# Université Mohamed Boudiaf - M'sila

## FACULTE DE TECHNOLOGIE

## DEPARTEMENT D'ELECTRONIQUE

Numéro de série…….

Numéro d'inscription…….

# Thèse

Présentée pour l'obtention du diplôme de

DOCTORAT SCIENCE

**Spécialité** : Electronique
**Option** : Electronique

## THEME

# Etude et Optimisation des Codes Pseudo-Aléatoires pour les Applications à Accès Multiple


**Présentée par**
**BOUKERMA Sabrina**


Soutenue le 15/07/2021

**Devant le jury composé de :**

| Nom & Prénom | Grade | Etablissement | Qualité |
|---|---|---|---|
| SAIGAA Djamel | Professeur | Université de Msila | Président |
| ROUABAH Khaled | Professeur | Université de BBA | Encadreur |
| LADJAL Mohamed | MCA | Université de Msila | Examinateur |
| BAKHTI Haddi | MCA | Université de Msila | Examinateur |
| FLISSI Mustapha | MCA | Université de BBA | Examinateur |
| MEZAACHE SalahEddine | MCB | Université de BBA | Invité |

**Année Universitaire : 2020/2021**

# *Remerciements*

*Je dédicace tout d'abord mes plus grands remerciements à mon Directeur de thèse, Professeur ROUABAH Khaled pour m'avoir orienté et guidé afin de réaliser ce travail.*

*Merci également à Monsieur SAIGAA Djamel, Professeur à l'Université de M'sila, de m'avoir fait l'honneur de présider ce jury.*

*J'exprime aussi mes sincères remerciements aux membres de mon Jury de soutenance, Monsieur LADJAL Mohamed, Professeur à l'Université de M'sila, Monsieur FLISSI Mustapha, MCA à l'Université de BBA, Monsieur BAKHTI Haddi, MCA à l'Université de M'sila d'avoir accepté d'examiner et d'évaluer de ce travail.*

*Je remercie aussi Messieurs MEZAACHE Salah-Eddine et Salim ATIA, Maîtres de conférences à l'université de Mohamed Elbachir El-Ibrahimi- Bordj-Bou-Arreridj, pour leur aide précieuse.*

*Un gros merci à mon mari Hichem, à la plus belle fille du monde Loudjain, à mon ange anas, à ma mère qui est ma fierté et ma raison de vivre, mes sœurs, mes frères, toute ma famille et la famille de mon mari pour leur soutien et leur confiance. Ils ont continuellement cru en moi et m'ont toujours repoussé à aller vers l'avant. Sans leur soutien et leur amour, ce travail n'aurait en aucun cas pu être réalisé. Je tiens à vous dire à tous « combien je vous aime ».*

*Mes derniers remerciements vont à mes collègues, mes enseignants et tout le personnel de la direction du commerce de la wilaya de bordj Bou Arreridj surtout Messieurs OUCHEN Farouk, ZOUAOUI Khalil, SAYEDJ Ahmed et Madame ZEDAME Samra, ainsi que toutes les personnes qui m'ont apporté un soutien moral de loin ou de près.*

# Résumé

Les travaux de recherche présentés dans cette thèse ont pour but l'étude et l'optimisation des séquences longues utilisées dans les systèmes d'accès multiple. Initialement, nous réalisons une étude des différentes méthodes de génération des séquences pseudo aléatoires PN (Pseudo-noise) utilisées dans les systèmes DS-SS (Direct Sequence Spread Spectrum). De plus, nous effectuons une comparaison des séquences d'étalement étudiées en s'appuyant plus particulièrement sur les propriétés de la corrélation. Ensuite, nous présentons le principe et les résultats de l'application de notre méthode proposée pour générer deux types de longues séquences d'étalement binaires optimisées OLBSS (optimized long binary spreading sequences) avec de bonnes propriétés de la fonction d'autocorrélation (FAC). Ici, le premier type proposé est construit à partir de sous-séquences binaires concaténées appartenant à la même famille de codes, telles que les sous-séquences de Walsh-Hadamard et les sous-séquences de Gold, ayants de bonnes propriétés de la fonction d'inter-corrélation (FIC). Le deuxième type utilise les mêmes sous-séquences mais qui sont plutôt entrelacées. Ici, le nombre et les tailles des sous-séquences sont liés à la taille choisie de la longue séquence finale construite et aux performances souhaitées. L'optimisation des OLBSS est réalisée à l'aide de deux techniques d'optimisation, à savoir les algorithmes génétiques (AGs) et l'optimisation par essaims de particules (OEP). Les résultats de simulations obtenus, basés sur l'outil Matlab, ont montré que les nouvelles OLBSS composées de sous-séquences de Walsh-Hadamard ou de Gold ont de bonnes propriétés de la FAC en comparaison avec les séquences aléatoires et celles de Gold originales de même longueur. De plus, l'étude comparative a montré que les OLBSS, obtenues en utilisant l'algorithme d'optimisation AG et les sous-séquences de Walsh-Hadamard entrelacées, présentent les meilleures performances.

**Mots clés :** Séquences pseudo aléatoires PN, les systèmes DS-SS, Longues séquences d'étalement binaires optimisées, Fonction d'autocorrélation (FAC), Fonction d'inter-corrélation (FIC), Walsh-Hadamard, Gold, algorithmes génétiques (AGs), l'optimisation par essaims de particules (OEP), Entrelacement, Concaténation.

# Table des matières

# Liste des figures

# Liste des tableaux

# Liste des abréviations

| | |
|---|---|
| **ABC** | Artificial Bee Colony |
| **A-CDMA** | Code Division Multiple Access asynchrone |
| **ACO** | Ant Colony Optimization : Optimisation par colonies de fourmis |
| **ACS** | Ant Colony System |
| **AE** | Algorithme évolutionnaire |
| **AGs** | Algorithmes Génétiques |
| **BF** | Bacteria Foraging |
| **CAN** | Cyclic Algorithm New |
| **CD** | Coordinate descent: la méthode de descente par coordonnée |
| **CDMA** | Code Division Multiple Access : Accès multiple par répartition de codes |
| **DBS** | De bruijn séquence : séquence de De Bruijn |
| **DSP** | Densité Spectrale de Puissance |
| **DS-SS** | Direct Sequence Spread Spectrum : Etalement de spéctre par séquences directes |
| **FAC** | Autocorrélation Function : La fonction d'Autocorrélation |
| **FDMA** | Frequency Division Multiple Access : Accès multiple par répartition de fréquence |
| **FHSS** | Frequency Hopping Spread Spectrum: Etalement par saut de fréquence |
| **FIC** | Cross-corrélation Function : La fonction d'Inter-corrélation |
| **FM** | Merit Factor: Le facteur de mérite |
| **GNSS** | Global Navigation Satellite System |
| **GPS** | Global Positioning System |
| **HC** | Hill Climbing |
| **IAM** | Multiple access interference : Interférences d'Accès Multiple |
| **IS-95** | Interim Standard 95 |
| **ISL** | Integrated Sidelobe Level: Le niveau des pics secondaires intégrés |
| **LABS** | Low autocorrelation binary sequences: Séquences binaires à faible autocorrélation |
| **LFSR** | Linear Feedback Shift Register: Registre à décalage à rétroaction linéaire |
| **MIMO** | Multiple-Input Multiple-Output: Entrées multiples, sorties multiples |
| **MOCS** | Mutually Orthogonal Complementary Set |
| **OEP** | Particle Swarm Optimization : Optimisation par Essaim de Particules |
| **OLBSS** | Optimized Long Binary Spreading Sequences |
| **OVSF** | Orthogonal Variable Spreading Factor: Code orthogonal a facteur d'étalement |
| **PE** | Evolutionary Programming : Programmation évolutionnaire |
| **PeCAN** | Periodic-correlation CAN |
| **PG** | Genetic Programming : Programmation génétique |
| **PN** | Pseudo-Noise sequences : Séquences pseudo aléatoires |
| **PSL** | Peak Sidelobe Level: Le niveau des pics secondaires |
| **RMSE** | Root Mean Square Error : Racine carrée de l'erreur quadratique moyenne |
| **SDS** | Stochastic Diffusion Search System |
| **S-CDMA** | CDMA synchrone |
| **SE** | Evolution strategie : Stratégies d'évolution |
| **SF** | Spreading Factor: Facteur d'étalement |
| **SRW** | Roulette Wheel Selection : Sélection par roulette Wheel |
| **TDMA** | Time Division Multiple Access : Accès multiple par répartition temporelle |
| **TD-SCDMA** | Time Division Synchronous Code Division Multiple Access |

| | |
|---|---|
| **TEB** | Taux d'Erreur Binaire |
| **THSS** | Time Hopping Spread Spectrum: Etalement par saut d'intervalle de temps |
| **UMTS** | Universal Mobile Telecommunication System |
| **UMTS-UTRA** | Universal Mobile Telecommunications System - Universal Terrestrial Radio Access |
| **W-CDMA** | Wideband– Code Division Multiple Access. |
| **WISL** | ISL pondérée |
| **ZCZ** | Zero Correlation Zone. |

# Introduction générale

L'objectif principal des systèmes de communication a toujours été de permettre l'échange d'informations à distance avec la meilleure qualité et le maximum d'efficacité possibles. Cet objectif est en fait réalisé à travers diverses techniques de traitement des différents signaux utilisés dans les processus de transmission et de réception à savoir le filtrage, la numérisation, la modulation/démodulation, le multiplexage, l'accès multiple, etc. Ces différentes techniques sont désormais obligées de faire face, continuellement, aux problèmes de la restriction des ressources spectrales afin d'accommoder, d'un côté, le nombre d'utilisateurs qui ne cesse de croitre et de répondre, d'un autre côté, aux exigences de plus en plus proclamées en termes d'efficacité, de précision et de résistance aux interférences, au bruit et aux multitrajets. En conséquence, les recherches se sont focalisées conjointement sur l'optimisation de toutes ces techniques de traitement afin de réaliser les meilleures performances globalement désirées. L'évolution de la technique de modulation, par exemple, a parcourue un long chemin quant à l'élaboration de nouvelles méthodes nettement plus adaptées et plus efficaces permettant de résoudre le problème d'interopérabilité dans divers systèmes de communication plus particulièrement les systèmes mondiaux de navigation par satellites.

Notre contribution majeure est plutôt liée à l'amélioration de la technique d'accès multiple par répartition de code CDMA, basée sur l'étalement du spectre par séquence directe DSSS, d'une part, en présence des difficultés liées au bruit, aux interférences et aux multitrajets et d'autre part vis-à-vis des problèmes de précision de compatibilités qui se présente comme la plus intéressante parmi les différentes techniques existantes [1]. En effet, elle n'impose aucune contrainte de temps ni de fréquence pour l'accès au canal de transmission.

Le principe général de la DS-SS consiste en le codage du signal informatif avec une par une séquence PN, connue seulement par les systèmes d'émission et de réception. Le résultat de cette opération est la répartition de la densité spectrale de puissance du signal informatif sur une large bande passante [2-3].

A la réception, en multipliant le signal reçu par le même code PN utilisé par l'émetteur, on peut facilement récupérer les données transmises. C'est ainsi que l'influence du code PN est éliminée.

Les caractéristiques des codes, utilisés par la technique CDMA, jouent un rôle déterminant dans la qualité de ses performances. En effet, ces caractéristiques ont une influence directe sur l'immunité des systèmes de communication vis-à-vis des problèmes susmentionnés. C'est justement dans ce cadre que s'inscrit ce travail de Doctorat. Le but est donc la conception de nouvelles séquences de codes qui disposent de caractéristiques permettant les meilleurs avantages pour les systèmes de télécommunication à base de CDMA. Ces caractéristiques ont été déjà établies dans la littérature scientifique comme suit : dans un premier temps le nombre $M$ de ces séquences doit être élevé pour autoriser l'accès au maximum d'usagers. Deuxièmement, ces séquences doivent posséder de bonnes propriétés de la FAC pour permettre une meilleure synchronisation des séquences en réception et assurer par conséquent la robustesse contre le bruit [4]. Troisièmement, elles doivent présenter des valeurs des FICs les plus faibles possibles pour éliminer les interférences d'accès multiple (IAM) dues à la transmission simultanée de plusieurs utilisateurs [5]. Enfin, les codes obtenus doivent être longs pour permettre un meilleur étalement du spectre et pour assurer la sécurité du message. Comme il est très difficile d'obtenir un générateur de tels codes ayant toutes ces caractéristiques, on s'intéresse dans ce travail à la dernière caractéristique qui représente en effet la plus importante pour les communications par satellites à base de technique DS-SS.

Dans la littérature scientifique, les séquences d'étalement, les plus couramment utilisées dans les systèmes de communication, sont : les m-séquences [6-7], les séquences de Gold [8], les séquences de Kasami [9], les séquences de Weil [10], les séquences de Barker [11], les séquences ZCZ (Zero Correlation Zone) [12] et les séquences de De Bruijn [13]. Les m séquences, les séquences de Gold et les séquences de Kasami sont bien connues comme des familles de séquences binaires avec de bonnes propriétés de la FIC [6]. Elles sont toutes générées à partir du registre à décalage à rétroaction linéaire LFSR (Linear-Feedback Shift Register) [14]. Les séquences de Gold, appartenant à la famille la plus connue des m-séquences, sont utilisées dans les systèmes de navigation par satellites GNSS (Global Navigation Satellite System) tels que l'américain GPS (Global Positioning System) et l'européen Galileo, [15].

Les signaux GNSS modernisés vont utiliser de nouvelles structures de codes PN qui ne sont pas basées sur des générateurs de type LFSR. Les codes de Weil, basés sur les séquences de Legendre, vont être utilisés dans certaines applications GPS nouvelle génération, en particulier dans la bande L1C [16]. Tandis que les signaux Galileo E1 OS vont mettre en œuvre des codes aléatoires appelés aussi codes mémoires. Ces derniers sont des codes optimisés pour être aussi aléatoires que possible [16].

L'ensemble des codes dont la FIC est nulle pour tout décalage temporel sont dits séquences orthogonales [17]. On peut citer comme exemples les séquences de Walsh-Hadamard [18], les séquences de Gold orthogonales, les séquences orthogonales de longueur variable OVSF (Orthogonal Variable Spreading Factor) [17, 19], et les séquences de Golay complémentaires [20].

Les séquences orthogonales ont été d'abord introduites pour les systèmes de communication 3G tels que : les normes UMTS-UTRA (Universal Mobile Telecommunications System - Universal Terrestrial Radio Access) [21], W-CDMA (Wide CDMA) [22] et TD-SCDMA (Time Division Synchronous Code Division Multiple Access) [23] qui utilisent les séquences OVSF. Les séquences de Walsh-Hadamard ont de nombreuses applications populaires dans les normes de communication sans fil actuelles telles que : l'IS-95 (Interim Standard 95) [24] et la CDMA2000 [25].

D'autres séquences d'étalements sont moins connues mais elles existent. On peut donner comme exemples : les séquences de Bent [26], les séquences de No [27], les séquences GMW (Gordon, Mills, and Welch) [28] et les séquences ZC (Zadoff–Chu) [29].

Bien qu'il soit très difficile de construire une famille de séquences qui rassemblerait toutes les caractéristiques citées précédemment, notre contribution s'articule essentiellement sur la proposition de deux types d'OLBSS résultant de la concaténation / l'entrelacement de sous-séquences binaires plus courtes et présentant des propriétés améliorées de la FAC. La construction des OLBSS est optimisée à l'aide de deux techniques d'optimisation différentes, à savoir les AGs [30-31] et la méthode OEP [32-33].

Les objectifs de cette thèse sont multiples et sont donnés comme suit :

- ✓ L'implémentation des structures de génération des différentes familles des séquences d'étalement ;

- ✓ La comparaison, en termes de plusieurs critères (plus précisément la FAC et la FIC), entre les différentes séquences étudiées ;
- ✓ La conception et l'optimisation des OLBSS, proposées dans le cadre de cette thèse, à l'aide de deux techniques d'optimisation notamment les AGs et la méthode OEP ;
- ✓ L'étude des performances de notre approche proposée et sa comparaison avec les approches classiques.

Le présent manuscrit est structuré autour de trois chapitres. En effet, dans le premier chapitre on présente, en premier lieu, le principe de base de la DS-SS. De plus, on décrit le principe d'une technique, couramment utilisée dans les systèmes de communications, à savoir la CDMA. En deuxième lieu, comme l'approche développée dans ce travail consiste à rechercher de longues séquences optimales avec de bonnes propriétés de la FAC, on rappelle le principe de génération et les propriétés des différentes familles de codes utilisés dans les systèmes CDMA. Finalement, dans la dernière partie de ce chapitre, on montre les critères et les conditions sur la base desquels le code le plus approprié est sélectionné.

Dans le chapitre 2, on présente tout d'abord un état de l'art des méthodes d'optimisations les plus utilisées dans les différents domaines technologiques. On met l'accent sur les AGs et la méthode OEP qui sont utilisés pour optimiser les codes que nous avons proposé. Ensuite, on expose les approches de génération et d'optimisation des séquences proposées récemment dans la littérature scientifique. Une attention particulière sera accordée aux méthodes de génération des séquences présentant de très bonnes performances en termes de la FAC. Finalement, on aborde les méthodes proposées pour générer de longues séquences binaires basées sur la combinaison d'un certain nombre de sous-séquences binaires plus courtes.

Le troisième chapitre est consacré à la mise en œuvre des méthodes efficaces que nous avons proposées pour générer deux types d'OLBSS avec de bonnes propriétés de la FAC. En effet, dans un premier temps, on donne une description détaillée de la méthode de génération de ces deux types d'OLBSS. Dans un deuxième temps, on décrit les principes des méthodes, utilisant les AGs et la méthode OEP, pour l'optimisation des séquences proposées. Le choix de la fonction fitness et les paramètres de chaque algorithme seront

aussi discutés dans la même section. Les résultats de simulation, pour différents scénarios de test, seront présentés dans la dernière section de ce chapitre.

Enfin, une conclusion générale récapitule les contributions de cette thèse et propose des perspectives aux travaux effectués.

# Chapitre 1 : Etude des caractéristiques des séquences PN pour les applications à accès multiple.

Chapitre 1

# Etude des caractéristiques des séquences PN pour les applications à accès multiple.

## 1.1 Introduction

Dans ce chapitre, nous présentons en détail, le principe de base de l'étalement de spectre ainsi que les avantages et les inconvénients qu'il présente. En effet, après un rappel de la technique DS-SS, qui est une technique utilisée dans l'étalement de spectre, nous décrivons après le principe d'une technique bien connue et largement répandue dans les systèmes de communications, à savoir la CDMA. Les méthodes de génération des différents types de séquences d'étalement utilisées dans cette technique seront données par la suite. Finalement, on va dévoiler les critères sur la base desquels on va présenter une étude comparative des différents types de codes d'étalement pour arriver finalement à choisir les codes le plus appropriés.

## 1.2 Etalement de spectre

Les systèmes à spectre étalé ont été développés depuis les années 1940. Les applications initiales étaient tout d'abord destinées aux communications militaires [34]. Elles montrent une bonne résistance aux interférences et admettent à la transmission d'être noyée dans le bruit. Avec l'essor des systèmes des systèmes de positionnement GNSS [35-36], les techniques d'étalement de spectre ont présenté un grand intérêt pour les applications grand public.

### 1.2.1 Principe des techniques d'étalement des spectres

Le principe de cette technique de communications consiste à partager l'énergie du signal à transmettre sur une bande de fréquence plus large. Il peut être expliqué par la relation de C. E. Shannon qui exprime la capacité maximale $C$ du canal, perturbé par un bruit, comme suit [37]:

$$C = B \cdot log_2(1 + S/N) \quad 1\text{-}1$$

Avec :

- $C$: Capacité maximale du canal en bit/s ;
- $B$ : Bande occupée par le signal à transmettre en Hertz (Hz) ;
- $S$ : Puissance du signal à transmettre en Watt (W) ;
- $N$ : Puissance du bruit en W.

D'après cette relation, il est clair que pour émettre, sans erreur, une quantité $C$ donnée, il est possible d'utiliser soit une bande $B$ étroite et un fort rapport signal sur bruit « $S/N$ », soit une large bande $B$ et un faible rapport signal sur bruit « $S/N$ ». L'étalement de spectre repose sur cette dernière idée. Il consiste à élargir le support fréquentiel du message à transmettre et donc d'émettre avec un rapport $S/N$ très faible. La bande passante du signal émis est alors largement supérieure à la bande du signal utile.

Il existe plusieurs méthodes d'étalement de spectre d'ont on peut citer [38] :

- L'étalement de spectre par saut de fréquence FHSS où les données sont émises sur des fréquences différentes qui changent régulièrement. Ce changement est indiqué par le code d'étalement [38].
- L'étalement de spectre par saut d'intervalle de temps THSS qui consiste à transmettre les chips qui composent la séquence de code sur différentes tranches de temps [38].
- L'étalement par séquence directe DSSS [39] où le codage des données s'effectue de manière directe et l'information est étalée en joignant un code d'étalement.

