-armed bandits. *Proceedings of the 7th International Conference on Educational Data Mining. IEDMS*, 2014.
- Rodolfo Ayala Lopes, Rodrigo C. Pedrosa Silva, Alan R.R. Freitas, Felipe Campelo, Frederico G. Guimarães. A study on the configuration of migratory flows in island model differential evolution. *Proceedings of the 16th annual conference on Genetic and evolutionary computation (GECCO)*, pages 1015–1022. ACM, 2014.
- Helena R. Lourenço, Olivier C. Martin, Thomas Stützle. *Iterated local search*. Springer, 2003.
- Guangzhou Lu, Jinlong Li, Xin Yao. Fitness-probability cloud and a measure of problem hardness for evolutionary algorithms. *Evolutionary Computation in Combinatorial Optimization*, pages 108–117. Springer, 2011.
- R. Duncan Luce. *Individual Choice Behavior a Theoretical Analysis*. Wiley, 1959.
- Katherine M. Malan, Andries P. Engelbrecht. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences*, 241:148–163, 2013.
- Narine Manukyan, Margaret J. Eppstein, Jeffrey S. Buzas. NM landscapes: beyond NK. *Proceedings of the 16th annual conference on Genetic and evolutionary computation (GECCO)*, pages 203–204. ACM, 2014.
- Marie-Éléonore Marmion, Clarisse Dhaenens, Laetitia Jourdan, Arnaud Liefoghe, Sébastien Verel. The road to VEGAS: Guiding the search over neutral networks. *Proceedings of the 13th annual conference on Genetic and evolutionary computation (GECCO)*, pages 1979–1986. ACM, 2011a.
- Marie-Éléonore Marmion, Clarisse Dhaenens, Laetitia Jourdan, Arnaud Liefoghe, Sébastien Verel. NILS: a neutrality-based iterated local search and its application to flowshop scheduling. *Evolutionary Computation in Combinatorial Optimization*, pages 191–202. Springer, 2011b.
- Marie-Éléonore Marmion, Aymeric Blot, Laetitia Jourdan, Clarisse Dhaenens. Neutrality in the graph coloring problem. *Learning and Intelligent Optimization (LION 7)*, pages 125–130. Springer, 2013.
- Silvano Martello, David Pisinger, Paolo Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science*, 45(3):414–424, 1999.
- Jorge Maturana, Álvaro Fialho, Frédéric Saubion, Marc Schoenauer, Michèle Sebag. Extreme compass and dynamic multi-armed bandits for adaptive operator selection. *Congress on Evolutionary Computation (CEC)*, pages 365–372. IEEE, 2009.
- Peter Merz, Bernd Freisleben. Fitness landscapes and memetic algorithm design. *New ideas in optimization*, pages 245–260, 1999.
- Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- Heinz Mühlenbein. How genetic algorithms really work. Mutation and hill-climbing. *Parallel Problem Solving from Nature (PPSN II)*, pages 15–25, 1992.
- Zbigniew Michalewicz. *Genetic algorithms + data structures = evolution programs*. Springer, 1996.
- Melanie Mitchell, Stephanie Forrest, John H. Holland. The royal road for genetic algorithms: Fitness landscapes and GA performance. *Proceedings of the first european conference on artificial life*, pages 245–254. MIT Press, 1992.
- Melanie Mitchell, John H. Holland, Stephanie Forrest. When will a genetic algorithm outperform hill climbing? *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann, 1993.

- Nejad Mladenović, Pierre Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.
- Pablo Moscato. Memetic algorithms: a short introduction. pages 219–234, 1999.
- Volker Nannen, Selmar K. Smit, Agoston E. Eiben. Costs and benefits of tuning parameters of evolutionary algorithms. *Parallel Problem Solving from Nature (PPSN X)*, pages 528–538. Springer, 2008.
- Mark E.J. Newman, Robin Engelhardt. Effects of selective neutrality on the evolution of molecular species. *Proceedings of the Royal Society B*, 265(1403):1333–1338, 1998.
- Edgar Noda, André L.V. Coelho, Ivan L.M. Ricarte, Akebo Yamakami, Alex A. Freitas. Devising adaptive migration policies for cooperative distributed genetic algorithms. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2002.
- Eugeniusz Nowicki, Czesław Smutnicki. A fast tabu search algorithm for the permutation flow-shop problem. *European Journal of Operational Research*, 91(1):160–175, 1996.
- Gabriela Ochoa, Sébastien Verel, Marco Tomassini. First-improvement vs. best-improvement local optima networks of NK landscapes. *Parallel Problem Solving from Nature (PPSN XI)*, pages 104–113. Springer, 2010.