### 1.2.2 Principe de l'étalement de spectre par séquence directe

La technique DS-SS est basée sur la multiplication de chaque bit de l'information par un code PN. Le débit de ce dernier est très grand par rapport à celui du signal informatif. De ce fait, la largeur de bande de fréquences à émettre devient très grande que celle du message informatif. En notant $T_d$ la durée d'un bit d'information et $T_c$ celle d'un chip de la séquence d'étalement, le signal émis a une largeur de bande égale à $1/T_c$ supérieure à celle du message à transmettre $1/T_d$. Le rapport *N* entre ces deux largeurs de bande est appelé gain de traitement. Il est donné par :

$$N = T_d / T_c \qquad 1\text{-}2$$

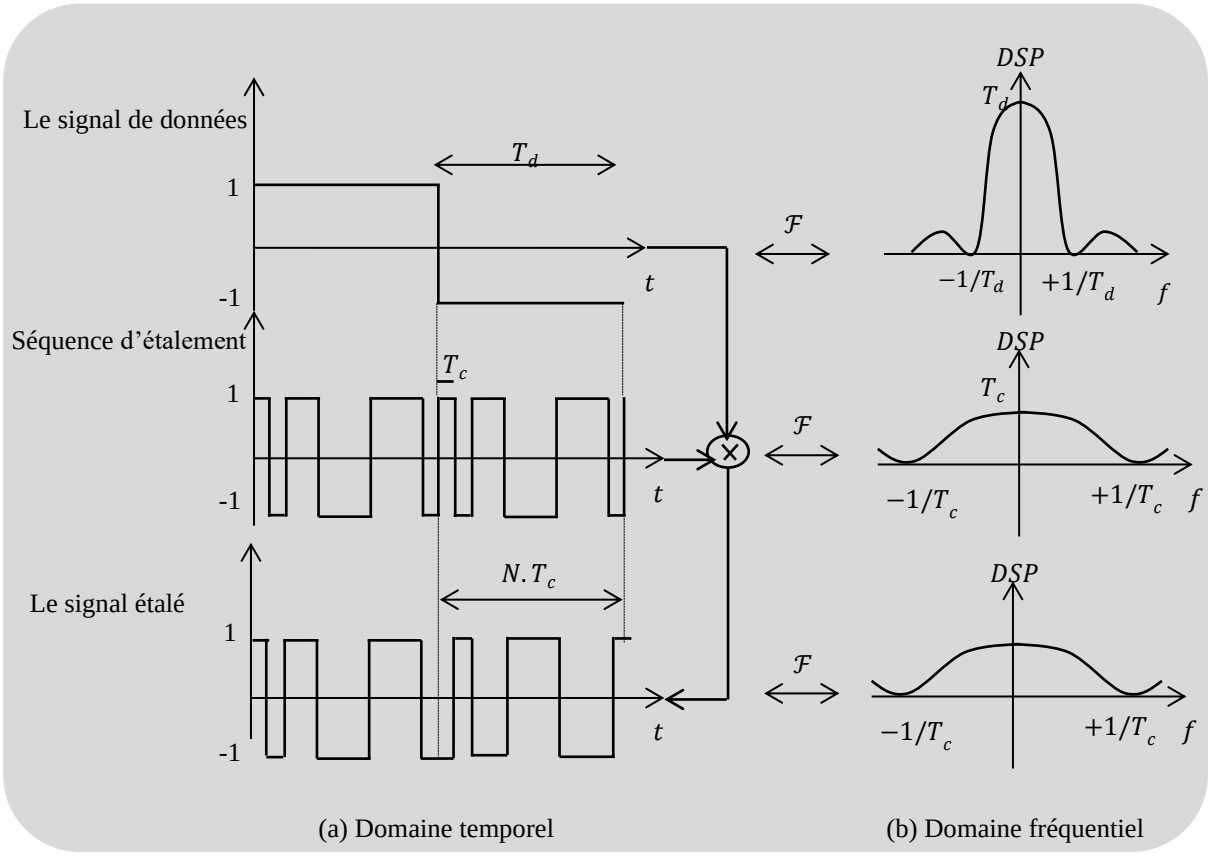



**Figure 1-1 Principe de l'étalement de spectre par séquence directe.**

La Figure 1-1 illustre le principe de la technique DS-SS [40]. Dans les Figure 1-1.a et 1-1.b, sont représentés respectivement et successivement :

- Le signal de données binaires à transmettre en bande de base (constitué ici de deux bits) et son occupation spectrale étroite avec une Densité Spectrale de Puissance (DSP) élevée.
- Un exemple de séquence d'étalement PN et son occupation spectrale étalée sur une bande de fréquence plus large introduisant une diminution de la DSP. Ce signal prend alors la forme d'un bruit.
- La version finale du signal DS-SS transmit sur le canal et résultant de la multiplication des données par la séquence d'étalement PN. L'occupation spectrale est étalée par un facteur $N$ et les bits du signal ont une durée d'émission $T_c$, identique à celle des bits de la séquence PN.

Au niveau du récepteur, comme montré dans la Figure 1-2, afin de récupérer le signal de données d'origine en bande de base, le signal reçu est multiplié par une copie générée localement du même code PN synchronisée avec la séquence appliquée à l'entrée.

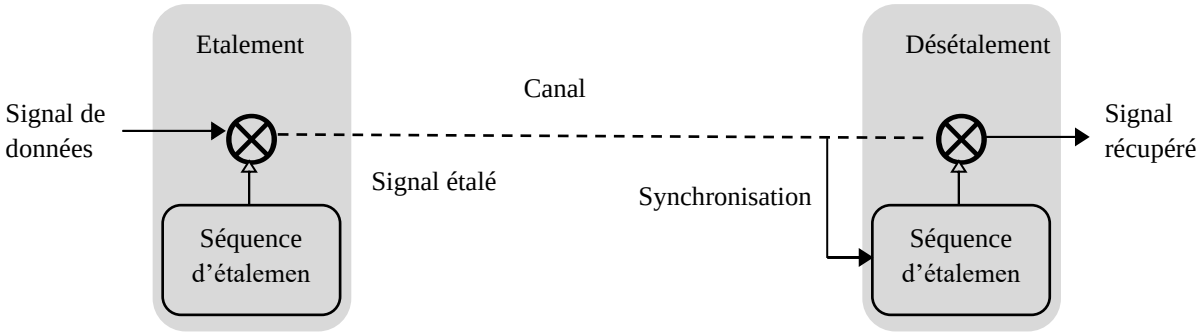



**Figure 1-2 Schéma général d'un système à étalement de spectre DS-SS.**

Dans un système à plusieurs utilisateurs, chaque utilisateur dispose d'un code PN propre à lui, ce qui oblige le récepteur à connaître le code associé à l'utilisateur qui l'intéresse pour pouvoir faire la séparation entre les différents utilisateurs du système.

### 1.2.3 Avantages de l'étalement de spectre

L'étalement de spectre est une technique possédant beaucoup d'avantages [3] [41]. Parmi ceux-ci, on peut notamment citer :

- **Confidentialité (faible probabilité d'interception)**

Ici, le signal émis correspond beaucoup à du bruit et par conséquent les utilisateurs ne possédant pas une copie du code d'étalement, utilisé en émission, ne peuvent pas découvrir le contenu de la communication.

- **Insensibilité aux effets des trajets multiples**

L'étalement de spectre a la capacité d'éliminer ou d'atténuer l'effet des trajets multiples.

- **Bonne résistance aux perturbations bande étroite**

Lors de l'émission, des perturbations à bande étroite peuvent s'ajouter au signal étalé. Le récepteur, en réalisant l'opération de dés-étalement, va récupérer l'information désirée à nouveau sur une bande étroite alors que les perturbations vont être étalées. C'est de cette façon qu'on se débarrasse de la puissance des perturbations qui devient alors négligeable devant celle du signal utile reconstitué.

- **Multiplexage**

L'opportunité de la mise en œuvre des méthodes CDMA permettant à plusieurs utilisateurs, disposant chacun d'un code spécifique, d'émettre en même temps dans les mêmes bandes de fréquences.

### 1.2.4 Inconvénients

Malgré ses nombreux avantages, l'étalement de spectre présente quelques inconvénients. On peut citer comme exemple :

- **Synchronisation temporelle précise**

Pour réaliser la corrélation entre le code local du récepteur et celui qui se trouve dans le signal émis, une synchronisation temporelle est nécessaire. Une mauvaise synchronisation temporelle peut provoquer l'apparition d'un bruit supplémentaire ce qui engendrerait une interférence additionnelle.

- **Complexité accrue**

La complexité accrue des systèmes à étalement de spectre rend leur coût plus élevé par rapport à celui des systèmes à bande étroite.

- **Large bande passante**

La bande passante utilisée à l'émission est largement très grande par rapport à celle du message à transmettre.

## 1.3 Accès multiples et CDMA

Dans ce qui suit, nous allons voir les différentes techniques d'accès multiple utilisées dans les systèmes de communications.

### 1.3.1 Techniques d'accès multiples

Les trois principales techniques d'accès multiple utilisées dans les systèmes de communication, telles que présentées par la Figure 1-3, sont :

- L'accès multiple par répartition de fréquences FDMA (Frequency Division Multiple Access) qui utilise le multiplexage fréquentiel [42]. Ici, les utilisateurs émettent simultanément mais dans des bandes de fréquences différentes.
- L'Accès multiple par répartition de temps TDMA (Time Division Multiple Access) où chaque utilisateur émet dans la même bande de fréquence mais à des instants différents [43-44].
- La CDMA où tous les utilisateurs émettent dans la même bande et en même temps.

Dans notre travail, nous nous sommes intéressés à la catégorie des systèmes CDMA car c'est la plus intéressante parmi les trois classes, du fait qu'elle n'impose aucune contrainte de temps ni de fréquence.

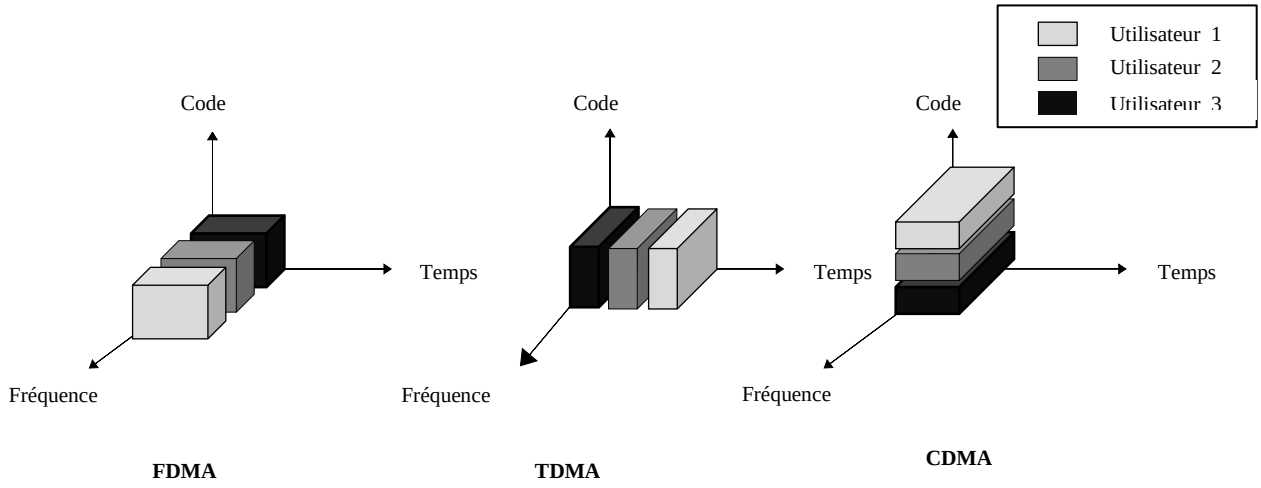



**Figure 1-3 Différents types de multiplexage.**

### 1.3.2 Technique d'accès CDMA

Dans la CDMA, de nombreux utilisateurs ont accès à un canal collectif et peuvent l'utiliser conjointement jusqu'à une certaine limite définie par la capacité du système. En effet, chaque utilisateur dans le système CDMA utilise un code PN différent pour moduler son signal, ce qui oblige le récepteur à connaître le code associé à l'utilisateur désiré.

Il existe deux différentes catégories de la technologie CDMA :

- **CDMA synchrone (S-CDMA) :** cette technologie exploite les propriétés d'orthogonalité entre les vecteurs qui représentent les chaînes de données. Chaque utilisateur pour la S-CDMA utilise un code orthogonal pour moduler son signal. Donc, la FIC est égale à zéro pour les codes orthogonaux.
- **CDMA asynchrone (A-CDMA) :** cette technologie utilise des séquences PN et ne fixe aucune contrainte de synchronisation entre les utilisateurs.

Contrairement aux techniques TDMA et FDMA, la capacité de multiplexage de la technique CDMA n'est pas limitée par des paramètres physiques (intervalles de temps disponibles ou fréquences utilisables), mais par la capacité de générer un maximum de codes avec de bonnes propriétés de corrélations et par l'ajout d'autres critères tels que le facteur de crête ou encore l'IAM. Donc, le choix des codes d'étalement est une étape très importante durant la préparation d'une chaîne CDMA. Les facteurs influant sur le choix de ces séquences seront décrits dans les sections suivantes.

## 1.4 Séquences d'étalement (Spreading codes)

Les séquences d'étalement sont définies comme étant des suites périodiques. Chaque suite ne peut prendre que deux valeurs. Nous désignerons les séquences binaires, avec des éléments de $\{0,1\}$, comme des séquences unipolaires et celles avec des éléments de $\{-1,1\}$ comme des séquences bipolaires. Les séquences ternaires ont des éléments de $\{-1,0,1\}$.

Dans cette partie, nous allons présenter les différentes familles de codes couramment utilisés dans les systèmes à accès multiple.

### 1.4.1 Séquences PN à longueur maximale ou m-séquences

Les séquences de longueur maximale (m-séquences) [4], sont par définition, les codes les plus longs qui peuvent être générés à partir des registres LFSR [5] de $n$ étages.

La Figure 1-4 montre la structure générale du registre LFSR à $n$ étages.

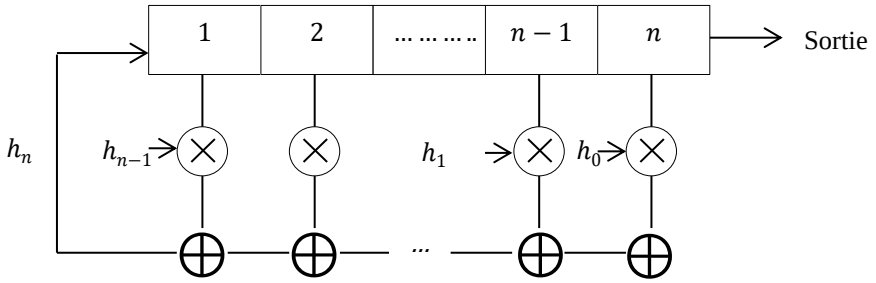



**Figure 1-4 Structure du générateur de M-séquence.**

L'opérateur $\oplus$ est l'opérateur modulo 2. Le polynôme $h(x)$ de longueur $n$, décrivant la séquence à longueur maximale, est donné par :

$$h(x) = \sum_{i=0}^{n} h_i \; x^i \qquad 1\text{-}3$$

Les $h_i$ prennent la valeur 1 s'il y a la connexion et 0 sinon et $h_0 = h_n = 1$.

Pour qu'une séquence de code soit de longueur maximale, il est nécessaire que son polynôme caractéristique soit primitif. La séquence à longueur maximale ainsi obtenue est de longueur $N$ égale à $2^n - 1$. Elle est composée de $(N-1)/2$ bits à «-1 » et $(N+1)/2$ bits à «1 ».

### 1.4.2 Séquences de Gold

L'une des séquences CDMA la plus populaire est la séquence de Gold [6, 45], qui a été étudiée pour la première fois par Robert Gold en 1968. Les séquences de Gold sont générées par l'addition modulo 2 (l'opération OU exclusif, XOR) d'une paire préférée [46] de m-séquences différentes $u$ et $v$ de même longueur $N = 2^n - 1$. La Figure 1-5 montre un schéma de génération des séquences de Gold.

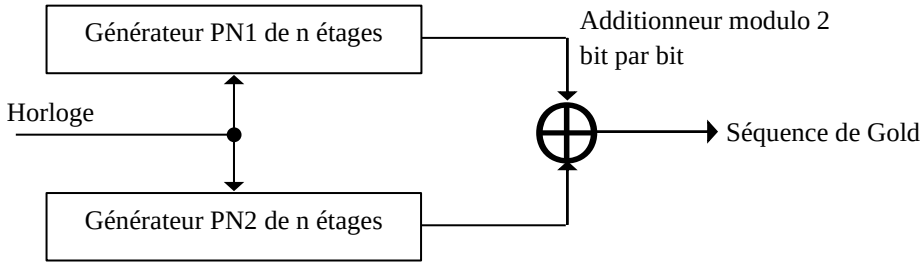



**Figure 1-5 Construction des Codes de Gold.**

Les registres à décalage sont initialisés avec différentes valeurs non nulles pour obtenir toutes les séquences de l'ensemble $G(u, v)$:

L'ensemble des séquences de Gold générées avec les deux m-séquences $u$ et $v$ est donné par l'Equation 1-4.

$$G(u, v) = \{u, v, u \oplus v, u \oplus T \cdot v, u \oplus T^2 \cdot v, \dots, u \oplus T^{N-1} \cdot v\} \qquad 1\text{-}4$$

Donc, les séquences de Gold sont générées par la somme modulo 2 de la première séquence $u$ avec des versions décalées de la deuxième séquence $v$ ou vice-versa.

La séquence résultante $G(u, v)$ n'est pas à longueur maximale mais toujours de longueur $N$. Il est possible de générer au total « $N + 2$ » séquences de longueur $N$.

Les paires de m-séquences préférées, nécessaires pour générer des codes de Gold, sont données dans l'annexe A.

### 1.4.3 Séquences de Gold orthogonales

Les séquences de Gold orthogonales [47-48] sont réalisées à partir des séquences de Gold précédemment présentées. Elles sont obtenues en ajoutant un chip supplémentaire «0» à la séquence de Gold originale et avec une FIC qui peut avoir des valeurs nulles. Le nombre total des codes de Gold orthogonaux, de longueur $N$, obtenus est égal à $2^n$. Ces

codes, initialement proposés pour les systèmes CDMA synchrones, ont l'avantage d'être orthogonaux.

### 1.4.4 Séquences de Gold-Like

Ces séquences [4] ont des paramètres très similaires à ceux des séquences de Gold. Elles sont générées à partir des m-séquences. Dans ce qui suit, nous allons voir la procédure utilisée pour générer ce type de séquences.

On considère une séquence $u$ de longueur $N$ générée par un polynôme générateur d'ordre $n$. Soit $q$ un nombre entier tel que $gcd\ (q, N) = 3$ (Greatest Common Divisor), la séquence $v^{(k)}$ $(k = 0,1,2)$ est obtenue par la décimation de $uT^{(k)}$ par $q$ avec une répétition de 3 fois. $T$ est l'opérateur de décalage cyclique. La séquence $v^{(k)}$est périodique de période égale à $N/3$. La nouvelle séquence, appelée séquence de Gold-like, est définie par [25] :

$$\begin{aligned} H_q(u,v) = \{u, u \oplus v^{(0)}, u \oplus T\, v^{(0)}, u \oplus T\, v^{(0)}, \dots, u \oplus T^{N'-1}\, v^{(0)}, u \\ \oplus v^{(1)}, u \oplus T\, v^{(1)}, \dots, u \oplus T^{N'-1}\, v^{(1)}, u \\ \oplus v^{(2)}, u \oplus T\, v^{(2)}, \dots, u \oplus T^{N'-1}\, v^{(2)}, \} \end{aligned} \quad 1\text{-}5$$

Il est possible de générer au total « $N + 1 = 2^n$ » séquences de longueur « $N = 2^n - 1$ » chacune.

### 1.4.5 Séquences de Kasami

Les séquences de Kasami [7] sont principalement générées à partir d'une séquence à longueur maximale $u$. Elles sont pareillement des séquences PN de longueur $N = 2^n - 1$, qui sont définies pour des valeurs paires de $n$. Il existe deux classes de séquences de Kasami: les séquences dites « small-set » et celles dites « large-set ».

#### a Séquences « Small-set »

Dans la procédure de génération de ces séquences, nous commençons par la génération d'une m-séquence $u$ en utilisant un polynôme générateur d'ordre $n$, et nous formons une séquence binaire $w$ en prenant chaque $q$ bits de $u$, tel que $q = 2^{n/2} + 1$. En d'autres termes, la séquence $w$ est obtenue par la décimation de $u$ par $q$ avec une répétition de $q$ fois de telle sorte que $w$ soit de même longueur que $u$ . Avec $u$ et $w$, on forme un nouvel ensemble de séquences par l'addition de $u$ et $w$ décalée cycliquement par $2^{n/2} - 2$.

On obtient ainsi un ensemble de $N = 2^{n/2}$ séquences binaires de longueur $2^n - 1$, défini comme suit [4] :

$$K_s(u) = \left\{u, u \oplus w, u \oplus Tw, \dots, u \oplus T^{2^{n/2}-2}w\right\} \quad 1\text{-}6$$

Où : $T$ est l'opérateur de décalage cyclique.

**b Séquences « Large-set »**

Les séquences « large-set » de Kasami notée $K_L$ , inclut à la fois des séquences de Gold (ou Gold-like) et « large-set » de kasami en tant que sous-ensembles.

Suivant la valeur de $n$, il existe deux manières pour définir ces séquences :

- Si $n \; mod \; 4 = 2$, les séquences « large-set » de Kasami sont construites comme suit :

Supposons que $u$ est une séquence à longueur maximale de période $N = 2^n - 1$ générée par un polynôme générateur d'ordre $n$, $w$ et $v$ deux séquences générées respectivement par la décimation de u par $2^{n/2} + 1$ et par $2^{(n+2)/2} + 1$. Alors, l'ensemble des séquences est généré en additionnant $u, v$ et $w$ avec différents décalages de $v$ et $w$.

Les séquences $K_L$ sont définies comme suit [4]:

$$K_L(u) = G(u,v) \bigcup \left[\bigcup_{i=0}^{2^{\frac{n}{2}}-2} \left\{T^i w \oplus G(u,v)\right\}\right] \quad 1\text{-}7$$

Où : $T$ est l'opérateur de décalage cyclique et $G(u,v)$ est défini dans l'Equation 1-4

Par conséquent, cette famille de codes contient au total $M = 2^{3n/2} + 2^{n/2}$ séquences.

- Si $n \; mod 4 = 0$, les séquences $\mathrm{K_L}$ sont définies comme suit [4]:

$$K_L(u) = H_q(u) \bigcup \left[\bigcup_{i=0}^{2^{\frac{n}{2}}-2} \left\{T^i w \oplus H_q(u)\right\}\right] \bigcup \left\{u^j \oplus T^k w : 0 \le j \le 2, 0 \le k < (2^{\frac{n}{2}} - 1)/3\right\} \quad 1\text{-}8$$

Où : $H_q(u)$ est défini dans l'Equation 1-5.

Le nombre de séquences dans l'ensemble est $M = 2^{3n/2}$ séquences.

### 1.4.6 Séquences de Walsh-Hadamard

Les séquences de Walsh-Hadamard [16] sont parmi les structures orthogonales les plus simples à construire. Une séquence de code correspond à chaque ligne et chaque colonne de la matrice de Hadamard.

Une matrice de Walsh $[N \times N]$ peut être construite récursivement de la manière suivante :

$$H_0 = [0] \tag{1-9}$$

$$H_{N/2} = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & \overline{H_{N/2}} \end{bmatrix} \tag{1-10}$$

Où : $N$ est une puissance de 2 et $\overline{H_{N/2}}$ symbolise le complément binaire de $H_{N/2}$
Les matrices obtenues sont constituées de $2^n$ séquences binaires, chacune de longueur $2^n$.

### 1.4.7 Séquences OVSF

Les codes OVSF [15] sont utilisés pour séparer les différents canaux physiques d'un utilisateur. Ils sont aussi appelés les codes de canalisation. Ces codes sont construits par une relation matricielle donnée par :

$$C_N = \begin{bmatrix} C_N(0) \\ C_N(1) \\ C_N(2) \\ C_N(3) \\ . \\ . \\ C_N(N-2) \\ C_N(N-1) \end{bmatrix} = \begin{bmatrix} C_{N/2}(0) & C_{N/2}(0) \\ C_{N/2}(0) & \overline{C_{N/2}(0)} \\ C_{N/2}(1) & C_{N/2}(1) \\ C_{N/2}(1) & \overline{C_{N/2}(1)} \\ & . \\ & . \\ C_{N/2}(N/2-1) & C_{N/2}(N/2-1) \\ C_{N/2}(N/2-1) & \overline{C_{N/2}(N/2-1)} \end{bmatrix} \tag{1-11}$$

Où $C_N(n)$ est la vectrice ligne de $N$ éléments. $N = 2^K$ ($K$ est un entier positif) et $\overline{C_{N/2}(n)}$ est le complément binaire de $C_{N/2}(n)$. On note que $C_N(n)$ n'est définie que si $N$ est une puissance de 2. Ces codes sont également définis par un arbre OVSF [17] (Figure 1-6) où chaque nœud possède 2 fils. Les codes de ces derniers sont issus de celui de leur père commun.

Dans la figure 1-6, les codes OVSF sont notés par $C_{SF,k}$, où $SF$ représente le facteur d'étalement (longueur du code) et $k$ ($0 \leq k \leq SF - 1$) le numéro du code. Toutes les séquences de code situées à un même niveau hiérarchique de l'arbre sont de même

longueur. Dans un tel arbre, il n'est pas possible d'utiliser simultanément tous les codes OVSF. En effet, le code de chaque nœud est déterminé en fonction du code du nœud père. Ceci implique que pour une branche donnée, les codes ont une relation entre eux.

**Figure 1-6 Arbre de génération des codes OVSF.**

Par exemple, si le code $C_{4,1}$ dans l'arbre est affecté à un utilisateur, les codes $C_{1,0}$ , $C_{2,0}$, $C_{8,2}$, $C_{8,3}$, etc, ne peuvent pas être attribués à n'importe quel autre utilisateur dans la même cellule. Cette règle impose que le nombre d'utilisateurs dans ce cas est limité.

### 1.4.8 Séquences de Golay

Les séquences de Golay [18] ont été introduites pour la première fois par Marcel J.E Golay en 1949. Celles-ci sont des séquences qui sont à la fois complémentaires et orthogonales. Elles sont bien adaptées aux systèmes de transmission synchrones.

Les codes de Golay (CG) sont obtenus à partir d'une matrice construite d'une récursive. En effet, les CG de longueur $N$ correspondent aux lignes de la matrice de taille $N \times N$ définie par :

$$\begin{cases} CG_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = [A_2 \quad B_2] \\ CG_N = [A_N \quad B_N \ ] \end{cases} \qquad 1\text{-}12$$

Une paire de Golay est composée de deux séquences $A_N$ et $B_N$, $N$ est le nombre de bits ou la longueur des séquences qui est égale à $2^n \neq 0$. Ces deux séquences sont définies par :

$$\begin{cases} A_N = \begin{bmatrix} A_{N/2} & B_{N/2} \\ A_{N/2} & B_{N/2} \end{bmatrix} = [A_2 \quad B_2] \\ B_N = \begin{bmatrix} A_{N/2} & -B_{N/2} \\ -A_{N/2} & B_{N/2} \end{bmatrix} \end{cases} \quad 1\text{-}13$$

Ou les matrice $A_N$ et $B_N$ sont de taille $N \times \frac{N}{2}$ chacune; par exemple en posant $N= 8$ on obtient:

$$CG_8 = \begin{bmatrix} +1 & +1 & +1 & -1 & +1 & +1 & -1 & +1 \\ +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 \\ +1 & +1 & -1 & +1 & +1 & +1 & +1 & -1 \\ +1 & -1 & -1 & -1 & +1 & -1 & +1 & +1 \\ +1 & +1 & +1 & -1 & -1 & -1 & +1 & -1 \\ +1 & -1 & +1 & +1 & -1 & +1 & +1 & +1 \\ +1 & +1 & -1 & +1 & -1 & -1 & -1 & +1 \\ +1 & -1 & -1 & -1 & -1 & +1 & -1 & -1 \end{bmatrix} \quad 1\text{-}14$$

Ces séquences sont orthogonales entre elles. Elles ont également la particularité d'être complémentaires deux à deux. Deux codes *A* et *B* sont dits complémentaires si et seulement si [49]:

$$R_{A,A}(\tau) + R_{B,B}(\tau) = 2N\delta(\tau) \quad 1\text{-}15$$

Où : $R_{A,A}(\tau)$ et $R_{B,B}(\tau)$ sont respectivement les FAC apériodiques des séquences $A$et $B$.

Pour $A$, un ensemble de codes composé de $L$ séquences ($L > 2$) de longueur $N$ noté par $A_i$ et $R_{A_i,A_i}(\tau)$ la FAC apériodique de la séquence $A_i$ , cet ensemble est dit complémentaire si et seulement si :

$$\sum_{i=1}^{L} R_{A_i,A_i}(\tau) = NL\delta(\tau) \quad 1\text{-}16$$

### 1.4.9 Séquences de Weil

Les séquences de Weil sont nommées d'après André Weil et seront utilisées dans les applications futures du GPS L1C [14]. Elles sont basées sur des séquences de Legendre et peuvent être obtenues par l'addition (ou exclusive) de ces dernières. Les séquences de Legendre sont basées sur les résidus quadratiques [50] qui peuvent être définis comme suit :

Soit $P$ un nombre premier et "$a$" un nombre positif ($a < P$), on dit que "a" est un résidu quadratique modulo $P$, s'il existe un entier $\ell$ tel que $\ell^2 \equiv a \, mod \, P$.

Nous pouvons prouver que "$a$" est un résidu quadratique par le calcul de la valeur de l'expression suivante [51] :

$$f(a) = a^{(L-1)/2} \, mod \, P \qquad 1\text{-}17$$

Cette expression ne peut prendre que les valeurs 1 et -1 ; par conséquent, la séquence aléatoire de Legendre est formée de la manière suivante [51] :

$$[\, 0 \, f(1) \, f(2) \ldots f(L-1)] \qquad 1\text{-}18$$

Une autre façon de générer les séquences de Legendre est basée sur le symbole de Legendre ($a \, / \, P$) qui est défini par [51] :

$$\left(\frac{a}{P}\right) = \begin{cases} +1 \; si \; a \; est \; un \; résidu \; quadratique \; modulo \; P \\ -1 \; si \; a \; n'est \; pas \; un \; résidu \; quadratique \; modulo \; P \\ 0 \quad si \; P \; divise \; a \end{cases} \qquad 1\text{-}19$$

Avec $a \in [0, P-1]$.

La séquence de Legendre $u$, de longueur $P$ est donnée par [51] :

$$\left(0, \left(\frac{1}{P}\right), \left(\frac{2}{P}\right), \ldots, \left(\frac{L-1}{P}\right)\right) \qquad 1\text{-}20$$

Les séquences de Weil « $W$ » sont obtenues par le « ou exclusif » des séquences de Legendre $u$ générées précédemment avec leurs répliques décalées. Elles sont données par [51] :

$$W(u) = \left\{u \oplus Tu, u \oplus T^2u, \dots T^{floor(P/2)}u\right\} \quad \text{1-21}$$

Il est donc possible de générer au total : $(P-1)\ /\ 2$ séquences de longueur $P$.

### 1.4.10 Séquences de Barker

Ces séquences existent uniquement pour les longueurs $n = 2, 3, 4, 5, 7, 11\ et\ 13$. Le tableau 1-1 liste les séquences de Barker pour les différentes longueurs.

Les séquences de Barker sont en nombre très limité et ne sont pas utilisées pour l'accès multiple. Elles semblent être les séquences les plus adaptées aux applications radar.

**Tableau 1-1 Séquences de Barker**

| **Longueur $N$ de la séquence** | **Séquence** |
|---|---|
| 2 | [1 1], [-1 1] |
| 3 | [1 1 -1] |
| 4 | [1 1 -1 1] |
| 4 | [1 1 1 -1] |
| 5 | [1 1 1 -1 1] |
| 7 | [1 1 1 -1 -1 1 -1] |
| 11 | [1 1 1 -1 -1 -1 1 -1 -1 1 -1] |
| 13 | [1 1 1 1 1 -1 -1 1 1 -1 1 -1 1] |

### 1.4.11 Séquences ZCZ

Les séquences ZCZ ont été proposées pour réduire l'IAM dans les systèmes CDMA. Le principe de génération de ces séquences est donné comme suit :

Soit un ensemble $Z = \{z_1, \dots, z_k, \dots, z_K\}$ de $K$ séquences de longueur $N$ chacune, où $z_k = \{z_{k,1}, \dots, z_{k,n}, \dots, z_{k,N}\}$. L'ensemble $Z$ peut également être écrit, sous forme matricielle, comme suit :

$$Z = \begin{matrix} z_{11} & \cdots & z_{1,n} & \cdots & z_{1,N} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ z_{k,1} & \cdots & z_{k,n} & \cdots & z_{k,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{K,1} & \cdots & z_{K,n} & \cdots & z_{K,N} \end{matrix} \quad \text{1-22}$$

L'ensemble des séquences ZCZ est désigné par $ZCZ(N, K, Z_0)$, où $N$ est la longueur de la séquence, $K$ le nombre des séquences et $Z_0$ est la longueur de la zone zéro. Toute

paire ( $z_i, z_j$) de séquences ZCZ dans l'ensemble possède la propriété de corrélation suivante [52].

$$E_{z_i z_j}(l) = \begin{cases} \mu_i N & l = 0, \quad i = j \\ 0 & l = 0, \quad i \neq j \\ 0 & 0 < |l| \leq Z_0 \end{cases} \quad \text{1-23}$$

La limite suivante a été définie pour les paramètres des séquences ZCZ.

$$Z_0 \leq \frac{N}{K} - 1 \quad \text{1-24}$$

Pour un ensemble de séquences bipolaires, la limite ZCZ devient :

$$Z_0 \leq \frac{N}{2K} \quad \text{1-25}$$

#### 1.4.11.1 Séquences ZCZ binaires

Il existe plusieurs méthodes de génération de ces séquences dont on peut citer les deux suivantes.

- $1^{ère}$ méthode :

Fan dans [52] a proposé des séquences ZCZ binaires. La matrice MOCS (Mutually Orthogonal Complementary Set), notée par $F^{(n)}$, de $K$ lignes à $M$ éléments chacune est définie par [49]:

$$F = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1M} \\ H_L & -H_L & \ddots & F_{2M} \\ & \vdots & \ddots & \vdots \\ F_{K1} & F_{K2} & \cdots & F_{KM} \end{bmatrix} \quad \text{1-26}$$

La longueur de la séquence est $N = ML$. Par l'utilisation de l'opération de concaténation, nous pouvons obtenir une matrice $F^{(n+1)}$ de $2K$ lignes, chacune de longueur $2N$.

$$\acute{F} = \begin{bmatrix} FF & (-F)F \\ (-F)F & FF \end{bmatrix} \quad \text{1-27}$$

Notez que : $FF$ désigne la matrice dont la i ème ligne est obtenue par la concaténation de la i ème ligne de $F$ avec sa copie [49]. Si $F^{(0)}$ est une matrice de Hadamard d'ordre 2 ($K = N = 2$), les lignes de $F^{(n)}$ étendues $n$ fois forment des séquences $ZCZ(2^{2n+1}, 2^{n+1}, 2^{n-1})$.

- $2^{\text{ème}}$ méthode

Cette méthode est décrite dans la référence [53]. L'ensemble des séquences ZCZ peut être construit à partir de la matrice WH de Hadamard et la technique d'entrelacement. La première matrice est donnée par :

$$F = \begin{bmatrix} F^1 \\ F^2 \end{bmatrix} = \begin{bmatrix} -H_L & +H_L & +H_L & +H_L \\ +H_L & -H_L & -H_L & +H_L \end{bmatrix} \quad 1\text{-}28$$

Il y a $K = 2L$ séquences dans $F$ chacune de longueur $N = 4L$. Nous utilisons l'opération d'entrelacement de $F^1$ et $F^2$ pour obtenir l'ensemble $Z^{(0)}$ donné par :

$$Z^{(0)} = \begin{bmatrix} F^1 \otimes F^2 \\ F^1 \otimes (-F^2) \end{bmatrix} \quad 1\text{-}29$$

L'ensemble $Z^{(0)}$ est une séquence ZCZ avec $ZCZ\,(8L, 2L.1)$. En général, l'ensemble $Z^{(m)}$ avec $ZCZ\,(2^{m+3}L, 2L.1^{m+1} - 1)$ peut être construit de manière récursive par l'entrelacement de la séquence $Z^{(m-1)}$ comme suit :

$$Z^{(m)} = \begin{bmatrix} Z^{(m-1)} \otimes Z^{(m-1)} \\ Z^{(m-1)} \otimes (-Z^{(m-1)}) \end{bmatrix} \quad 1\text{-}30$$

#### 1.4.11.2 Séquences ZCZ ternaires

Similairement aux séquences binaires, il existe plusieurs méthodes de génération des séquences ternaires dont on peut citer les deux suivantes.

- $1^{\text{ère}}$ méthode

Cette méthode est proposée par Hayashi dans la référence [54]. L'ensemble des séquences ternaires $F$ de $K = 2L$ séquences de longueur $N = 2(L + 1)$ chacune peut être construit à partir de la matrice $H_L$ de Hadamard en ajoutant des colonnes de zéro élément comme suit :

$$F = \begin{bmatrix} F^1 \\ F^2 \end{bmatrix} = \begin{bmatrix} +H_L & 0 & +H_L & 0 \\ +H_L & 0 & -H_L & 0 \end{bmatrix} \quad 1\text{-}31$$

Un ensemble de séquences ZCZ ternaire $Z^0$ est obtenu en entrelaçant les lignes de $F^1$ et $F^2$ comme suit :

$$Z^0 = \begin{bmatrix} F^1 \otimes F^2 \\ F^1 \otimes (-F^2) \end{bmatrix} \quad 1\text{-}32$$

L'ensemble $Z^0$ constitue l'ensemble de départ. En général, l'ensemble $Z^{(m)}$ peut être construit de manière récursive par l'entrelacement de ZCZ séquences $Z^{(m-1)}$ comme suit :

$$Z^{(m)} = \begin{bmatrix} Z^{(m-1)} \otimes Z^{(m-1)} \\ Z^{(m-1)} \otimes (-Z^{(m-1)}) \end{bmatrix} \quad 1\text{-}33$$

Donc, on obtient un ensemble de séquences ZCZ ternaires de $ZCZ\ (2^{m+2}(L+1), 2L.2^{m+1}-1)$.

- $2^{\text{ème}}$ méthode

Cette construction est basée sur une suite ternaire parfaite $t = (t_1, \dots, t_n, \dots, t_{N_1})$ et l'ensemble orthogonal binaire $H_{N_2} = [h_1, \dots, h_i, \dots, h_{N_2}]^T$ . Si $gcd\ (N_1, N_2) = 1$ , l'ensemble des séquences ZCZ ternaires $Z$ avec $ZCZ\ (N_1N_2, N_2.N_1 - 1)$ , peut être construit comme suit :

$$Z_{i,k} = t_{k\ mod N_1} h_{i,k\ mod N_2} \quad 1\text{-}34$$

Où : $k = 1 \dots, N_1N_2$, pour l'élément $Z_{i,k}$ de la séquence ZCZ.

La séquence ZCZ obtenue à partir de l'Equation 1-25 est donnée par :

$$Z_i = \{Z_{i,1}, \dots, Z_{i,k} \dots, Z_{i,N_1N_2}\} \quad 1\text{-}35$$

Les exemples qui détaillent les différentes méthodes de génération de l'ensemble des séquences ZCZ mentionné ci-dessus, sont donnés dans l'annexe A.

### 1.4.12 Séquences de De Bruijn

La DBS, nommée d'après le mathématicien néerlandais Nicolaas Govert DBen 1946 [11], est une séquence cyclique $b = \{b_1, \dots, b_{2^n}\}$ d'ordre $n$ et de longueur $2^n$ bits notée $b_i \in \{0,1\}$, dans laquelle chaque sous- séquence $\{a_1, \dots, a_n\} \in \{0,1\}^n$ de longueur $n$ apparaît exactement une fois dans $b$.

Les DBS peuvent être caractérisées par deux paramètres : le nombre $k$ d'entités dans l'alphabet, par ex. {0,1,2,3,4,5,6,7,8,9} pour $k = 10$, et l'ordre $n$ (longueur de la sous-séquence requise). Celles-ci sont généralement notées par $B\ (k\ , n\ )$ et ont une longueur égale à $k^{\ n}$. Pour les séquences binaires, $k = 2$.

Cette définition peut peut-être mieux éclairée en considérant comme exemple la séquence cyclique de DB d'ordre *n=3* et de longueur 10, soit : 0001110100. Les sous-chaînes uniques de longueur 3 sont :

000 →001 →011 →111 →110 →101 →010 →100 →000

Le nombre de séquences de DBuniques pour une valeur de $n$ donné est égal à $2^{2^{n-1}-n}$ [11].

#### 1.4.12.1 Méthodes de génération

L'un des problèmes de la séquence DBest de savoir comment construire efficacement cette dernière. De nombreux chercheurs ont proposé divers algorithmes pour réaliser cette tâche. Le plus efficaces sont données comme suit :

- La méthode du registre à décalage LFSR basé sur des polynômes primitifs de Golomb [55] ;
- La méthode des règles des successeurs proposées par Swada, Williams et Wong [56] ;
- La méthode de concaténation de mots de Lyndon en ordre lexicographique proposé par Fredricksen et Maiorana [57] ;
- Les méthodes des constructions « Greedy » y compris : « prefer-smallest » (ou équivalent « prefer-largest ») [58,59], le « prefer-same » [60], prefer-one [60] et « prefer-opposite » [61].

Certains de ces algorithmes seront décrits dans ce qui suit.

**a Méthode du registre à décalage LFSR**

Les DBS d'ordre n sont construites par un registre LFSR de $n$ étages et généré par un polynôme primitif de $\mathbb{F}_2[x]$ de degré $n$ , en ajoutant « 0» au motif $(n-1)$ bits (00.. 0) dans la sortie du LFSR.

Par exemple, on peut considérer un registre LFSR généré avec le polynôme primitif : $P(x) = x^3 + x + 1$. L'état initial du registre est $\{0\ 0\ 1\}$. Donc, la séquence générée par ce polynomial est [0 0 1 0 1 1 1]. En ajoutant 0 dans le terme le plus nul de la séquence, nous obtenons la DBS [0 0 0 1 0 1 1 1].

Si le polynôme utilisé pour générer la séquence est $P(x) = x^3 + x^2 + 1$ avec l'état initial {0 0 1}, alors la DBS binaire obtenue est [0 0 0 1 1 1 0 1]. En effet, les différents états initiaux du registre avec le même polynôme génèrent la même séquence.

Supposons que, nous utilisons le polynôme $x^4 + x + 1$ et {0001} comme état initiale. Ce polynôme génère la séquence [0 0 0 1 0 0 1 1 0 1 0 1 1 1 1]. Cependant, si l'on sélectionne tous les états initiaux possibles, il en résulte la même séquence comme le montre le Tableau 1-2.

Pour générer l'ensemble des séquences de De Bruijn, on a besoin de polynômes primitifs différents tout en ajoutant un 0 dans le terme le plus nul à la fin de la séquence obtenue. Par conséquent, Le nombre de DBS distinctes générées par cette méthode est limité par le nombre de polynômes primitifs de degré $n$ sur GF (2) qui est égal à $\emptyset\,(2^n - 1)\,/\,n$, où $\emptyset$ est la fonction d'Euler-phi.

**Tableau 1-2 DBS binaire avec le polynôme primitif $x^4 + x + 1$**

| Etat initial | Séquence |
|---|---|
| 0 0 0 1 | 0 0 0 1 0 0 1 1 0 1 0 1 1 1 1 |
| 0 0 1 0 | 0 0 1 0 0 1 1 0 1 0 1 1 1 1 0 |
| 0 1 0 0 | 0 1 0 0 1 1 0 1 0 1 1 1 1 0 0 |
| 1 0 0 0 | 1 0 0 0 1 0 0 1 1 0 1 0 1 1 1 |
| 0 0 1 1 | 0 0 1 1 0 1 0 1 1 1 1 0 0 0 1 |
| 0 1 0 1 | 0 1 0 1 1 1 1 0 0 0 1 0 0 1 1 |
| 0 1 1 0 | 0 1 1 0 1 0 1 1 1 1 0 0 0 1 0 |
| 1 0 0 1 | 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0 |
| 1 0 1 0 | 1 0 1 0 1 1 1 1 0 0 0 1 0 0 1 |
| 1 1 0 0 | 1 1 0 0 0 1 0 0 1 1 0 1 0 1 1 |
| 0 1 1 1 | 0 1 1 1 1 0 0 0 1 0 0 1 1 0 1 |
| 1 0 1 1 | 1 0 1 1 1 1 0 0 0 1 0 0 1 1 0 |
| 1 1 0 1 | 1 1 0 1 0 1 1 1 1 0 0 0 1 0 0 |
| 1 1 1 0 | 1 1 1 0 0 0 1 0 0 1 1 0 1 0 1 |
| 1 1 1 1 | 1 1 1 1 0 0 0 1 0 0 1 1 0 1 0 |

### b Règle de successeur basée sur un collier

La règle de successeur basée sur un collier est un nouvel algorithme basé sur l'ensemble des chaînes cycliques d'équivalence développé par Swada, William et Wong [56]. Un collier est le plus petit élément lexicographique de la classe d'équivalence d'une

chaîne cyclique. Par exemple, 0001, 0010, 0100, 1000 sont dans la même classe et 0001 est le collier.

La construction de DBS binaire par cet algorithme est très simple en appliquant la règle suivante :

$$f(x_1\, x_2\, \ldots x_n) = \begin{cases} x_2\, \ldots x_n \overline{x_1} & ,si\ x_2\, \ldots x_n 1\ est\ un\ collier \\ x_2\, \ldots x_n x_1 & Ailleurs \end{cases} \quad \text{1-36}$$

Le symbole $\overline{x}$ signifie le complément de *x* ou en binaire cela signifie *1– x*.