- Ronald Ortner, Daniil Ryabko, Peter Auer, Rémi Munos. Regret bounds for restless markov bandits. *Algorithmic Learning Theory*, pages 214–228. Springer, 2012.
- Ewan S. Page. Continuous inspection schemes. *Biometrika*, pages 100–115, 1954.
- Gintaras Palubeckis, Armantas Ostreika, Arūnas Tomkevičius. An iterated tabu search approach for the clique partitioning problem. *The Scientific World Journal*, 2014.
- Sandeep Pandey, Deepayan Chakrabarti, Deepak Agarwal. Multi-armed bandit problems with dependent arms. *Proceedings of the 24th international conference on Machine learning*, pages 721–728. ACM, 2007.
- Christos H. Papadimitriou, Kenneth Steiglitz. *Combinatorial Optimization : Algorithms and Complexity*. Dover Publications, 1998.
- Panos M. Pardalos, Henry Wolkowicz. *Quadratic Assignment and Related Problems: Dimacs Workshop May 20-21, 1993*, volume 16. American Mathematical Society, 1994.
- Marc Pirlot. General local search methods. *European journal of operational research*, 92(3):493–511, 1996.
- Frank J. Poelwijk, Daniel J. Kiviet, Daniel M. Weinreich, Sander J. Tans. Empirical fitness landscapes reveal accessible evolutionary paths. *Nature*, 445(7126):383–386, 2007.
- Alain Ratle. Kriging as a surrogate fitness landscape in evolutionary optimization. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 15(01):37–49, 2001.
- Ingo Rechenberg. Cybernetic solution path of an experimental problem. Technical report, Royal Air Force Establishment, 1965.
- Christian M. Reidys, Peter F. Stadler. Neutrality in fitness landscapes. *Applied Mathematics and Computation*, 117:187–207, 1998.
- J. Neal Richter. *On Mutation and Crossover in the Theory of Evolutionary Algorithms*. PhD thesis, Montana State University, 2010.
- Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.
- Leiba Rodman. On the many-armed bandit problem. *The Annals of Probability*, pages 491–498, 1978.
- Marek Ruciński, Dario Izzo, Francesco Biscani. On the impact of the migration topology on the island model. *Parallel Computing*, 36(10):555–571, 2010.
- Rubén Ruiz, Concepción Maroto. A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, 165(2):479–494, 2005.

- Rubén Ruiz, Thomas Stützle. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033–2049, 2007.
- Sartaj Sahni, Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, 1976.
- Steven L. Scott. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.
- Benjamin Skellett, Benjamin Cairns, Nicholas Geard, Bradley Tonkes, Janet Wiles. Maximally rugged NK landscapes contain the highest peaks. *Genetic and Evolutionary Computation (GECCO)*, pages 579–584. ACM, 2005.
- Zbigniew Skolicki. Linkage in island models. *Linkage in Evolutionary Computation*, pages 41–60. Springer, 2008.
- Zbigniew Skolicki, Kenneth A. De Jong. The influence of migration sizes and intervals on island models. *Genetic and Evolutionary Computation (GECCO)*, pages 1295–1302. ACM, 2005.
- Robert Elliott Smith, Jim Smith. New methods for tunable, random landscapes. *Foundations of Genetic Algorithms*, 6:47–67, 2001.
- Tom Smith, Phil Husbands, Paul J. Layzell, Michael O’Shea. Fitness landscapes and evolvability. *Evolutionary Computation*, 10:1–34, 2002.
- Kenneth Sörensen. Metaheuristics – the metaphor exposed. *International Transactions in Operational Research*, 2013.
- Jean-Luc Souciet, Bernard Dujon, Claude Gaillardin, Mark Johnston, Philippe V. Baret, Paul Cliften, David J. Sherman, Jean Weissenbach, Eric Westhof, Patrick Wincker *et al.* Comparative genomics of protoplloid saccharomycetaceae. *Genome research*, 19(10):1696–1709, 2009.
- William M. Spears. *Evolutionary algorithms: the role of mutation and recombination*. Springer, 2000.
- Mandavilli Srinivas, Lalit M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 24(4):656–667, 1994.
- Peter F. Stadler. Towards a theory of landscapes. *Complex systems and binary networks*, pages 78–163. Springer, 1995.
- Thomas Stützle, Irina Dumitrescu. Topics in local search. *MN Summerschool, Tenerife*, 2003.
- Richard S. Sutton, Andrew G Barto. *Introduction to reinforcement learning*. MIT Press, 1998.
- Éric Taillard. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64(2):278–285, 1993.