La construction de la DBS binaire d'ordre 3 par cette règle est effectuée en considérant par exemple 000 comme chaîne initiale, puis les chaînes suivantes sont 001, 011, 111, 110, 101, 010, 100. Enfin, en concaténant le premier symbole de chaque chaîne, on obtient la séquence : [0 0 0 1 1 1 0 1].

**c Méthodes de « Prefer- one » et « Prefer- opposite »**

Dans ce qui suit, nous allons donner le principe de ces deux méthodes.

- **Méthode de « Prefer- one »**

C'est la méthode de construction la plus simple qui commence par $n$ zéros et ajoute le bit 1 à la séquence chaque fois que cela est possible. Les étapes de cet algorithme, pour la séquence d'ordre $n$, sont donnés comme suit [60] :

1. Démarrer avec une séquence de $n$ zéros ;
2. Ajouter, dans la mesure du possible, un 1 à la suite. Si les $n$ derniers bits n'ont pas été rencontrés avant, accepter et répéter l'étape 2, sinon passer à l'étape suivante ;
3. Ajouter, dans la mesure du possible, un 0 à la suite. Si les $n$ derniers bits n'ont pas été rencontrés avant, accepter et répéter l'étape 2; sinon arrêter.

- **Méthode de « Prefer- opposite »**

Cet algorithme, proposé par Alhakim [61], est semblable au cas antérieur. Ici, au lieu d'ajouter le bit 1 à chaque étape, on continue par le bit opposé au dernier bit de la séquence. En cas d'échec, on ajoute le même bit, et si on perd encore, l'algorithme s'arrête.

Les séquences construites à partir de ces deux algorithmes sont indiquées dans le Tableau ci-dessous pour $1 \leq n \leq 5$ [61].

**Tableau 1-3 Comparaison de la séquence construite par Prefer-one et Prefer-opposite.**

| ***N*** | **Méthode « Prefer-one »** | **Méthode « Prefer-opposite »** |
|---|---|---|
| **1** | 01 | 01 |
| **2** | 0011 | 0011 |
| **3** | 00011101 | 00010111 |
| **4** | 0000111101100101 | 0000101001101111 |
| **5** | 00000111110111001101011000101001 | 00000101011010010001100111011111 |

### d Graphe de DB

Pour vérifier que toutes les chaînes de bits possibles de longueur $n$ se produisent une et une seule fois dans la séquence de De Bruijn, on utilise la théorie des graphes de De Bruijn. Le graphe de DB est un graphe orienté défini par :

$$DB_n = (\{0,1\}^{n-1}, \{0,1\}^n) \qquad 1\text{-}37$$

Où $\{0,1\}^{n-1}$ est l'ensemble des « sommets », et $\{0,1\}^n$est l'ensemble des arcs entre deux sommets.

Si $a = \{a_1, \dots, a_{n-1}\}$ et $\acute{a} = \{\acute{a}_1, \dots, \acute{a}_{n-1}\}$ sont deux sommets, il y a un arc de $a$ vers $\acute{a}$ si : $a_2 = \acute{a}_1, a_3 = \acute{a}_2, \dots, a_{n-1} = \acute{a}_{n-2}$.

Cela revient à dire qu'il y a un arc de $a$ vers $\acute{a}$ si $a$ est un préfixe d'une chaîne $b_1 \dots b_n$ et $\acute{a}$ est un suffixe de la même chaîne de successeurs. La présence d'un arc signifie donc un chevauchement maximal entre deux mots de même longueur. Pour illustrer cette définition, on considère le graphe $B(2,3)$ présenté dans la Figure 1-7 qui est construit sur une chaine binaire $A = \{0,1\}$ pour des mots de longueur $n = 3$. Les $2^3 = 8$ mots de longueur 3 sont : {000, 001, 010, 011, 111, 110, 101, 100}.

Il existe un arc allant du sommet {000} vers le sommet {001} car le suffixe de longueur 2 de {000} est égal au préfixe de longueur 2 de {001}. De même, il existe un arc allant du sommet {100} vers le sommet {000}.

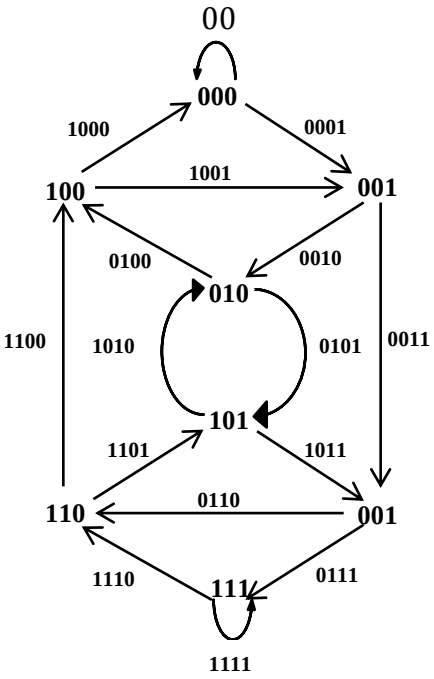



**Figure 1-7 Graphe DB d'ordre 3.**

## 1.5 Facteurs influant sur le choix des codes d'étalement

Il y a plusieurs facteurs qui peuvent être utilisés pour l'étude des performances des systèmes de communication à base de CDMA. Les plus importants sont donnés dans ce qui suit.

### A. L'Autocorrélation et l'Intercorrélation

Le premier critère agissant sur le choix des séquences est leurs propriétés d'autocorrélation et d'inter-corrélation qui permettent de mesurer la dépendance d'une séquence avec sa copie ou avec une autre séquence.

#### a. L'autocorrélation

La FAC est habituellement définie comme la mesure de ressemblance entre un code cyclique $c_i$ et une copie de ce même code décalée. La FAC est définie par :

$$R_{i,i}[\tau] = \sum_{i=1}^{N} c_i . c_{i-\tau} \qquad 1\text{-}38$$

La FAC du code doit être maximale en zéro, et très faible ailleurs pour permettre une meilleure synchronisation des séquences en réception.

**b. L'inter-corrélation**

La FIC décrit les termes d'interférences entre deux codes $c_i$ et $c_j$. Elle est définie par :

$$R_{i,j}[\tau] = \sum_{i=1}^{N} c_i . c_{j-\tau} \qquad 1\text{-}39$$

La FIC mesure le degré de ressemblance entre deux séquences différentes. Lorsque celle-ci est nulle quel que soit le décalage $\tau$, les codes sont dits orthogonaux (en réalité, les codes ne sont pas parfaitement orthogonaux). Par conséquent, ces codes doivent avoir des valeurs très faibles de la FIC (proche de 0) pour minimiser l'IAM dû à la transmission simultanée de plusieurs utilisateurs.

**B. Facteur de mérite**

Le facteur de mérite FM est une autre mesure quantitative importante de la qualité d'une séquence. Ce facteur a été établi par Golay [62]. Pour une séquence binaire $S$ de longueur $N$, le FM est défini par :

$$FM(S) = \frac{R_S(0)^2}{\sum_{u\neq 0}|R_S(u)|^2} = \frac{N^2}{2\sum_{u=1}^{N-1}|R_S(u)|^2} \qquad 1\text{-}40$$

Où $|R_S(u)|^2$ est l'énergie du $k^{ième}$ pic secondaire. Les valeurs élevées des pics

secondaires correspondent à l'énergie inefficace et non désirable dans le signal. Ainsi, le FM relie l'énergie du pic principal et l'énergie totale des pics secondaires. En effet, les séquences qui possèdent de grandes valeurs du FM sont très utiles pour des applications telles que l'estimation de canal, le traitement des signaux Radar, et les communications DSSS.

### C. Longueur et taille de l'ensemble de codes

La longueur du code doit être assez grande pour que le signal étalé accomplisse les propriétés d'un bruit aléatoire. Cela permettra également de garantir la sécurité adéquate du message.

Pour garantir la couverture d'un grand nombre d'utilisateurs dans le système, la famille de codes doit être très grande. Il est très difficile d'assurer un grand dictionnaire constitué de codes ayant les propriétés souhaitables.

## 1.6 Comparaison entre les différentes séquences étudiées

Le Tableau 1-4 résume les différentes propriétés de chacune des séquences décrites précédemment. Ces propriétés seront considérées dans l'étude réalisée dans le cadre de cette thèse de Doctorat. Les valeurs des FIC périodiques paires données sont les valeurs absolues maximales que peuvent prendre ces dernières.

A la vue des différentes propriétés présentées dans le Tableau 1-4, il apparaît que la longueur des différentes séquences orthogonales est $2^{\mathrm{N}}$. Cette quantité qui est une puissance de deux est simple à mettre en œuvre par les algorithmes base de transformées de Fourier Rapide.

Les séquences de Barker, quant à elles, ont des longueurs très limitées. Par conséquent, ces codes ne peuvent pas être utilisés seuls dans les systèmes DS-CDMA.

Les longueurs des séquences (à longueur maximale, Gold, Gold-like et Kasami) sont égales à $2^{n}-1$. La longueur des séquences de Weil est un nombre premier. Ces séquences sont préférables à générer en comparaison aux autres séquences puisque la densité des nombres premiers dans l'ensemble des nombres naturels est supérieure à $2^{n}-1$.

**Tableau 1-4 Propriétés des séquences d'étalements étudiées**

| Famille de séquences | $n$ | Longueur des séquences | Nombre des séquences M | Valeurs max de corrélation |
|---|---|---|---|---|
| M-séquence | Pair / Impair | $2^n - 1$ | *1* | |
| Séquence de Gold | Impair | $2^n - 1$ | $N + 2$ | $2^{\frac{n+1}{2}} + 1$ |
| Séquence de Gold- Like | $gcd(n;\ k) = 3$ | $2^n - 1$ | $2^n$ | $2^{\frac{n+2}{2}} + 1$ |
| « Small set » de Kasami | Pair | $2^n - 1$ | $\sqrt{N+1}$ | $2^{\frac{n+2}{2}} + 1$ |
| « Large set » de Kasami | Pair et<br>$mod\ (n,4) = 2$<br>$mod\ (n,4) = 0$ | $2^n - 1$ | <br>$2^{n/2}(2^n + 1)$<br>$2^{n/2}(2^n + 1)$-*1* | $2^{\frac{n+2}{2}} + 1$ |
| Séquence de Gold orthogonale | | $N$ | $2^n$ | $0$ |
| Séquence de Walsh-Hadamard | ------- | $N$ | $N$ | $0$ |
| Séquence de Golay | ------- | $N$ | $N$ | $0$ |
| Séquence de Weil | ------- | $P$ est premier | $(P-1)\ /\ 2$ | $5 + 2\sqrt{N}$ |
| Barker | ------- | *2, 3, 4, 5, 7, 11, 13* | *7* | ---------- |
| Séquence ZCZ | Pair | $2^{2n+1}$ | $2^{n+1}$ | |
| Séquence de De Bruijin | | $2^n$ | $2^{2^{n-1}-n}$ | $2^n - 4 * \frac{2^n}{2n}$ |
| Séquence de Bent | $mod\ (n,4) = 0$ | $2^n - 1$ | $\sqrt{N+1}$ | $2^{\frac{n+2}{2}} + 1$ |
| Séquence de No | *mod (n, 2)=0* | $2^n - 1$ | $\sqrt{N+1}$ | $2^{\frac{n+2}{2}} + 1$ |
| Séquence de Zadoff-Chu | | $N$ est premier | *N-1* | $\sqrt{N}$ |

La Figure 1-8 représente une comparaison entre les longueurs possibles des séquences à générer dans le cas des codes PN et Weil.

Afin de faciliter la comparaison de ces différentes familles de séquences étudiées, le Tableau 1-5 résume leurs propriétés pour une longueur de séquence $N$ égal à :

- 63 bits pour les m-séquences, les séquences de Gold et les séquences de Kasami ;
- 13 bits pour les séquences de Barker ;
- 64 bits pour les séquences de Gold orthogonale, les séquences de Walsh-Hadamard, les séquences de Golay et les séquences de ZCZ ;
- 61 bits pour les séquences de Weil.

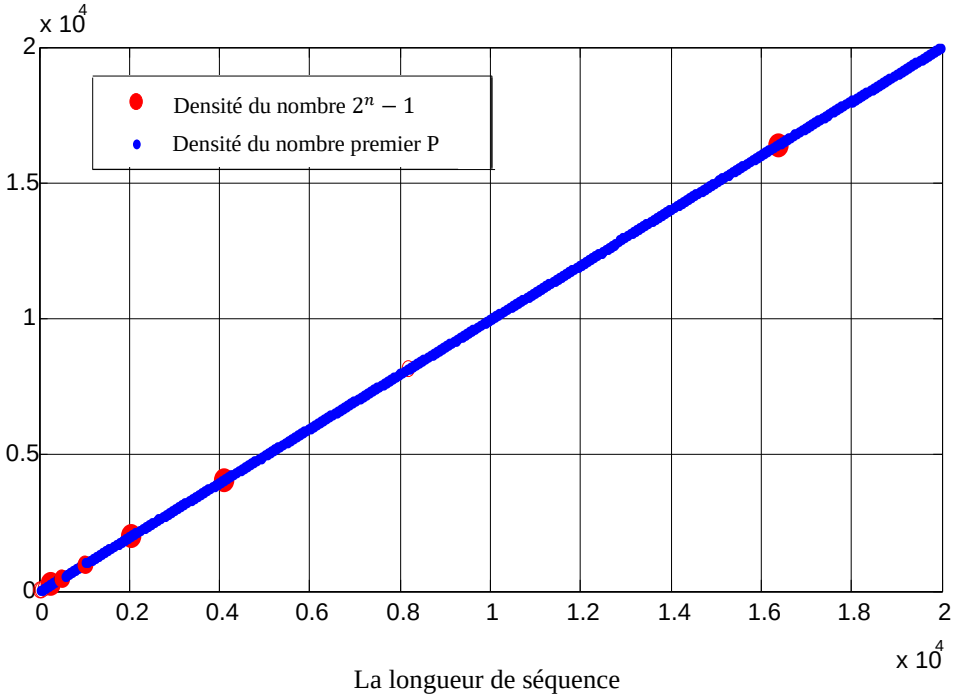



**Figure 1-8 Comparaison entre les longueurs possibles à générer dans le cas des séquences de longueur $2^n - 1$ et les séquences de Weil**

**Tableau 1-5 Comparaison des propriétés de séquences étudiées**

| **Famille des séquences** | **Longueur de la séquence** | **Nombre des séquences M** | **FIC périodique pair** | |
|---|---|---|---|---|
| | | | $\tau = 0$ | $\tau \neq 0$ |
| Séquence de Gold orthogonale | 64 | 64 | 0 | $\chi$ |
| Séquence de Walsh-Hadamard | 64 | 64 | 0 | $\chi$ |
| Séquence de Golay | 64 | 64 | 0 | $\chi$ |
| Séquence de Gold | 63 | 65 | 17 | |
| Séquence « Small set » de Kasami | 63 | 8 | 9 | |
| Séquence « Large set » de Kasami | 63 | 520 | 17 | |
| Séquence de Weil | 61 | 30 | 15 | |
| Séquence ZCZ | 64 | 8 | | |

Le Tableau 1-5 montre que la famille des séquences du « large-set » de Kasami offre un nombre de séquences beaucoup plus important que celui des autres séquences. Dans le même contexte, l'inconvénient majeur des m-séquences et des séquences de Barker est la limitation du nombre de codes qu'on peut générer, ce qui limite le nombre minimal des utilisateurs rendant ainsi ces codes insuffisants pour les applications à accès multiple.

Les Figures 1-9 et 1-10 montrent les valeurs de la FAC pour les différentes séquences de longueurs $N$ bits.

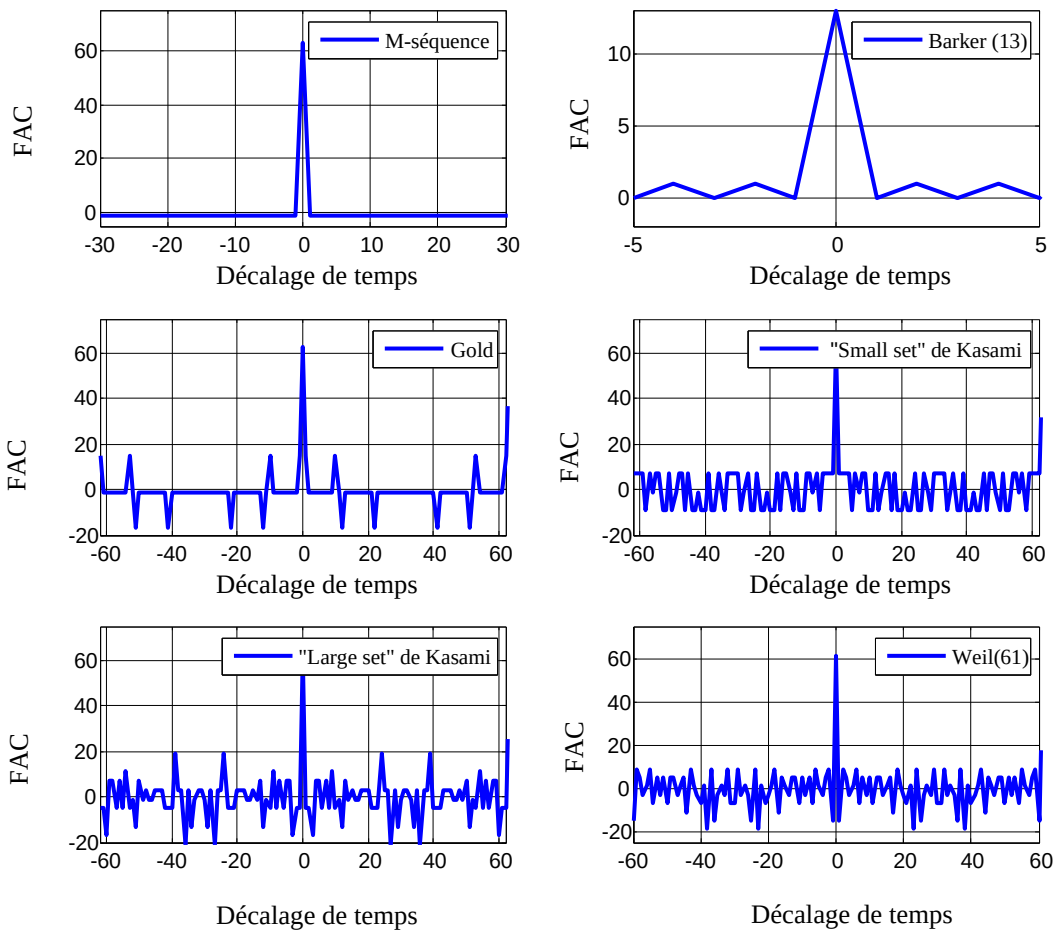



**Figure 1-9 Comparaison des propriétés des FACs des différentes séquences, de longueur $N$, étudiées.**

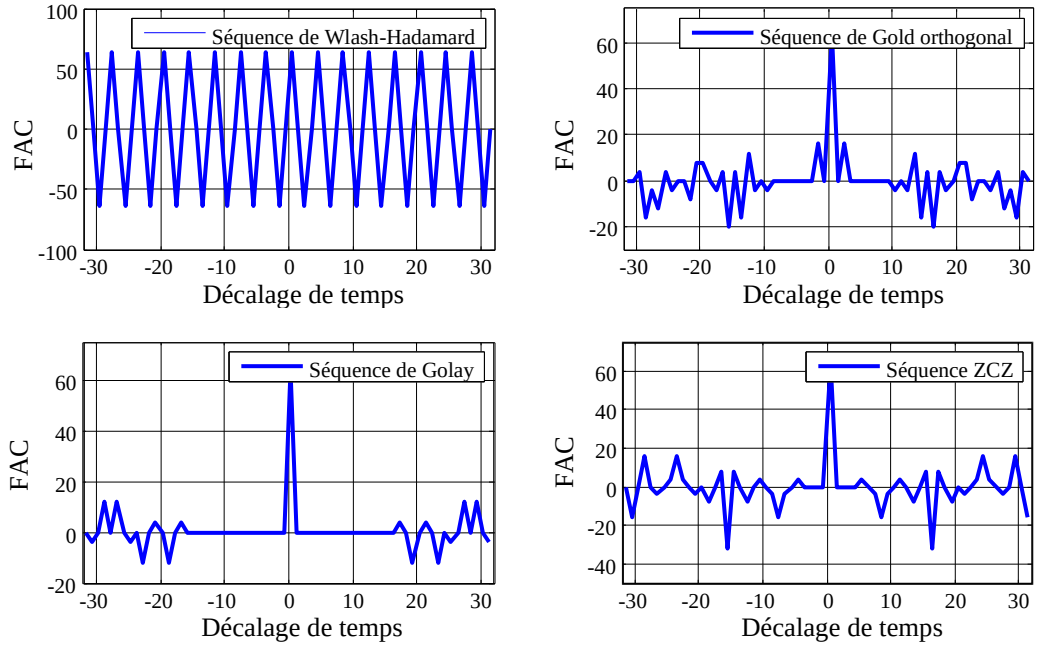



**Figure 1-10 Comparaison des propriétés des FACs des différentes séquences orthogonales, de longueur $N$, étudiées.**

D'après la Figure 1-9, on remarque que les m-séquences ont de bonnes propriétés de la FAC avec l'absence totale de pics secondaires. D'après ces résultats, parmi toutes les autres séquences, les séquences de Barker présentent une FAC très proches par rapport aux m-séquences. En effet, ces séquences montrent des pics secondaires de très faibles amplitudes.

Les codes appartenant à l'ensemble de « small set » de Kasami possèdent de meilleures propriétés de FAC en comparaison avec ceux de « large set » de Kasami et de Gold. La famille des séquences de Weil présente des FACs ayant des pics secondaires qui sont plus faibles par rapport à ceux de la plupart des autres familles.

Comme le montre la Figure 1-10, l'inconvénient majeur des séquences de Walsh-Hadamard est qu'elles possèdent de mauvaises propriétés de la FAC en comparaison avec celles des autres séquences car ces dernières contiennent des pics secondaires non négligeables pour des valeurs non nulles de $\tau$. De plus, les séquences de Golay possèdent de meilleures propriétés de la FAC pour une valeur non nulle de $\tau$, en comparaison avec les séquences de Walsh-Hadamard et celles de Gold orthogonaux.

Le dernier plan à droite de la Figure 1-10 montre que la FAC des séquences ZCZ possède des pics secondaires qui ont de grandes valeurs pour des décalages temporels différents de 0.

Le Tableau 1-5 résume les propriétés de la FIC des séquences étudiées pour la même longueur $N$ de la séquence.

Dans un contexte synchrone (aucun décalage de temps), la FIC des séquences de Gold orthogonaux, de Walsh-Hadamard et de Golay est toujours nulle. C'est pour cette raison que ces codes sont dits orthogonaux.

De la même façon, les caractéristiques de la FIC sont également représentées sur la Figure 1-11.

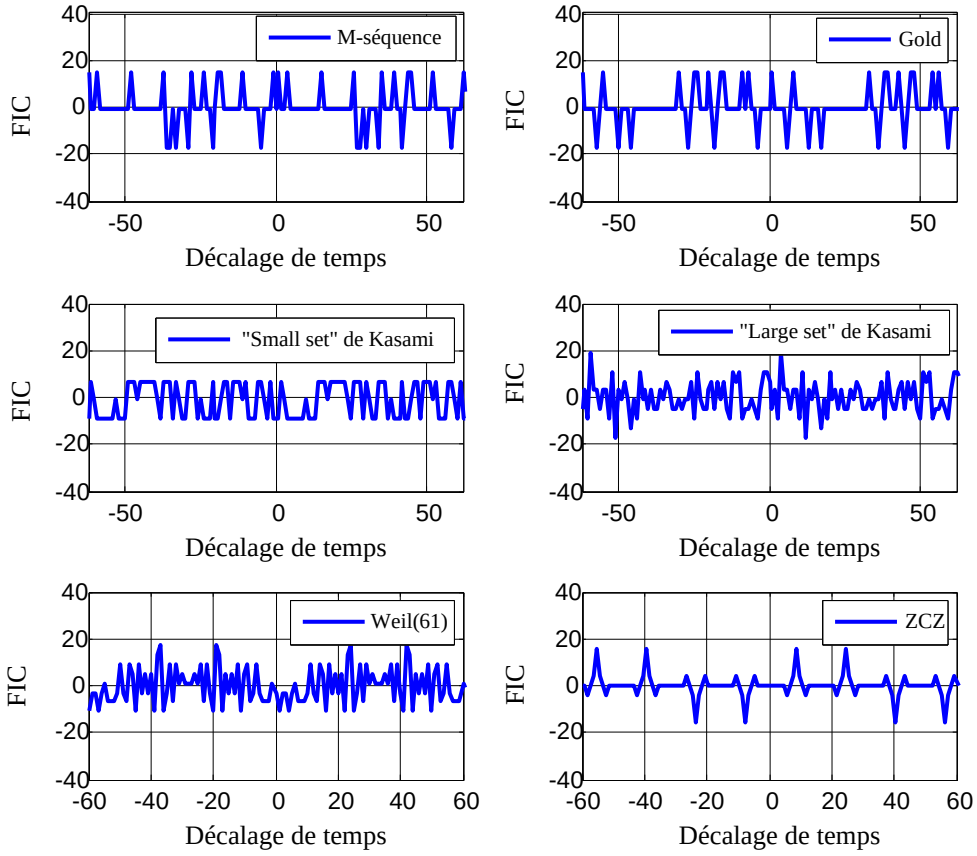



**Figure 1-11 Comparaison des propriétés des FIC des différentes séquences, de longueur *N*, étudiées.**

A la vue des résultats du Tableau 1-5 et de la Figure 1-11, il apparaît que les séquences du « small set » de Kasami possèdent les valeurs de la FIC les plus faibles (-9,-l et 7). Les séquences de Weil ont des pics secondaires de la FIC qui sont à des limites inférieures (-15, 13). Les familles des codes du « large-set » de Kasami et les codes de Gold possèdent des propriétés de corrélations conformes. La FIC des séquences ZCZ possède des pics secondaires qui ont de grandes amplitudes.

## 1.7 Conclusion

Dans ce chapitre, nous avons présenté les caractéristiques techniques qui forment la base des systèmes de communication. Nous avons montré que l'étalement du spectre, qui offre la possibilité de mise en œuvre des techniques CDMA, permet à plusieurs utilisateurs d'émettre simultanément dans les mêmes bandes de fréquences et en même temps. Après avoir présenté les différentes familles de séquences d'étalement, les propriétés de chacune d'elles, ont été montrées et analysées. L'étude menée montre, en premier lieu, que les séquences à longueur maximale et les séquences de Barker ont de bonnes propriétés de

corrélations mais leur nombre est limité comparer aux codes de Gold et de Kasami. La famille des codes du large-set de Kasami offre un nombre de séquences beaucoup plus important que celui des codes de Gold tout en possédant des propriétés de corrélation presque similaires. De plus, Les séquences de Weil ont de bonnes propriétés de corrélation en comparaison avec d'autres familles de séquences. En outre, la taille de la famille est très importante. Ces séquences ont l'avantage d'avoir des longueurs significativement plus grandes par rapport aux séquences des autres familles. Tous ces faits rendent les séquences de Weil potentiellement très applicables dans les signaux modernisés du système de navigation GNSS.

Dans le prochain chapitre, nous donnons un état de l'art sur les principaux travaux de recherche sur les méthodes d'optimisation et de génération des séquences d'étalement.

# Chapitre 2 : Etat de l’art sur les techniques de génération et d’optimisation des séquences d’étalement

Chapitre 2

# Etat de l’art sur les techniques de génération et d’optimisation des séquences d’étalement.

## 2.1 Introduction

Dans ce chapitre, Nous donnons un état de l’art sur quelques travaux publiés dans la littérature scientifique traitant le problème d’optimisation et de génération des séquences d’étalement ayant de bonnes propriétés de corrélation.

En effet, dans une première partie, nous présentons un état de l’art des principaux algorithmes d’optimisation en mettant l’accent sur les algorithmes les plus répandus.

La deuxième partie est consacrée aux principaux travaux publiés sur les méthodes de génération des séquences qui présentent de très bonnes performances en termes de la FAC.

Dans la dernière partie, nous montrons les méthodes de génération de longues séquences binaires basées sur l'utilisation d'un certain nombre de sous-séquences binaires plus courtes.

## 2.2 Méthodes d’optimisation : Etat de l’art

De nombreux auteurs ont publié des études détaillées sur les diverses méthodes employées pour résoudre le problème d’optimisation et de génération des séquences d’étalement. Ces méthodes se subdivisent en deux groupes : les méthodes exactes et les méthodes approchées. Dans cette section, nous abordons les méthodes les plus usuelles.

### 2.2.1 Méthodes de résolution exactes

Une méthode de résolution exacte permet de trouver une solution optimale à un problème donné. Parmi les méthodes exactes les plus connues, nous pouvons citer l'algorithme de séparation et d'évaluation B&B (Branch and Bound) [63], ses dérivées (Branch and cut, Branch and-Price etc.), la programmation dynamique [64], la programmation par contraintes [65] et la programmation linéaire [66]. Dans ce qui suit un bref aperçu sera donné pour chacun des types de cette catégorie.

#### 2.2.1.1 Algorithme de séparation et d'évaluation

L'algorithme B&B, proposé par Land et Doig [63], permet de trouver une solution optimale en divisant récursivement le problème original en plus petits problèmes et en résolvant ces derniers.

La procédure B&B se base sur les deux principes suivants :

- Un principe de séparation qui consiste à diviser le problème original $S_0$ en sous-ensembles ;
- Un principe d'évaluation qui consiste à définir une évaluation de la valeur des solutions, pour chaque sous-ensemble obtenu, par séparation. Il permet de réduire l'espace de recherche en éliminant quelques sous-ensembles qui ne contiennent pas la solution optimale.

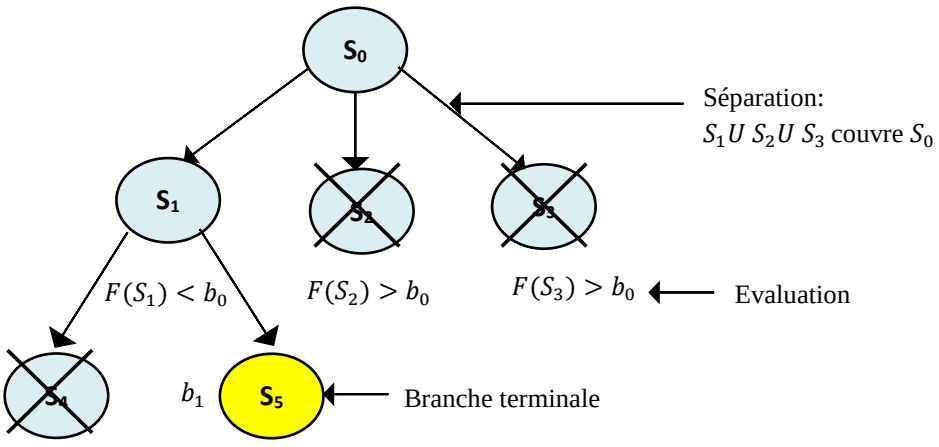



**Figure 2-1 Etude d'un arbre de recherche.**

Comme montré dans la Figure 2-1, la méthode B&B peut être représentée à travers une arborescence. En effet, la racine $S_0$ de cette arborescence représente l'ensemble de toutes les solutions du problème considéré. Les autres sommets représentent des sous-ensembles de $S_0$ obtenus successivement par séparation.

Pour prévenir l'évaluation de la totalité des solutions réalisables, les bornes inférieure et supérieure sont calculées à chaque division du problème. Lorsque la borne supérieure d'une branche de l'arbre est inférieure à une borne inférieure déjà connue, toutes les solutions générées par ce assemblage sont ignorées puisqu'elles ne peuvent pas renfermer la solution optimale.

#### 2.2.1.2 Programmation dynamique

La programmation dynamique a été introduite au début des années 1950 par Richard Bellman [64]. Elle repose sur le principe d'optimalité qui stipule que « Toute politique optimale ne peut être formée que de sous-politiques optimales ». Elle ressemble à la méthode B&B. La seule différence entre ces deux méthodes est que la programmation dynamique permet aux sous-problèmes de se superposer. En d'autres termes, un sous-problème peut être utilisé dans la solution de deux sous-problèmes différents. En contrepartie, l'approche B&B crée des sous-problèmes qui sont complètement séparés et qui peuvent être résolus distinctement l'un de l'autre.

### 2.2.2 Heuristiques

Le mot heuristique vient du verbe du grec ancien heuriskein (ευρισκειν) et qui signifie 'trouver'. Les heuristiques sont des méthodes d'optimisation simple permettant de trouver rapidement une solution à un problème donné sans toutefois fournir de garantie sur la qualité des solutions. Les heuristiques font parties de la famille des procédés stochastiques. En effet, compte tenue de l'aspect aléatoire présent dans ce type de méthodes, l'algorithme ne retrouve pas la même solution entre deux exécutions indépendantes. Elles comprennent, par exemple :

- Les méthodes constructives qui engendrent des solutions à partir d'une solution initiale ;
- Les méthodes de recherche locales, qui démarrent avec une solution initialement complète.

### 2.2.3 Métaheuristiques

Ce terme est constitué de « méta » qui signifie "au-delà" et qui est exprimable par « à un plus haut niveau » et « heuristique ». Il a été cité pour la première fois par Fred Glover dans son article [67] lors de la génération de la recherche tabou. Il n'existe actuellement aucune définition communément acceptée pour les métaheuristiques. En 1996, I. H. Osman et G. Laporte [68], ont défini la métaheuristique comme un processus itératif qui guide une heuristique subordonnée en combinant d'une façon intelligente différents concepts d'exploration et d'exploitation de l'espace de recherche. Des moyens d'apprentissage sont utilisées pour organiser l'information afin de trouver efficacement des solutions optimales, Elles peuvent être classées de nombreuses façons. On peut distinguer celles qui ne manipulent qu'une seule solution à la fois et celles qui travaillent avec une population de solutions.

#### 2.2.3.1 Métaheuristiques à solution unique

Ces méthodes présentent plusieurs noms. En fait, elles peuvent être nommées méthodes de recherche locale ou bien méthodes de trajectoire. Leur principe consiste à faire évoluer itérativement une solution dans l'espace de recherche afin de se mener vers des solutions optimales. Dans ce qui suit, on donne les exemples les plus connus de ces méthodes.

##### 2.2.3.1.1 Méthode de descente

La méthode de descente (« Descent method » en anglais) peut paraitre la plus inspirée et la plus simple à comprendre. Le processus de la méthode de descente est structuré comme suit :

1) Sélectionner aléatoirement une solution initiale $s$ de $\mathcal{N}$ ;
2) Choisir une solution $s'$ dans un voisinage $\mathcal{N}(s)$ de $s$. Si cette solution est meilleure que "$s$" c-à-d $f(s') \leq f(s)$, alors on accepte cette solution comme nouvelle solution $s$.
3) On recommence le processus (2) jusqu'à ce que pour tout voisin $s'$ de $s$, $(f(s') \leq f(s)$. $s$ est alors un optimum local.

Il existe diverses méthodes de choix d'un meilleur voisin, dont le HC (Hill Climbing) qui est très utilisé et qui consiste à choisir le voisin de la solution courante ayant la meilleure qualité. Cet algorithme est une méthode qui converge assez rapidement vers

l'optimum local le plus proche et ne peut plus en sortir. Cependant, elle nécessite des améliorations pour obtenir une précision remarquable.

#### 2.2.3.1.2 Méthode du Recuit Simulé

La méthode du recuit simulé a été développée par S Kirkpatrick et al [69] en 1983, puis par V. Černy en 1985 [70]. Son origine provient de la métallurgie, où, pour atteindre les états de basse énergie d'un solide, on chauffe celui-ci jusqu'à des températures très élevées, après on le laisse refroidir lentement. Ce processus est appelé le recuit.

Dans le cadre d'un problème d'optimisation, la fonction objective à minimiser est l'énergie du matériau et les paramètres de l'algorithme sont la température du système (un paramètre fictif est contrôlé par une fonction décroissante définissant un schéma de refroidissement) et le critère de changement de palier de température.

L'algorithme de Recuit Simulé fonctionne de la manière suivante :

1) Générer une solution initiale $s$ ;
2) À chaque nouvelle itération, une solution $s'$ est générée de manière aléatoire dans le voisinage $\mathcal{N}(s)$ de la solution courante $s$ ;
3) Calculer la différence entre les valeurs de la fonction objective pour ces deux solutions: $f = f(s) - f(s')$. On se retrouve alors dans deux cas possibles :

   - $\Delta f \leq 0$, la nouvelle solution est meilleure par rapport à la solution initiale, on la remplace donc $s$ par $s'$.
   - $\Delta f > 0$, la nouvelle solution est moins bonne par rapport à la solution initiale. Cependant, on a tout de même la possibilité de la remplacer avec une probabilité d'acceptation $exp\left(\frac{\Delta f}{T}\right)$. Cette probabilité dépend de deux facteurs : d'une part l'importance de la dégradation ($\Delta f$). D'autre part, un paramètre de température $T$. Ainsi, au début de la recherche, la probabilité d'accepter des dégradations est élevée et elle diminue progressivement.

#### 2.2.3.1.3 La recherche Tabou

C'est la première méthode à avoir porté le nom de "métaheuristique". Elle a été développée par Fred Glover en 1986 [67] spécifiquement pour des problèmes d'optimisation combinatoires. Les étapes principales de cette méthode sont :

1) L'exploration de tout le voisinage de la solution courante ;
2) Le choix de la solution de ce voisinage qui minimise le critère à optimiser.

La particularité de cette méthode est l'introduction d'une mémoire des solutions explorées lors de la recherche de l'optimum. Cette dernière est nommée liste « tabou » car elle renferme les solutions déjà fréquentées et par lesquelles il est interdit de repasser. Toute nouvelle solution observée enlève de cette liste celle la plus anciennement visitée. Subséquemment, la recherche de la solution usuelle suivante se fait dans l'entourage de la solution courante sans considérer les solutions appartenant à la liste « tabou ». La taille de la liste « tabou » examine la mémoire du processus de recherche. Pour favoriser l'accroissement, il suffit de réduire la taille de la liste « tabou ». En revanche, l'augmentation de la taille de la liste « tabou », obligera le processus de recherche à parcourir des régions plus larges, favorisant ainsi la diversification.

### 2.2.3.2 Métaheuristiques à population de solutions

Dans ce qui suit seront présentés quelques algorithmes appartenant à cette catégorie.

#### 2.2.3.2.1 Algorithmes évolutionnaires

Les algorithmes évolutionnaires (AEs) [71] appartiennent à une famille de méthodes s'inspirant de la théorie de l'évolution par la sélection naturelle appelée la théorie « Darwinienne », énoncée par Charles Darwin en 1859 [72]. Le principe fondamental étant que les individus les plus adaptés à leur milieu survivent et peuvent se reproduire, laissant une descendance qui communiquera leurs gènes.

Le terme AEs englobe une classe assez large de métaheuristiques telles que les AGs [30], la programmation génétique [73], les stratégies d'évolution [74] et la programmation évolutive [75].

Cette section présente brièvement les quatre classes d'AEs.

#### 2.2.3.2.1.1 Algorithmes Génétiques

La technique des AGs est la plus populaire et est plus largement utilisée par rapport aux autres méthodes appartenant à la catégorie des AEs. Elle débute à partir des années 1970, avec les travaux de John Holland et ses élèves à l'Université du Michigan sur les systèmes adaptatifs [6]. L'ouvrage de référence de David E. Goldberg [7] a fortement participé à leur essor. Au fait, ce sont des algorithmes stochastiques qui se basent sur une simulation du processus de sélection naturelle : les individus les plus forts d'une espèce ont plus de chance de survivre que les plus faibles.

Les AG fonctionnent de la manière suivante :

1) Initialisation : L'algorithme commence avec une population (ensemble de solutions candidates) initiale de $N$ individus générés d'une façon aléatoire ;
2) Évaluation : on apprécie chaque individu par la fonction objective ou fitness ;
3) Sélection : on détermine les individus de la génération P qui vont être reproduits dans la nouvelle population ;
4) Reproduction : on utilise des opérateurs génétiques pour créer la nouvelle génération. Les opérateurs de croisement engendrent un ou plusieurs enfants à partir de combinaisons de deux parents tandis que les opérateurs de mutation modifient un individu pour en former un autre.

Dans ce qui suit, nous allons expliquer les points concernant le processus d'optimisation par les AGs.

**1. Initialisation**

L'initialisation sert à la création de la population initiale et le codage.

**a. Création de la population initiale**

La résolution d'un problème par débute avec la génération de la population initiale. Le choix de la population initiale est important car il a une influence sur la rapidité de convergence vers l'optimum global. Si la position de la solution optimale dans l'espace de recherche est totalement inconnue, il est naturel de générer aléatoirement une population initiale afin d'explorer une grande partie de l'espace de recherche. Si par contre, une certaine connaissance du problème est disponible, il paraît évident de générer les individus dans un espace particulier afin d'accélérer la convergence. Concernant la taille de la population, elle a un effet direct sur la convergence, sur la qualité des solutions obtenues et

sur la performance de l'algorithme. En effet, une taille de population trop grande augmente le temps de calcul et nécessite un espace mémoire considérable, alors qu'une taille de population très petite ne permet pas une bonne exploration de l'espace de recherche et peut mener à une convergence prématurée.

### b. Codage des données

Pour ajuster les AGs à un problème donné, il faut au préalable déterminer un procédé de codage. Le choix de ce dernier est l'élément le plus important dans la conception de l'algorithme puisqu'il influe sur la mise en œuvre des opérations génétiques.

## 2. Evaluation

L'évaluation sert, comme son nom l'indique, à mesurer le degré d'adaptation des individus à leur milieu. Elle associe ainsi une valeur à chaque individu de la population pour mettre de faire des comparaisons entre les différentes solutions. Dans la suite du document, le terme 'fonction fitness' est utilisé pour désigner la fonction $f$.

## 3. Sélection

La sélection sert à désigner les chromosomes aptes à se reproduire donc les chromosomes responsables de la création de la nouvelle génération. Cet opérateur utilise la fonction fitness comme critère de sélection. Dans ce qui suit, nous présentons quelques types du mécanisme de sélection.

### a. Sélection par roulette Wheel

La sélection par roulette RWS (Roulette Wheel Selection), introduite par Goldberg en 1989, est une méthode stochastique qui s'inspire des roues de loterie (roulette de casino). Elle consiste à associer à chaque individu de la population un secteur d'une roue dont l'angle est proportionnel à la fonction fitness. La roue étant tournée, l'individu sélectionné est celui sur lequel la roue s'est arrêtée. Cette méthode favorise les meilleurs individus, mais tous les individus conservent néanmoins des chances d'être sélectionnés.

Si on considère la population $P = \{x_1, x_2, .., x_N\}$, alors la probabilité $p_i$ de sélection d'un individu $x_i$ est donnée par $p_i = \frac{f(x_i)}{\sum_{j \in P} f(x_j)}$, où f représente le fitness de l'individu $x_i$ et $N$ est la taille de la population.

### b. Sélection par tournoi

La sélection par tournoi fait intervenir le concept de comparaison entre les individus et ne nécessite pas le tri de la population. Elle possède un paramètre $q$ appelé taille du tournoi. Son principe est de choisir un groupe de $q$ individus aléatoirement dans la population $n$ et de sélectionner d'une manière déterministe le meilleur dans ce groupe qui servira de parent. On reproduit ce processus autant de fois que nécessaire pour l'obtention des $n$ nouveaux individus de la $2^{ème}$ génération.

### c. Sélection par rang

Le principe de cette méthode est similaire à celui de la sélection par roulette. Cependant, dans la sélection par rang les secteurs de la roue ne sont plus proportionnels à la qualité des individus, mais à leur rang dans la population. Il s'agit de classer la population en fonction de la qualité des individus puis leur attribuer à chacun un rang. Le plus mauvais individu aura le premier rang (à partir de 1), le suivant le deuxième rang, et ainsi en itérant sur chaque individu on finit par attribuer le rang $N$ au meilleur individu (où $N$ est la taille de la population). L'angle de chaque secteur de la roue sera proportionnel au rang de l'individu qu'il représente.

### d. Sélection élitiste

Cette une méthode de sélection déterministe consiste à copier un ou plusieurs des meilleurs chromosomes dans la nouvelle génération. Postérieurement, on génère le reste de la population selon un algorithme de reproduction usuel. Ce procédé présente un effet positif sur les AGs.