- Éric Taillard. Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 47(1):65–74, 1990.
- El-Ghazali Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8(5):541–564, 2002.
- Reiko Tanese. Distributed genetic algorithms. *Proceedings of the third international conference on Genetic algorithms*, pages 434–439. Morgan Kaufmann, 1989.
- Mandayam A.L. Thathachar, P. Shanti Sastry. A class of rapidly converging algorithms for learning automata. *IEEE Transactions on Systems, Man and Cybernetics*, 15:168–175, 1985.
- Dirk Thierens. An adaptive pursuit strategy for allocating operator probabilities. *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1539–1546. ACM, 2005.
- William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294, 1933.
- Michel Tokic, Günther Palm. Value-difference based exploration: adaptive control between epsilon-greedy and softmax. *KI – Künstliche Intelligenz*, pages 335–346. Springer, 2011.
- Marco Tomassini, Sébastien Verel, Gabriela Ochoa. Complex-network analysis of combinatorial spaces: The NK landscape case. *Physical Review E*, 78(6):066114, 2008.

- Andrew Tuson, Peter Ross. Adapting operator settings in genetic algorithms. *Evolutionary computation*, 6(2): 161–184, 1998.
- Nico L.J. Ulder, Emile H.L. Aarts, Hans-Jürgen Bandelt, Peter J.M. van Laarhoven, Erwin Pesch. Genetic local search algorithms for the traveling salesman problem. *Parallel problem solving from nature (PPSN I)*, pages 109–116. Springer, 1991.
- Rasmus K. Ursem. Diversity-guided evolutionary algorithms. *Parallel Problem Solving from Nature (PPSN VII)*, pages 462–471. Springer, 2002.
- Leonardo Vanneschi, Yuri Pirola, Philippe Collard. A quantitative study of neutrality in GP boolean landscapes. *Genetic and Evolutionary Computation Conference (GECCO)*, pages 895–902. ACM, 2006.
- Sébastien Verel, Philippe Collard, Manuel Clergue. Scuba search: when selection meets innovation. *Congress on Evolutionary Computation (CEC)*, volume 1, pages 924–931. IEEE, 2004.
- Sébastien Verel, Gabriela Ochoa, Marco Tomassini. The connectivity of NK landscapes’ basins: A network analysis. *Proceedings of the 11th international conference on the Simulation and Synthesis of Living Systems*, pages 648–655, 2008.
- Sébastien Verel, Gabriela Ochoa, Marco Tomassini. Local optima networks of NK landscapes with neutrality. *IEEE Transactions on Evolutionary Computation*, 15(6):783–797, 2011.
- Nikolay Vyahhi, Adrien Goëffon, Macha Nikolski, David J. Sherman. Swarming along the evolutionary branches sheds light on genome rearrangement scenarios. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation (GECCO)*, pages 241–246. ACM, 2009.
- Chih-Chun Wang, Sanjeev R. Kulkarni, H. Vincent Poor. Bandit problems with side observations. *IEEE Transactions on Automatic Control*, 50(3):338–355, 2005.
- Christopher J.C.H. Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge, 1989.
- Christopher J.C.H. Watkins, Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.
- Edward Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological cybernetics*, 63(5):325–336, 1990.
- Darrell Whitley, Joan Kauth. *GENITOR: A different genetic algorithm*. Colorado State University, Department of Computer Science, 1988.
- Darrell Whitley, Soraya Rana, Robert B. Heckendorn. The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7:33–47, 1998.
- Darrell Whitley, Adele E. Howe, Doug Hains. Greedy or not? Best improving versus first improving stochastic local search for MAXSAT. *Proceedings of 27th Conference on Artificial Intelligence*, pages 940–946. AAAI, 2013.
- Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, pages 287–298, 1988.
- David H. Wolpert, William G. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, 1997.
- Sewall Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. *Proceedings of the VI International Congress of Genetics*, 1:356–366, 1932.
- Shengxiang Yang. Non-stationary problem optimization using the primal-dual genetic algorithm. *Congress on Evolutionary Computation (CEC)*, volume 3, pages 2246–2253. IEEE, 2003.
- Mihalis Yannakakis. The analysis of local search problems and their heuristics. *Proceedings of the 7th annual symposium on Theoretical aspects of computer science (STACS)*, pages 298–311. Springer, 1990.
- Jia Yuan Yu, Shie Mannor. Piecewise-stationary bandit problems with side observations. *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 1177–1184. ACM, 2009.
- Hantao Zhang, Mark Stickel. Implementing the Davis–Putnam method. *Journal of Automated Reasoning*, 24(1-2): 277–296, 2000.