## 4. Croisement

Le croisement a pour objectif d'enrichir la diversité de la population en manipulant les gènes des individus. Le croisement consiste à sélectionner deux individus parmi les parents potentiels, aléatoirement ou à l'aide d'une méthode de sélection, pour donner naissance à un ou plusieurs (généralement deux) individus appelés "enfants", dont le code génétique est une combinaison de ceux des parents. Ces derniers sont croisés suivant une probabilité $p_{crois}$ appelée probabilité de croisement. Il existe plusieurs méthodes de croisement. Le croisement de deux chromosomes peut se faire en un seul point de coupure ou en multi points.

### a. Croisement à un point

L'opérateur de croisement classique à un point est le croisement le plus simple et le plus connu dans la littérature. Il consiste à choisir de façon uniforme une position de coupure aléatoire à partir de laquelle les partants échangent leurs gènes. Si les parents $P_1$ et $P_2$ ont été sélectionnés pour la reproduction, les solutions enfants $E_1$ et $E_2$ sont alors construites de la façon indiquée sur la Figure 2-2.

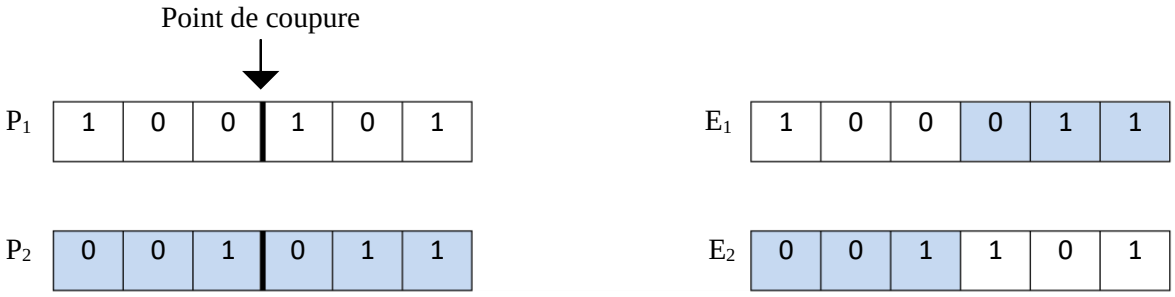



**Figure 2-2 Croisement à un point.**

### b. Croisement multi points

Ici, les enfants sont produits en échangeant l'information chromosomique entre les k points de coupure. En pratique, la valeur de k=2 est la plus communément utilisée et est appelée croisement à deux points. La Figure 2-3 présente un exemple pour le croisement à deux points où les chromosomes sont coupés entre le deuxième et le quatrième emplacement.

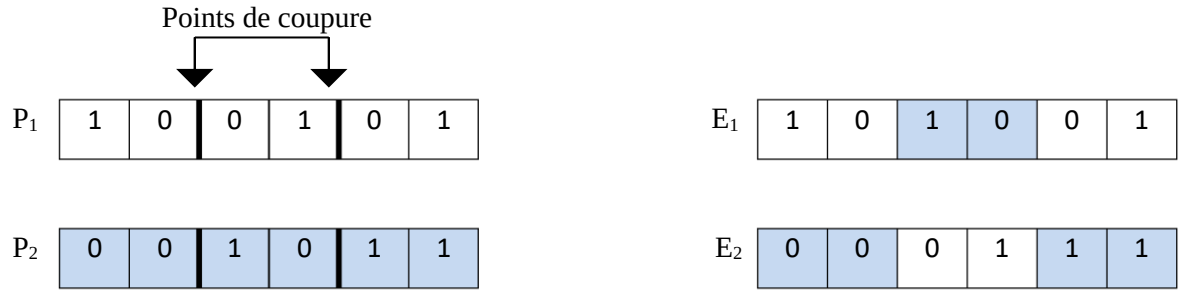



**Figure 2-3 Croisement à 2- points.**

## 5. Mutation

Dans cette étape, un composant d'un individu est modifié aléatoirement avec une certaine probabilité très faible.

Il existe de nombreuses variantes de mutation qui ont été adaptées aux problèmes d'optimisation topologique telle que la mutation uniforme ou la mutation générale.

Chaque bit $a_i \in \{0,1\}$ est remplacé, selon une probabilité $p_m$ , par son complémentaire $a_i = 1 - \overline{a_i}$. Dans l'exemple de la Figure 2-4, une mutation a eu lieu sur le troisième gène du chromosome et elle a transformé ce gène de 1 en 0.

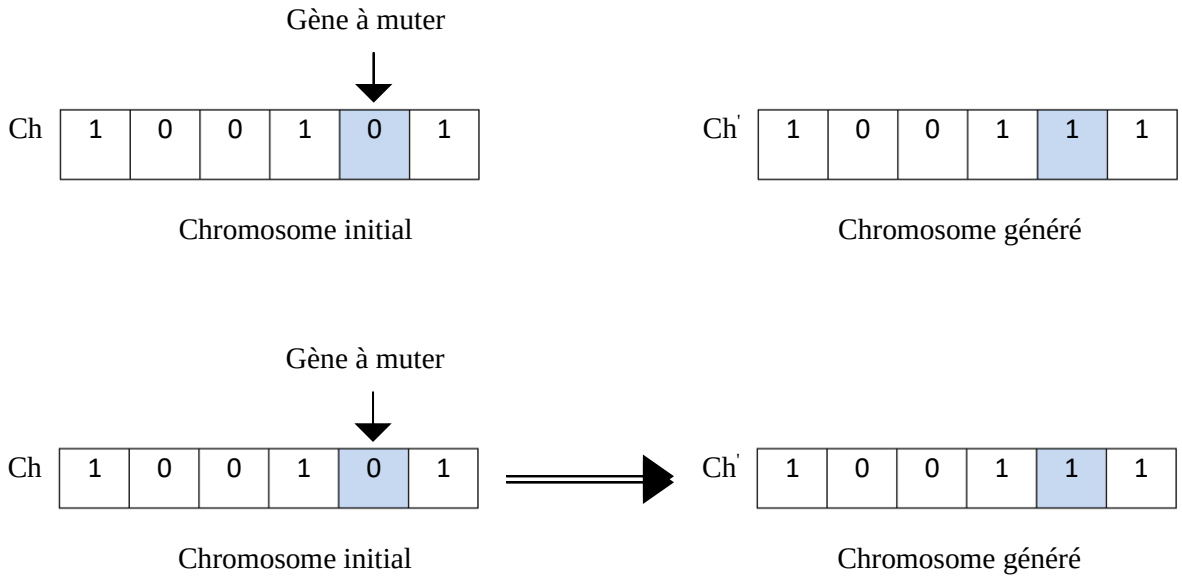



**Figure 2-4 Principe de l'opérateur de mutation.**

### 6. Critère d'arrêt

Plusieurs critères d'arrêt de l'algorithme sont possibles :

- Arrêt après un nombre de générations fixé à priori ;
- Arrêt lorsque la population cesse d'évoluer ou en présence d'une population homogène (stagnation de la population).

#### 2.2.3.2.1.2 Programmation génétique

La programmation génétique (PG) a été initialement développée par Koza au début des années 1990 [73]. Elle est apparue initialement comme sous domaine des AGs, car elle n'en diffère que par la représentation des individus. Effectivement, contrairement aux AGs, les individus pour cette famille sont des programmes qui sont représentés sous forme d'arbre, c'est-à-dire des graphes orientés et sans cycle, dans lesquels chaque nœud est associé à une opération élémentaire relative au domaine du problème.

La représentation la plus courante utilise des arbres dans le langage LISP. La Figure 2-5 présente un exemple de deux arbres correspondants, en ce langage, aux expressions : « $(x * 5) + x$ » et « $fact(n)$ ».

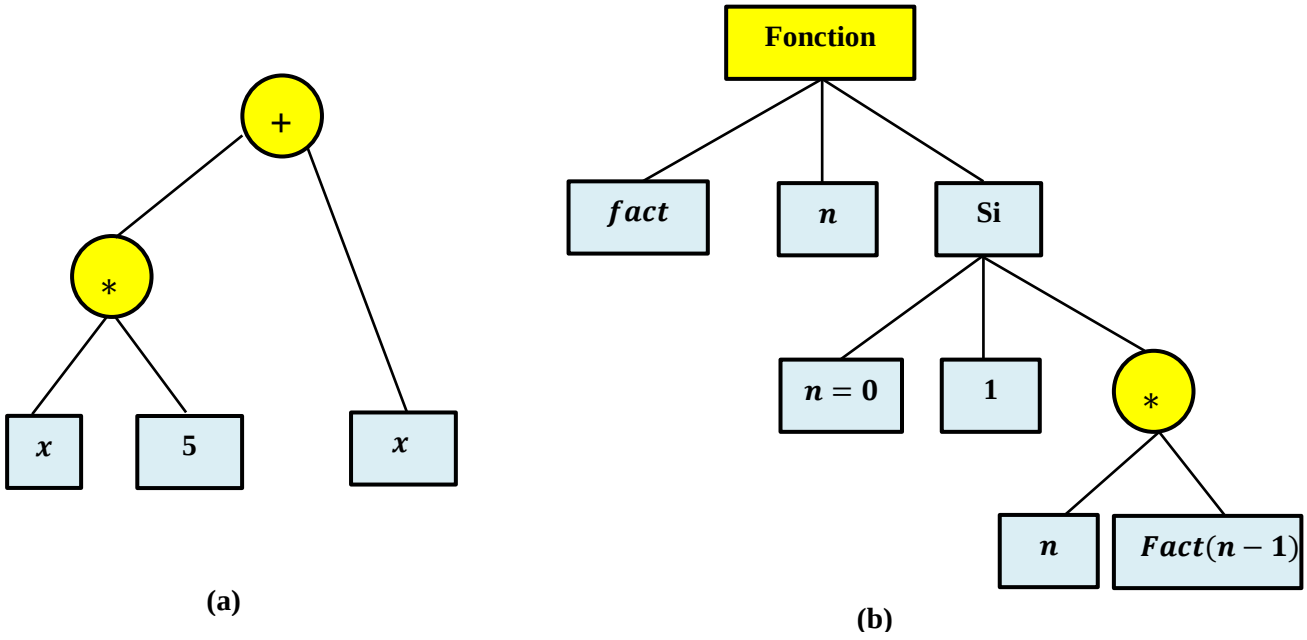


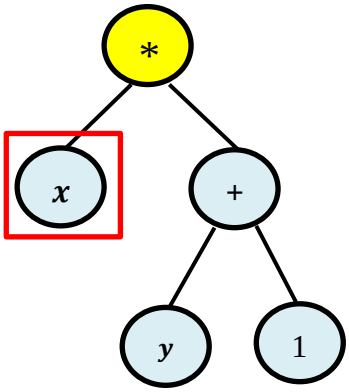

**Figure 2-5 Exemple de la représentation en arbre de PG**

Ce graphe est constitué de deux types de nœuds reliés par des arcs, les nœuds internes comme ($*$, Si, $fact$) sont des nœuds représentant les fonctions et les feuilles comme ($x, 5, n, fact(n-1)$) représentant les terminaux ; ces nœuds ne peuvent en effet avoir des fils. L'ensemble des fonctions choisies dépend de la nature du domaine du problème. Il existe par exemple : l'ensemble de fonctions arithmétiques (+, -, *, /), mathématiques (sin, cos, exp), booléennes (AND, OR, NOT), conditionnelles (IF-THEN-ELSE) et boucles (FOR, Do- Until). En général, les terminaux sont des variables (X, Y, Z), constantes (1, 8,..) ou fonctions utilisées comme entrées pour le programme.

Dans le cadre de la programmation génétique, au début, la population est composée de programmes engendrés aléatoirement. Chaque programme est estimé selon une méthode propre au problème posé. A chaque génération, on classe les programmes en fonction des notes obtenues. On crée après une nouvelle population, où les meilleurs programmes auront une plus grande chance de rester. Ce principe est le même que celui utilisé dans l'algorithmique génétique classique. La différence est que les opérateurs de croisement et de mutation sont différents. Le croisement dans ce cas met en jeu deux parents ; un sous

arbre est choisi dans chacun des parents, et ces sous arbres sont échangés. Les Figures 2-6 et 2-7 montrent un exemple de cette opération.

L'expression : $x * (y + 1)$
L'arbre associé :




L'expression : $x + (y - abs(z))$
L'arbre associé :

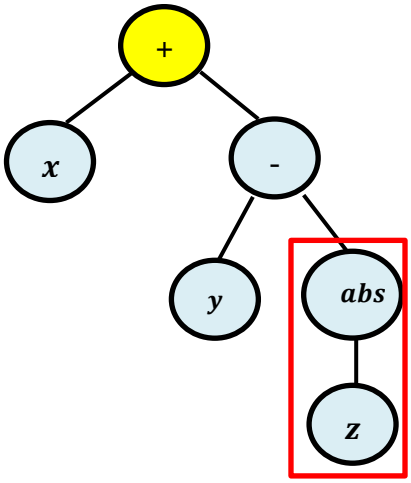
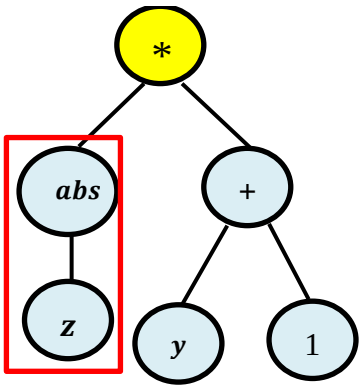



**Figure 2-6 Programmes sélectionnés**

L'expression : $abs(z) * (y + 1)$
L'arbre associé :




L'expression : $x + (y - x)$
L'arbre associé :

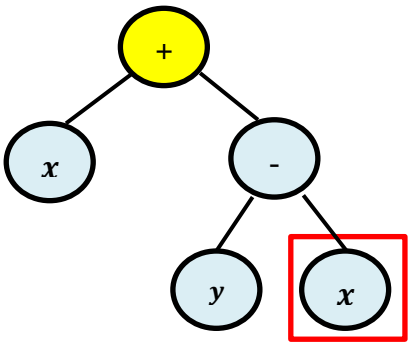
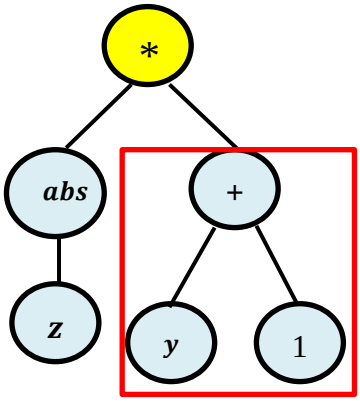



**Figure 2-7 Programmes croisés**

La mutation classique, ou structurelle, permet une exploitation plus fine de l'espace de recherche en comparaison avec l'opérateur de croisement. Comme la montre la Figure 2-8, la forme la plus utilisée de cet opérateur consiste à tirer un point au hasard puis modifier ce nœud par un nouveau sous arbre. Cette mutation est appliquée parfois comme un croisement entre un arbre est un nouvel arbre généré arbitrairement. On nomme cette opération le «headless-chicken» [76].

L'expression : $abs(z) * (y + 1)$

L'arbre associé :




L'expression : $abs(z) * cos(y)$

L'arbre associé :

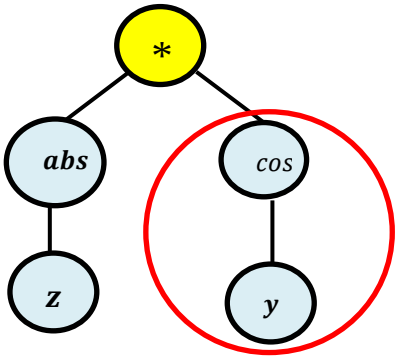



**Figure 2-8 Mutation d´un programme où le nœud $(y + 1)$ est remplacé par le sous arbre généré aléatoirement dans le cercle.**

#### 2.2.3.2.1.3 Stratégies d'évolutions

Dans cette famille de méthodes, historiquement la première parmi tous les AEs [74], les stratégies d'évolutions utilisent un ensemble de μ individus de la population pour produire $\lambda$ « enfants ». Pour créer l'enfant, à chaque itération, ρ individus sont sélectionnés et croisés. Une fois produits, les enfants sont après changés par des mutations qui consistent à additionner une valeur engendrée aléatoirement selon une fonction de densité de probabilité paramétrée. Les paramètres de la fonction de densité probabilité sont nommés les paramètres de la stratégie.

Pour former la nouvelle population, l'étape de sélection est appliquée, soit :

- Uniquement aux enfants, (stratégie ",") dans ce cas l'algorithme est noté $(\mu/\rho, \lambda)$-ES ;
- Soit à l'ensemble des parents et enfants (stratégie "+"), dans ce cas l'algorithme est noté $(\mu/\rho + \lambda)$-ES.

L'algorithme général fonctionne comme suit :

1) On génère une population de parents de taille $\mu$ de façon aléatoire ;
2) On génère une population d'enfants de taille $\geq \mu$ , en suivant les étapes suivantes :
    1. Choisir ρ parents au hasard ;
    2. Recombiner ces parents entre eux pour former un unique individu ;

3. Faire muter cet individu ;
4. Évaluer la valeur de la fonction objective associée à cette nouvelle solution ;
5. Insérer la solution dans la population des enfants ;

3) Sélectionner les μ meilleures solutions parmi les enfants (stratégie ",") ou parmi les parents et enfants (stratégie "+").

#### 2.2.3.2.1.4 Programmation évolutionnaire

La programmation évolutionnaire (PE), introduite par Fogel [75], a été originellement créée pour faire évoluer les machines à états finis. Dans les années 1990, elle a été utilisée par D.B. Fogel pour résoudre des fonctions plus générales [77]. Cette approche met la recherche sur la relation entre les parents et leurs descendants plutôt que de simuler des opérateurs génétiques d'inspiration naturelle. En effet, la PE n'utilise pas une reproduction significative mais plutôt un modèle évolutionnaire jumelé à une représentation des solutions et à un opérateur de mutation choisi en fonction du problème à résoudre.

Pour effectuer la PE les étapes suivantes sont effectuées :

1) On génère une population de taille $\mu$ de façon aléatoire ;
2) Chaque individu de la population va générer $\lambda$ enfants suite à l'application de l'opérateur de mutation ;
3) Une opération de sélection naturelle (sélection de tournois) permet de créer une nouvelle génération de taille $\mu$ à partir des parents et des enfants ;
4) Le processus de mutation/sélection est répété de manière itérative jusqu' à ce qu'une solution approuvable soit trouvée.

Les principales caractéristiques de ces algorithmes par rapport aux autres AEs, étaient, à l'origine, le fait qu'ils n'utilisaient pas de croisement.

Cette classe d'AE est la moins utilisée, comparée aux autres algorithmes de la même famille, car elle ressemble à un ES, bien que les deux approches se soient développées indépendamment.

#### 2.2.3.2.2 Algorithmes d'intelligence en essaim

L'intelligence en essaim est basée sur l'aspect général des systèmes autoorganisés. Les schémas typiques d'intelligence en essaim incluent l'OEP, la recherche par diffusion stochastique SDS (Stochastic Diffusion Search), le système de colonies de fourmis ACS (Ant Colony System), la colonie d'abeilles artificielle ABC (Artificial Bee Colony), la recherche de nourriture pour les bactéries BF (Bacteria Foraging), …etc. Dans ce qui suit, quelques exemples d'algorithmes d'intelligence en essaim seront présentés.

#### 2.2.3.2.2.1 Méthode de colonies de fourmis

Cette méthode, conçue par Dorigo [78], s'inspire du comportement des fourmis lorsque celles-ci recherchent de la nourriture. En effet, les fourmis utilisent leur milieu pour communiquer entres elles. Il s'agit d'un mécanisme dît stigmergique grâce auquel certaines fourmis entreposent des phéromones sur le sol pour signifier aux autres le chemin parcouru pour atteindre la nourriture. Ainsi, les autres fourmis sauront suivre la piste de phéromones pour recouvrer la source de nourriture. Malgré cela, les phéromones s'évaporent avec le temps et les chemins les plus courts seront certainement ceux qui auront une intensité de phéromones plus robuste. Ce qui, au final, mènera automatiquement les fourmis vers le chemin le plus court.

Dans un algorithme d'ACS, un problème donné peut être modélisé dans un graphe, dans lequel les fourmis se déplacent le long de toutes les branches d'un nœud à l'autre et construisent ainsi des chemins représentant des solutions.

Le premier algorithme appartenant à cette catégorie a été proposé par Colorni, Dorigo et Maniezzo [78] dans le but de régler le problème du voyageur de commerce. Ce dernier consiste à trouver le plus court cycle hamiltonien dans un graphe (chaque sommet du graphe représente e effet une ville). La distance entre deux villes $i$ et $j$ est représentée par $d_{ij}$, et le couple $(i, j)$ représente l'arête entre ces deux villes. Chaque fourmi parcourt le graphe et construit un trajet complet. A chaque étape de la construction des solutions, la fourmi doit décider à quel sommet elle va se déplacer, cette décision est prise d'une manière probabiliste fondée sur les valeurs de phéromone et d'une information statistique qui permet notamment de trouver une bonne solution.

La règle de déplacement (probabilité) pour chaque fourmi déplacée d'une ville $i$ à une ville $j$ est la suivante :

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{(\tau_{ij}(t))^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in J_i^k}(\tau_{il}(t))^{\alpha}(\eta_{il})^{\beta}} & si\ j \in J_i^k \\ 0 & si\ j \notin J_i^k \end{cases} \quad 2\text{-}1$$

Où :

- $J_i^k$ : liste des déplacements possibles, quand la fourmi $k$ est sur la ville $i$ ;
- $\eta_{ij} = \frac{1}{d_{ij}}$ : appelée visibilité, c'est l'inverse de la distance entre deux villes $i$ et $j$. Cette information est utilisée pour diriger les fourmis vers des villes proches et, ainsi, éviter de trop longs déplacements ;
- $\tau_{ij}(t)$ : appelée intensité de la piste, c'est la quantité de phéromones déposée sur l'arête reliant deux villes $i$ et $j$. Cette quantité définit l'attractivité d'une piste et est modifiée après le passage d'une fourmi ;
- $\alpha$ et $\beta$ sont deux paramètres contrôlant l'importance relative de l'intensité et de la visibilité.

Après un tour complet, chaque fourmi dépose une quantité de phéromones $\Delta\tau_{ij}(t)$ sur l'ensemble de son parcours. Cette quantité dépend de la qualité de la solution trouvée et est définie par :

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L^k(t)} & si\ (i,j) \in T^k(t) \\ 0 & sinon \end{cases} \quad 2\text{-}2$$

où

- $T^k(t)$ est le trajet effectué par la fourmi $k$ à l'itération $t$ ;
- $L^k(t)$ est la longueur de $T^k(t)$ ;
- $Q$ est un paramètre fixé.

L'ajout de la quantité de phéromones dépend sûrement de la qualité de la solution obtenue.

Le processus de mise à jour des traces de phéromones est généralement implémenté comme suit :

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad 2\text{-}3$$

Où :

- $\rho$ : un paramètre appelé le taux d'évaporation ($0 < \rho < 1$). Pour ne pas négliger toutes les mauvaises solutions obtenues, et ainsi éviter la convergence vers des optimums locaux de mauvaise qualité, il est nécessaire d'introduire un processus d'évaporation des phéromones à travers ce paramètre ;
- $\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$ et $m$ est le nombre de fourmis.

#### 2.2.3.2.2.2 Optimisation par essaim de particules

L'OEP est une technique d'optimisation développée par Kennedy et Eberhart en 1995 [8]. Cette méthode est inspirée du comportement social des animaux évoluant en essaim tels que les nuées d'oiseaux et les bancs de poissons. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors que séparément chaque individu a une « intelligence » limitée, et ne dispose que d'une connaissance locale de sa situation dans l'essaim. L'OEP repose sur un ensemble de solutions ou individus disposés aléatoirement au début de l'algorithme. Chaque individu de la population est dit (particule), tandis que la population est connue sous le nom d'essaim. Chaque particule se situe à une position donnée dans l'espace de recherche et se déplace selon une certaine vitesse. Le mouvement de la particule est influencé par sa propre expérience et l'expérience des particules voisines. Les particules de l'essaim communiquent entre elles pour construire une solution à un problème, en s'appuyant sur leur expérience collective.

De plus, chaque particule possède une mémoire lui permettant de se souvenir de sa meilleure performance, c'est à dire sa qualité ainsi que la meilleure position visitée. Cette valeur est appelée meilleur local « Local best (Lbest) ». Une autre meilleure valeur obtenue par les particules voisines de l'essaim est appelée global best (Gbest). La performance de chaque particule est mesurée au moyen d'une fonction objective « fitness » $f$ qui varie avec le problème d'optimisation. La Figure 2-9 illustre la stratégie de déplacement d'une particule.

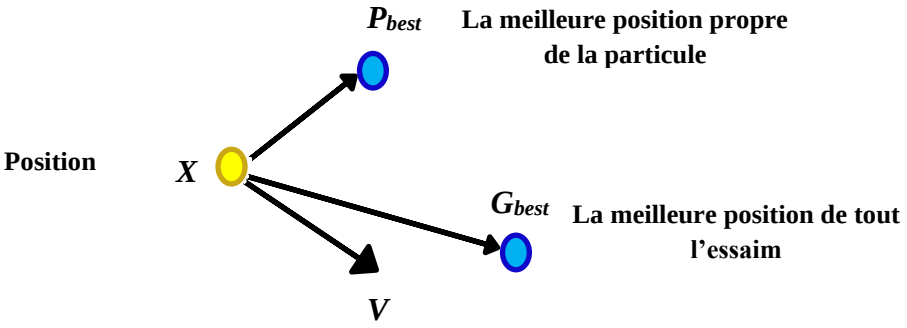



**Figure 2-9 Déplacement d'une particule.**

L'OEP commence avec un essaim de particules qui se déplacent dans l'espace de recherche de dimension $n$ pour trouver la solution optimale. Chaque particule est considérée comme une solution potentielle du problème d'optimisation. Chaque particule $i$ est composée des trois vecteurs suivants :

- $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ qui représente la position actuelle de la particule $i$ ;
- $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ qui représente les vitesses de la particule $i$ ;
- $Pbest_i = (Pbest_{i1}, Pbest_{i2}, \ldots, Pbest_{in})$ qui est la meilleure position personnelle de la particule $i$ ;

La meilleure position atteinte par toutes les particules de l'essaim est indiquée par un vecteur donné par :

- $Gbest = (Gbest_1, Gbest_2, \ldots, Gbest_n)$.

Avec ces notations, les équations de mouvement d'une particule sont, pour chaque itération $t$, données par :

$$v_{ij}^{t} = w * v_{ij}^{t-1} + c_1 r_1 \left(Pbest_{ij}^{t-1} - xi^{t-1}\right) + c_2 r_2 \left(Gbest_{ij}^{t-1} - xi^{t-1}\right) \qquad j \in \{1, .., n\} \qquad 2.4$$

Et

$$x_{ij}^{t} = v_{ij}^{t-1} + v_{ij}^{t} \qquad 2.5$$

Avec

- $v_i(t)$: la vitesse de la particule à l'itération t ;
- $w$ est en général une constante appelée, coefficient d'inertie ;
- $c_1$ et $c_2$sont deux constantes appelées coefficients d'accélération ;
- $r_1$ et $r_2$ sont deux nombres aléatoires tirés uniformément dans [0,1] à chaque itération et pour chaque dimension ;
- La première partie de l'Equation 2-4 représente la composante d'inertie de la vitesse précédente contrôlée par le paramètre $w$. La deuxième partie est la

composante cognitive qui implique la pensée personnelle de la particule et qui est contrôlée par le paramètre $c_1$. La troisième partie représente la coopération entre les particules et est donc nommée composante sociale contrôlée par le paramètre $c_2$.

#### 2.2.3.3 Métaheuristiques avancées

L'idée de combiner des méthodes exactes et/ou des méthodes approchées pour créer de nouvelles méthodes a donné naissance des méthodes hybrides. Un type des méthodes, appartenant à cette catégorie, sera présenté dans ce qui suit.

##### 2.2.3.3.1 Algorithmes mimétiques

Moscato [79] a introduit en 1989, pour la première fois, les algorithmes mimétiques. On rencontre ces méthodes aussi sous le nom des AGs hybrides, de recherche locale génétique ou celui d'AE Baldwiniens [79]. Ils appartiennent à la famille des méthodes hybrides qui combine les algorithmes de recherche locale et les AGs.

L'hybridation permet de tirer profit des avantages cumulés de différentes métaheuristiques.

Le schéma général de l'algorithme mimétique est le suivant :

1) Construire une population initiale de $N$ individus de manière aléatoire ;
2) Appliquer l'opérateur de sélection ;
3) Appliquer l'opérateur de croisement ;
4) Appliquer l'opérateur de recherche locale aux nouveaux individus créés (enfants) pendant n itérations et renvoyer les meilleurs individus trouvés (Population des minimums locaux) ;
5) Appliquer l'opérateur de mutation utilisé pour introduire de nouveaux gènes C dans la population ;
6) Mettre à jour, par remplacement de $P$ (le plus mauvais parent) par $C$. Le critère d'arrêt est différent d'un problème à un autre.

## 2.3 Etat de l'art sur les méthodes de construction des séquences binaires

Les séquences, possédant de bonnes propriétés en termes de la fonction de corrélation, sont particulièrement intéressantes pour la réalisation d'un système CDMA performant. Il existe de nombreuses familles de séquences, présentées dans la littérature, et plusieurs études ont été menées pour étudier leurs performances suivant différents critères tels que le niveau des pics secondaires de la FAC. Dans cette section, nous donnons un aperçu sur les plus importants travaux publiés qui traitent ce type de problème.

Afin de mesurer l'effet des pics secondaires de la FAC d'une séquence S de longueur N, les chercheurs ont proposé diverses métriques parmi lesquelles nous avons:

1. Le niveau maximum des pics secondaires $PSL$ (Peak Sidelobe Level) de la FAC définit par:

$$PSL = max_{k=1,2,\dots,N-1}\{|\mathrm{R_S(k)}|\} \quad 2\text{-}6$$

2. Le niveau des pics secondaires intégrés $ISL$ (Integrated Sidelobe Level) définit par :

$$ISL = \sum_{k=1}^{N-1} |\mathrm{R_S(k)}|^2 \quad 2\text{-}7$$

3. Le $FM$ définit dans le chapitre 1 et qui peut être donné en fonction de l'$ISL$ comme suit :

$$FM = \frac{\mathrm{N}^2}{2\ ISL} \quad 2\text{-}8$$

4. La métrique $ISL$ pondérée $WISL$ (Weighted ISL) définit par:

$$WISL = \sum_{k=1}^{N-1} w_k |\mathrm{R_S(k)}|^2 \quad 2\text{-}9$$

Où $w_k$ $(k = 1, \dots, N-1)$ sont les poids non négatifs. Cette métrique est utilisée lorsque les $R_S(k)$ ne sont pas importants et que nous voulons particulièrement supprimer certains d'entre eux.

Les séquences, avec de bonnes propriétés de la FAC, peuvent être obtenues en minimisant la métrique PSL ou ISL (notez que la minimisation de la métrique ISL est équivalente à la maximisation du FM).

Ci-dessous, nous présentons les principaux résultats de la recherche globale de séquences binaires optimales avec un PSL minimal (Tableau 2-1) :

**Tableau 2-1 Principaux résultats de la recherche globale de séquences binaires optimales avec un PSL minimal**

| Longueur de la séquence | Année | Référence |
|---|---|---|
| $N \leq 40$ | 1975 | [80] |
| $N \leq 48$ | 1990 | [81] |
| $N = 64$ | 2005 | [82] |
| $N \leq 68$ | 2012 | [83] |
| $N \leq 74$ | 2013 | [84] |
| $N \leq 80$ | 2014 | [85] |
| $N \leq 82$ | 2015 | [86] |
| $83 \leq N \leq 88$ | 2017 | [87] |

Les meilleures valeurs, actuellement connues, du PSL pour $85 \leq N \leq 105$ sont publiées dans [88] et pour $106 \leq N \leq 300$ dans [89].

Dans cette dernière référence, les séquences de longueur $N$ appartenant à l'intervalle $106 \leq N \leq 300$ avec un $PSL$ égal à 5 sont construites par un algorithme simple basé sur la méthode de descente HC.

Dans la référence [90], A. Bose et M. Soltanalian ont proposé une méthode de construction des séquences binaires avec un PSL de croissance optimale à partir des ensembles de séquences ayant de bonnes propriétés de corrélation telles que les séquences de Gold, de Kasami, de Legendre et de Weil. La méthode proposée construit les séquences binaires à travers un programme quadratique non convexe qui peut être manipulé en temps polynomial.

Dans [91], les auteurs ont proposé un algorithme, permettant de construire de longues séquences binaires à faible FAC LABS (Low Autocorrelation Binary Sequences) définit en termes du PSL, de longueurs allant jusqu'à plusieurs milliers. L'algorithme proposé est un EA avec une nouvelle combinaison de nombreuses fonctionnalités inspirées des AGs, des stratégies d'évolution, et des algorithmes mimétiques.

En raison de l'importance pratique et des applications répandues des séquences avec de bonnes propriétés de la FAC, en particulier avec des valeurs ISL faibles ou des valeurs FM élevées, beaucoup d'efforts dans la littérature ont été consacrés à la conception de ces séquences via des méthodes de construction analytiques ou des approches informatiques telles que la recherche exhaustive [92], les métaheuristiques, les algorithmes d'optimisations d'heuristique [93] et les méthodes d'optimisation stochastique [94]. Cependant, ces méthodes ne sont généralement pas capables de concevoir de longues séquences, disons pour $N \sim 10^3$ ou plus, en raison de la complexité de calcul croissante. Récemment, un certain nombre d'approches, axées sur l'optimisation, ont été proposées pour concevoir de longues séquences avec faibles niveaux des pics secondaires de la FAC. Parmi celles, on trouve l'algorithme cyclique CA (Cyclic Algorithm) qui, tout d'abord, a été proposé par Li, Stoica et Zheng dans [95] pour concevoir des séquences satisfaisant la métrique ISL.

Par la suite, Stoica et al dans [96,97] ont développé deux extensions de l'algorithme CA, appelées CAN (CA New) et PeCAN (Periodic-correlation CAN) pour générer des séquences de longueur $N \sim 10^6$ dans le cas d'ISL périodique et apériodique, respectivement. Par rapport au CA, le CAN et le PeCAN utilisent les opérations de transformation de Fourier rapide FFT (Fast Fourier Transform) et réduisent efficacement le temps de calcul.

Dans [98], les auteurs ont proposé une méthode de calcul efficace pour construire des séquences avec une bonne FAC et de bonnes propriétés de distribution. La méthode proposée est basée sur l'utilisation de l'algorithme CAN introduit dans [96] qui est basé sur la FFT. La méthode proposée est efficace en termes de calcul et peut concevoir des séquences très longues (de longueurs allant jusqu' à $N \sim 10^6$ et même plus) en un temps relativement court.

Dans [99], M. Soltanalian, Stoica et al ont proposé une extension de l'algorithme CAN pour l'ensemble des séquences complémentaires (appelé CANARY). L'algorithme CANARY, qui fonctionne dans le domaine fréquentiel, est utilisé pour construire des séquences binaires apériodiques avec de bonnes propriétés de la FAC (en particulier avec des valeurs d'ISL faibles).

Dans [100], R Lin, M. Soltanalian, et al ont proposé deux algorithmes 1bCAN et 1bPeCAN qui combinent la notion de fonctions convergentes avec les algorithmes CAN et

PeCAN [96,97], pour améliorer considérablement leurs performances pour la conception de très longues séquences binaires de longueurs arbitraires (jusqu'à $N \sim 10^{6}$ ou même plus). De plus, ils ont analysé leurs propriétés de convergence. L'algorithme 1bCAN montre une performance supérieure en termes de FM pour la conception des séquences binaires apériodiques par rapport aux algorithmes CANARY [99]. Cependant, toutes les séquences obtenues ne sont pas valables pour les applications basées sur la FAC périodique.

Sur la base d'itérations alternées [101], M. Soltanalian et al ont proposé un algorithme d'approximation itérative torsadée ITROX (Iterative Twisted Approximation). L'algorithme ITROX peut être utilisé pour concevoir des séquences binaires avec de faibles niveaux des pics secondaires de la FAC pour les cas périodiques et apériodiques.

Dans [102], L. Zhao et al ont présenté une métrique standard pour concevoir des séquences à faible FAC. La métrique standard inclut le niveau d'ISL, le WISL et le niveau du PSL. La technique d'optimisation utilisée est l'algorithme de Majoration-Minimisation (MM), qui est un algorithme itératif et bénéficie d'une convergence garantie vers une solution stationnaire. Cependant, cette méthode est limitée par la taille du réseau nécessaire pour la méthode qui est un paramètre très sensible.

Dans [103], les auteurs ont proposé une autre façon pour construire un ensemble de séquences binaires avec de bonnes FAC et de FIC apériodique / périodique pour les systèmes Radar MIMO (Multiple-Input Multiple-Output). La méthode proposée est basée sur la minimisation de la somme du niveau du PSL et du niveau de l'ISL et utilise la méthode de descente par coordonnée CD (Coordinate Descent) pour minimiser la fonction objective multidimensionnelle.

Comme nous l'avons déjà vu, le FM est une mesure importante pour déterminer la qualité de la séquence. Dans ce contexte, plusieurs études dans la littérature ont été menées pour trouver des séquences binaires à faibles FAC avec des FMs optimaux.

Dans le Tableau 2-2, nous présentons les principaux résultats de la recherche globale de séquences binaires optimales basée sur le $FM$ :

**Tableau 2-2 Principaux résultats de la recherche globale de séquences binaires optimales basée sur le FM**

| Longueur de la séquence | Année | Référence |
|---|---|---|
| $\mathbf{2 \leq N \leq 6}$ | 1965 | Lunelli [104] |
| $\mathbf{7 \leq N \leq 19}$ | 1966 | Littlewood [105] |
| $\mathbf{N \leq 32}$ | 1982 | Golay [106] |
| $\mathbf{N \leq 48}$ | 1996 | Mertens [92] |
| $\mathbf{N \leq 60}$ | 2004 | Mertens et Bauke [107] |

Dans ce même contexte, T. Packebusch et S. Mertens [108] ont présenté un nouvel algorithme pour générer des séquences optimales de longueur $N \leq 66$. Pour se faire, les auteurs de cette référence ont utilisé l'algorithme B&B pour calculer toutes les séquences avec un FM maximal.

Un autre type de séquences qui présentent de très bonnes performances en terme de la FAC et qui ont une grande complexité linéaire sont les séquences de Legendre. En 1983, Golay [109] a fourni la valeur asymptotique du FM de la rotation d'une séquence de Legendre. Cinq ans plus tard, Hoholdt et Jensen [110] ont montré que la valeur obtenue par Golay est correcte. Kristiansen et Parker [111], dans leur travail, ont montré que les séquences de Legendre avec rotation périodique peuvent atteindre un FM de 6,34. B. Suribabu Naick et al ont appliqué ces séquences pour générer de longues séquences binaires. En effet, ils ont utilisé l'algorithme du gradient SD (Steepest Descent) avec quelques modifications. En utilisant ces méthodes améliorées, appelées les algorithmes de Legendre modifiés, les auteurs ont pu obtenir un FM de 6,4245 pour les longues séquences binaires. Il existe de nombreuses méthodes utilisées pour générer de longues séquences binaires autres que celles basées sur les séquences de Legendre à savoir les séquences de Jacobi et les séquences de Jacobi modifiées. Ces deux dernières sont construites en combinant deux séquences de Legendre appropriées, et elles ont également de bonnes propriétés de corrélation et une grande complexité linéaire. Dans [112], DH. Green, dans sa recherche, a utilisé des séquences de Jacobi modifiées pour construire de longues séquences binaires avec des valeurs élevées du FM. Dans [113], B. Suribabu Naick et al ont utilisé les mêmes méthodes de construction en combinaison avec différents algorithmes de recherche pour obtenir de meilleurs résultats.

Dans [114], Ding et al ont appliqué l'extension cyclotomique, qui est un vieux sujet de la théorie algébrique des nombres, pour construire et étudier une famille de séquences PN, appelées séquences cyclotomiques, constituée de séquences de Legendre, de Jacobi et de Jacobi modifiées.

Un autre type de séquences qui présente de très bonnes performances en termes de la FAC et qui fournit les valeurs optimales du FM sont les séquences de Barker ; les résultats, concernant ces séquences, ont été prouvés par les auteurs dans les références [115-116].

## 2.4 Méthodes de génération de longues séquences binaires

Dans cette section, nous présentons les plus importants travaux publiés sur les méthodes de génération de longues séquences binaires basées sur l'utilisation de sous-séquences binaires plus courtes en se basant sur différentes méthodes d'optimisation.

### 2.4.1 Conception de séquences longues avec une faible FAC par des séquences binaires entrelacées

Dans [117], Meng et Yan ont proposé deux nouvelles constructions de séquences binaires de période 4N à faible FAC basées sur la technologie d'entrelacement.

#### a. Construction A

Dans ce cas, la séquence "$a$" est composée des sous-séquences binaires $\mathrm{s} = (\mathrm{s}(0), \mathrm{s}(1), \dots, \mathrm{s}(\mathrm{N}-1))$ qui sont entrelacées. Elle est définit par [117]:

$$a = I(s_1, L^d(\overline{s_1}), s_2, L^d(\overline{s_2})) \qquad 2\text{-}10$$

Où :

- $I$ désigne l'opérateur d'entrelacement.
- $s_1$ et $s_2$ sont deux séquences binaires de longueur N chacune, $s_1 = (\mathrm{s}(0), \mathrm{s}(2), \mathrm{s}(2\mathrm{t}) \dots)$ et $s_2 = (\mathrm{s}(1), \mathrm{s}(3), \mathrm{s}(2\mathrm{t}+1) \dots)$ où $t = 0,1,2, \dots, N-1$ et $\mathrm{N} \equiv 3 \pmod 4$ et elles ont une FAC idéale [118].
- $L$ est l'opérateur de décalage cyclique à gauche.
- $d$ est un entier tel que : $d \neq \frac{N+1}{4}$.
- $\overline{s_1}$ est la séquence complémentaire de $s_1$.
- $\overline{s_2}$ est la séquence complémentaire de $s_2$.

Sachant que la séquence "$a$" résultante est équilibrée, c'est-à-dire qu'elle possède le même nombre de symboles 1 et de 0 dans une période égale à 4N, la FAC de la nouvelle séquence est définie par [117]:

- Si : $d \neq \frac{N+1}{4}$

$$R_a(\tau) = \begin{cases} 4\,N & 2\,fois \\ 2-2N & 4\,fois \\ 4 & 2N-4\,fois \\ -1 & 2N-2\,fois \end{cases} \quad 2\text{-}11$$

- Si : $d = \frac{N+1}{4}$

$$R_a(\tau) = \begin{cases} 4\,N & 2\,fois \\ -4N & 2\,fois \\ 4 & 2N-4\,fois \\ -4 & 2N-2\,fois \end{cases} \quad 2\text{-}12$$

**b. Construction B**

Dans cette construction, les auteurs ont utilisé les mêmes sous-séquences mais d'ordres différents. La nouvelle séquence entrelacée, de période $4N$, est définie comme suit:

$$a = I(s_1, L^d(\bar{s_1}), \bar{s_2}, L^d(s_2)) \quad 2\text{-}13$$

La FAC de la nouvelle séquence est définie par [117]:

- Si : $d \neq \frac{N+1}{4}$

$$R_a(\tau) = \begin{cases} 4\,N & 1\,fois \\ -4N & 1\,fois \\ -4 & N-1\,fois \\ 4 & N-1\,fois \\ 0 & 2N-4\,fois \\ -2-2N & 2\,fois \\ 2+2N & 2\,fois \end{cases} \quad 2\text{-}14$$

- Si : $d = \frac{N+1}{4}$

$$R_a(\tau) = \begin{cases} 4\,N & 1\,fois \\ -4N & 1\,fois \\ 4 & N-1\,fois \\ -4 & N-1\,fois \\ 0 & 2\,N\,fois \end{cases} \quad 2\text{-}15$$

Les valeurs de la FAC déphasée de la séquence "$a$" sont toutes contenues dans l'ensemble $\{0, -4, 4\}$ (Sauf pour $-4N$). Par conséquent, la séquence "$a$" est une séquence binaire avec une amplitude de la FAC presque optimale.

- **Exemple 1**

Soit $N = 7$, $d = \frac{N+1}{4}$ et $s = [1\ 1\ 1\ 0\ 0\ 1\ 0]$, une m-séquence de période 7. La nouvelle séquence "$a$" de période $4N = 28$, définie par la construction A, est donnée par :

$$a = [1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1]:$$

Après tout calcul fait, la FAC de "$a$" est donnée par :

$$R_a(\tau) = \{28\ 4-4\ 4-4\ 4-4-28-4\ 4-4\ 4-4\ 4\ 28\ 4-4\ 4-4\ 4 \\ -4\ 28-4\ 4-4\ 4-4\ 4\}$$

- **Exemple 2**

Soit N $= 7$, d $= \frac{N+1}{4}$ et s $= [1\ 1\ 1\ 0\ 0\ 1\ 0]$, une m-séquence de période 7. La nouvelle séquence "a" de période 4N $= 28$, définie par la construction B, est donnée par:

$$a = (1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0)$$

Après tout calcul fait, la FAC de "a" est donnée par :

$$R_a(\tau) = \{28\ 0\ 4\ 0\ -4\ 0\ 4\ 0\ -4\ 0\ 4\ 0\ -4\ 0\ 28\ 0-4\ 0\ 4\ 0 \\ -4\ 0\ 4\ 0-4\ 0\ 4\ 0\}$$

Les résultats montrent que les séquences obtenues présentent une faible FAC dans certaines conditions liées à la valeur de "d" avec moins de complexité linéaire encore à résoudre.

### 2.4.2 Conception de séquences PN de longue période par l'addition des M –séquences

Dans la référence [119], Jian Ren a proposé une méthode de construction de séquences de longue période appelées AMPLS ayant une grande complexité et des propriétés de corrélation faible à partir de l'addition modulo 2 des m-séquences. Le principe et les résultats obtenus de cette méthode seront donnés dans ce qui suit.

**a. Méthode de génération de ces séquences**

Les AMPLS sont construites de la manière suivante :

soit "$a_i$" une m-séquence générée avec le polynôme primitif $f_i(x)$ de degré $n_i$, $i = 1, 2, \cdots, s$ ou "$s$" est le nombre des m-séquence utilisées. Si $n_1, n_2, \cdots, n_s$ sont des paires

premiers, alors la séquence AMPLS: $a = a_1 + a_2 + \cdots + a_s$ est l'addition modulo 2 des m-séquences $a_1, a_2, \cdots, a_s$.

La période de la séquence AMPLS générée est égale à $(2^{n_1} - 1)(2^{n_2} - 1) \cdots (2^{n_s} - 1)$ [119].

- **Exemple**

Soit $a = [0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0 \cdots]$, une m-séquence de période 15 générée avec le polynôme primitif f(x) $= x^4 + x + 1$ et $b = [1011100 \cdots]$ une autre m-séquence de période 7 générée avec le polynôme primitif $g(x) = x^3 + x + 1$. Soit $c = a + b \bmod 2$, alors : c = [1 1 1 1 0 1 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 0 0 0 1 0 1 1 0 0 1 1 0 0 0 0 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 · · · ] est une séquence d'AMPLS de période 7 × 15 = 105 et de polynôme générateur $f(x)g(x) = (x^4 + x + 1)(x^3 + x + 1)$.

b. **Autocorrélation des séquences d'AMPLS**

Soit " $a_i$" une m-séquence de période $N_i$, où $i = 1, 2, \cdots, s$ et soit $a = \sum_{i=1}^{s} a_i \bmod 2$, où $a_i$ est une séquence non nulle, alors [119] :

1) La période moyenne $N$ de "$a$" est égale à $lcm\ (N_1, N_2, \cdots, N_s)$ ;
2) Si $gcd\ (N_1, N_2, \cdots, N_s) = 1$, alors la FAC est définie par :

$$R_a(\tau) = R_{a_1}(\tau) R_{a_2}(\tau) \ldots R_{a_s}(\tau) \qquad 2\text{-}16$$

La FAC des séquences AMPLS modulo p (avec p un nombre premier impair) est déterminée comme suit [119]:

Soit $a = \sum_{i=1}^{s} a_i \bmod 2$. Puisque la période de chaque $a_i$ est égale à $p^n - 1$ pour un nombre entier pair n, on peut décomposer $a_i$ en deux séquences $b_i$ et $c_i$ de telle sorte que $a_i$ est une séquence résultant de l'entrelacement de $b_i$ et de $c_i$, c'est-à-dire: $a_i = (b_{i,1}, c_{i,1}, b_{i,2}, c_{i,2}, \ldots, b_{i,N_I}, c_{i,N_I})$ où $i = 1, 2, \cdots, s$. Supposons que $gcd\ (N_1, N_2, \cdots, N_s) = 1$, alors :

1) La période moyenne $N$ de "$a$" est égale à $2 \times lcm\ (N_1, N_2, \cdots, N_s)$ ;
2) La FAC est définie par :

$$R_a(\tau) = \begin{cases} \prod_{i=1}^{s} R_{c_i,b_i}\left(\frac{\tau-1}{2}\right) + \prod_{i=1}^{s} R_{c_i b_i}\left(\frac{\tau+1}{2}\right) & si\ \tau\ est\ impair \\ \prod_{i=1}^{s} R_{b_i}\left(\frac{\tau}{2}\right) + \prod_{i=1}^{s} R_{c_i}\left(\frac{\tau}{2}\right) & si\ \tau\ est\ pair \end{cases} \quad 2\text{-}17$$

### 2.4.3 Construction de longues séquences avec de bonne propriétés de corrélations par optimisation « OEP »

Dans la référence [120], Sarayloo M. et al ont proposé une méthode de construction de longues séquences binaires, présentant les propriétés souhaitées de la FAC et la FIC, basée sur l'utilisation d'un certain nombre de courtes DBSs non linéaires. La construction de ces séquences est réalisée à l'aide de la technique d'optimisation « OEP ».

#### D. Méthode de construction des séquences et analyse de la FAC

La nouvelle séquence est, au fait, composée de deux ou plusieurs séquences différentes de même longueur, c'est-à-dire appartenant à la même famille des séquences DBSs. Les méthodes de génération des DBSs sont données dans la section 1.4.12 du chapitre 1. Il y a deux raisons importantes qui motivent l'utilisation des DBSs dans la méthode de construction des séquences décrite ci-dessous. Premièrement, la génération des DBSs de grandes tailles est très difficile, de sorte qu'elle reste toujours un problème ouvert. Deuxièmement, le $\max(|FAC(k \neq 0)|)$ et le $max|FIC|$ des DBSs augmentent proportionnellement en fonction du nombre d'étages du registre LFSR [120]. Dans ce qui suit, un exemple de construction de ces séquences sera abordé.

Supposant le cas simple de deux séquences différentes notées par :

$\mathrm{a} = \{\mathrm{a}_1, \mathrm{a}_2, \dots, \mathrm{a}_\mathrm{N}\}$ et $\mathrm{b} = \{\mathrm{b}_1, \mathrm{b}_2, \dots, \mathrm{b}_\mathrm{N}\}$, où $N$ est la longueur de la séquence, la nouvelle séquence sera présentée par : $\mathrm{S} = \{\mathrm{a}_1\mathrm{b}_1, \mathrm{a}_2\mathrm{b}_2, \mathrm{a}_3\mathrm{b}_3, \dots, \mathrm{a}_\mathrm{N}\mathrm{b}_\mathrm{N}\}$

La FAC et la FIC périodiques de ces séquences, sont définis par l'Equation 2.18.

$$C_{ab}[k] = \frac{1}{N}\sum_{i=0}^{N-1} a(i)b(i+k) \quad 2.18$$

Si $a = b$, la FAC est mesurée, sinon c'est la FIC qui est calculée.

Supposons, pour des raisons de simplicité et sans perte de généralité, que la longueur de chaque séquence ($a$ et $b$) est 5 alors, la longueur de la séquence mixte est de 10 et sa FAC est périodique. Par conséquent, l'analyse peut être limitée seulement aux cinq premiers pics latéraux, dont les valeurs sont indiquées dans le Tableau 2-3 [120]:

**Tableau 2-3 Valeurs de la FAC de la séquence mixte $S = \{\mathbf{a_1 b_1}, \mathbf{a_2 b_2}, \dots, \mathbf{a_5 b_5}\}$, en fonction des valeurs de FAC / FIC des sous séquences a et b pour différentes valeurs du décalage.**

| Décalage | Formule |
|---|---|
| **0** | $\mathrm{FAC_a}(0) + \mathrm{FAC_b}(0)$ |
| **1** | $\mathrm{FIC_{ab}}(0; \mathrm{b^R}) + \mathrm{FIC_{ab}}(1; \mathrm{b^R})$ |
| **2** | $\mathrm{FAC_a}(1^{\mathrm{R}}) + \mathrm{FAC_b}(1^{\mathrm{R}})$ |
| **3** | $\mathrm{FIC_{ab}}(4; \mathrm{b^R}) + \mathrm{FIC_{ab}}(2; \mathrm{b^R})$ |
| **4** | $\mathrm{FAC_a}(2^{\mathrm{R}}) + \mathrm{FAC_b}(2^{\mathrm{R}})$ |
| **5** | $\mathrm{FIC_{ab}}(3; \mathrm{b^R}) + \mathrm{FIC_{ab}}(3; \mathrm{b^R})$ |

La FAC d'une séquence mixte S peut être formulée comme suit [120] :

$$AC(k) = \begin{cases} FAC_a\left(\frac{k}{2}\right) + FAC_b\left(\frac{k}{2}\right) & \text{si k: le même} \\ FIC_{ab}\left(2N - \left\lfloor\frac{k}{2}\right\rfloor, b^R\right) + FIC_{ab}\left(\left\lfloor\frac{k}{2}\right\rfloor, b^R\right) & \text{si k: impair} \end{cases} \quad 2\text{-}19$$

Ou $\lfloor . \rfloor$ et $\lceil . \rceil$sont les valeurs d'arrondi au plus grand entier inférieur et au plus grand entier supérieur, respectivement.

On peut noter, d'après ce résultat, ce qui suit :

- Lorsque le décalage $k$ est impair, la FAC de la séquence $S$ dépend des deux termes calculés de la FIC des sous-séquences ;
- Sinon, la FAC de la séquence $S$ est la somme des FACs des sous-séquences.

L'équation suivante peut être développée pour le cas général d'une séquence composée de $M$ sous-séquences différentes pour un décalage $k$ [120]:

$$FAC_s(k) = \begin{cases} \sum_{i-1}^{M} FIC_{a_i a_{\lambda(j,k)}}\left(Shf_{in}, a_{\lambda(j,k)}^R\right) & si\ mod(k, M) \neq 0 \\ \sum_{i-1}^{M} FAC_{a_i}\left(\frac{k}{M}\right) & si\ mod(k, M) = 0 \end{cases} \quad 2\text{-}20$$

Où :

- $l_s$ est la longueur de chaque sous séquence ;

- $Shf_{in}$ est l'indice de décalage qui devrait être appliqué à une sous-séquence $a_\lambda$
- $M$ est le nombre de sous-séquences ;
- $M \times l_s$ est la longueur de la séquence résultante $S$ ;
- les sous-séquences sont illustrées par $a_i$, où i est l'indice de la sous-séquence (allant de 1 à $M$). Les bits de la séquence sont présentés par $a_{i,k}$ , où $i = 1 \dots M$ et $q = 1 \dots l_s$.

Dans la séquence *S*, toutes les sous-séquences sont répétées régulièrement, tel que $S = \{a_{11}, a_{21}, \dots, a_{M1}, \dots, a_{1l_s}, a_{2l_s}, \dots, a_{Ml_s}\}$, dans le cas où la séquence $S$ est composée de $M$ sous-séquences $a_1$, $a_2$ et $a_M$, de longueur $l_s$.

Dans le but d'arriver à une séquence finale $S$ avec une FAC quasi-optimale, nous devons trouver [120]:

1) les valeurs de $i$ et $\lambda$ pour que les sous-séquences $a_i$ et $a_\lambda$ soient corrélées entre-elles ;
2) combien de fois $(k)$ la sous-séquence correspondante doit-être décalée.

Le Tableau 2-4 montre les contributions du premier $M-TUPLE$ de la séquence $S$ : le premier argument de chaque parenthèse identifie le numéro de la sous-séquence, et le deuxième argument représente la position du bit dans cette sous-séquence.

**Tableau 2-4 Souscriptions de l'élément de la séquence $S$ pour le $M-Tuple$.**

| | $(j,q)$ | | | | |
|---|---|---|---|---|---|
| **K** | **(1,1)** | **(2,1)** | **(3,1)** | **…** | **(M , $l_s$)** |
| **1** | (M , $l_s$) | (1,1) | (2,1) | … | (M-1,1) |
| **2** | (M-1 , $l_s$) | (M , $l_s$) | (1 ,1) | … | (M-2 , 1) |
| ⋮ | ⋮ | ⋮ | ⋮ | | ⋮ |
| **$M$** | (1 , $l_s$) | (2 , $l_s$) | (3 , $l_s$) | … | (M , $l_s$) |
| **$M+1$** | (M , $l_s-1$) | (1 , $l_s$) | (2 , $l_s$) | … | (M-1 , $l_s$) |
| **$M+2$** | (M-1 , $l_s-1$) | (M , $l_s-1$) | (1 , $l_s$) | … | (M-2 , $l_s$) |
| ⋮ | ⋮ | ⋮ | ⋮ | | ⋮ |
| **$2M$** | (1 , $l_s-1$) | (2 , $l_s-1$) | (3 , $l_s-1$) | ... | (M , $l_s-1$) |
| **$2M+1$** | (M , $l_s-2$) | (1 , $l_s-1$) | (2 , $l_s-1$) | ... | (M-1 , $l_s-1$) |
| **$2M+2$** | (M-1 , $l_s-2$) | (M , $l_s-2$) | (1 , $l_s-1$) | ... | (M-2 , $l_s-1$) |
| ⋮ | ⋮ | ⋮ | ⋮ | | ⋮ |
| **$M.l_s$** | (1,1) | (2,1) | (3,1) | … | (M, 1) |

Puisque les sous-séquences sont répétées périodiquement (comme mentionné ci-dessus), tous les numéros de sous-séquences peuvent être trouvés à n'importe quel $M-TUPLE$ dans le premier argument. Également, le premier argument entre parenthèses dans la première colonne se répète périodiquement $[M, M-1, \dots, 2, 1]$. En réalité, nous devons

placer l'indice de décalage k sur la matrice mentionnée. Par conséquent, l'Equation 2-21 peut être utilisée pour la planification, où $j\ et\ \lambda$ sont les numéros de sous-séquences de la séquence non décalée (représentée par la couleur bleue dans le Tableau 2-5) et la séquence décalée (représentée par la couleur rouge dans le Tableau 2-4), respectivement. En plus, $Seq\_Ind\ =\ [1, 2\ , \dots\ , M\ -\ 1, M, 1, 2, \dots\ , M\ -\ 1, M]$ [120]. $\lambda$ est donné par :

$$\lambda = Seq\_Ind\ (M - (k-1) + (j-1))\ \ , j = 1 \dots M \qquad 2\text{-}21$$

Il est important de noter que l'Equation 2-21 ne fonctionne pas pour $k > M$, car $k$ augmente continuellement de 1 à $M.l_s$ et $[M, M\ -\ 1, \dots, 2, 1]$ sont répété périodiquement dans le premier argument de la première colonne. Ainsi, le facteur $k$ devrait être modifié dans l'Equation 2-21. L'Equation 2-22 peut être utilisée pour résoudre ce problème. En fait, le paramètre $k$ de l'Equation 2-21 est remplacé par $x$ dans l'Equation 2-22.

$$x\ =\ T(mod(k, M) + 1), \qquad T = [M, 1,2, \dots, M-1] \qquad 2\text{-}22$$

Le même raisonnement est utilisé pour formuler l'indice de décalage mentionné dans l'Equation 2-22. Le Tableau suivant montre la quantité de décalage pour chaque sous-séquence à chaque rotation. Le Tableau 2-5 est dérivé du Tableau 2-4. Ce tableau peut être formulé selon l'Equation 2-23, où x est obtenu à partir de l'Equation 2-22.

$$Shf_{in}(k) = \begin{cases} \left\lfloor \frac{k}{M} + 1 \right\rfloor & \text{si } x \le i \\ \left\lfloor \frac{k}{M} \right\rfloor & \text{si } x < i \end{cases} \qquad 2\text{-}23$$

**Tableau 2-5 Valeur de l'indice de décalage à chaque rotation.**

| $k$ | 1 | 2 | 3 | … | $M$ |
|---|---|---|---|---|---|
| **1** | 1 | 0 | 0 | … | 0 |
| **2** | 1 | 1 | 0 | … | 0 |
| ⋮ | ⋮ | ⋮ | ⋮ | | ⋮ |
| $M$ | 1 | 1 | 1 | … | 1 |
| $M+1$ | 2 | 1 | 1 | … | 1 |
| $M+2$ | 2 | 2 | 1 | … | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | | ⋮ |
| $2M$ | 2 | 2 | 2 | … | 2 |
| $2M+1$ | 3 | 2 | 2 | … | 2 |
| $2M+2$ | 3 | 3 | 2 | … | 2 |
| ⋮ | ⋮ | ⋮ | ⋮ | | ⋮ |
| $M.l_s$ | 0 | 0 | 0 | … | 0 |

### E. Technique OEP appliquée au processus de génération proposé

Les plus longues séquences générées à partir des séquences DBSs sont nommées NSDBSs (New Shuffled DBSs). La FAC des NSDBS dépend des FIC et des FACs de chaque sous-séquence. Pour obtenir une FAC appropriée de la nouvelle séquence plus longue, la meilleure position de bit de départ SBP (Starting Bit Position) de chaque sous-séquence impliquée dans les NSDBS doit être trouvée. La technique OEP est utilisée pour trouver les SBP des sous-séquences. La fonction d'erreur quadratique moyenne RMSE (Root Mean Square Error) est adoptée pour calculer la fonction de fitness [120].

Les résultats obtenus ont montré que le mélange de séquences de De Bruijn, selon l'approche susmentionnée, offre des performances remarquables en termes de suppression des valeurs des pics secondaire de la FAC de la séquence NSDBS générée.

## 2.5 Conclusion

Dans la première partie de ce chapitre on a essayé de présenter brièvement les différentes méthodes d'optimisations les plus utilisées dans les différents domaines technologiques. Nous avons mis l'accent sur celles que nous avons étudiées dans le cadre de cette thèse de Doctorat. Il s'agit des AGs qui sont des méthodes relativement anciennes et qui ont été beaucoup utilisées pour résoudre des problèmes difficiles d'optimisations et de recherche. De plus, un autre outil évolutionnaire d'optimisation très récent et très important dans l'engineering qui est l'Intelligence en Essaim, tout particulièrement, l'OEP, a été abordé. Notre présentation n'est absolument pas exhaustive, mais elle montre les grandes lignes de ces méthodes. La qualité des solutions trouvées par ces méthodes dépend de leur paramétrage, et de l'équilibre entre un balayage de tout l'espace des solutions (diversification) et une exploration locale (l'intensification).

Nous avons présenté dans la deuxième partie de ce chapitre un état de l'art sur les principaux travaux de recherche réalisés dans le domaine de recherche et de conception de séquences présentant de bonnes propriétés de corrélation.

Nous avons montré que les algorithmes, récemment proposés et largement adoptés, notamment le CAN et le PeCAN sont des méthodes efficaces en termes de calcul, du fait qu'elles utilisent des opérations à base d'algorithme FFT pour concevoir des séquences avec de bonnes propriétés de la FAC.

Le chapitre suivant sera consacré à la présentation de notre contribution majeure réalisée dans le cadre de notre thèse de Doctorat. Il s'agit de la conception et de l'optimisation de séquences PN longues destinées pour les applications à accès multiple plus particulièrement pour les applications GNSS.

# Chapitre 3 : Génération & Optimisation des séquences PN longues

Chapitre 3

# Génération & Optimisation des séquences PN longues

## 3.1 Introduction

Dans ce chapitre, nous présentons le principe et les résultats de l'implémentation de notre méthode, proposée comme contribution principale, pour générer deux types d'OLBSS avec de bonnes propriétés de la FAC. Rappelons ici que l'optimisation des OLBSS, obtenues par l'application de cette méthode, est réalisée à l'aide de deux techniques différentes, à savoir les AGs et la méthode OEP.

La présentation du principe de mise en œuvre de notre méthode est dévoilée suivant quatre sous-sections. En effet, dans la première sous-section, nous allons présenter une description détaillée de la méthode de génération des deux types d'OLBSS. La deuxième sous-section est consacrée à l'explication des deux méthodes utilisées, au cours de ce travail de recherche, pour l'optimisation des caractéristiques de la FAC des longues séquences OLBSS, en l'occurrence les AGs et la méthode OEP. Ici, la structure et les différents opérateurs qui composent ces deux méthodes sont montrées d'une façon détaillée. Par la suite, nous donnons la preuve mathématiquement du raisonnement sur lequel la méthode proposée est fondée. La dernière sous-section est une étude des performances des longues séquences OBLSS obtenues par l'application de notre approche. Cette dernière sous-section contient principalement les résultats et les discussions des cinq scénarios de test.

## 3.2 Principe de construction des longues séquences binaires proposées

Comme nous l'avons déjà donné dans le résumé et l'introduction générale, le premier type d'OLBSS est construit à partir de sous-séquences binaires courtes concaténées appartenant à la même famille de codes, telles que les sous-séquences de Walsh-Hadamard et de Gold, à condition que leurs FICs aient de bonnes propriétés. La deuxième catégorie utilise les mêmes sous-séquences mais qui sont plutôt entrelacées. Par conséquent, selon la façon dont les sous-séquences sont utilisées (concaténation ou entrelacement), on peut construire deux types de longues séquences binaires S à partir de M sous-séquences plus courtes différentes $A_i (i = 1, \dots M)$ de même longueur $l_s$ appartenant à la même famille. Dans ce qui suit, le principe de chaque configuration sera montré en détail.

### 3.2.1 Configuration des OLBSS basée sur la concaténation

Dans cette configuration, la séquence longue S est composée de M sous-séquences $A_i$ $(i = 1, \dots M)$ qui sont concaténées horizontalement. Dans ce cas, nous utilisons la méthode OEP ou les AGs pour trouver les positions optimales des sous-séquences dans la séquence longue proposée. Ces dernières peuvent être représentées par le vecteur d'affectation des positions $P = [P_1, \dots, P_M]$. Où $P_j$ $(j = 1, \dots M)$ est un entier tel que $1 \leq P_j \leq M$; cela signifie que la sous-séquence $A_{P_j}$ est affectée à la $j^{ième}$ position dans la longue séquence S. Par exemple, pour $M = 3$, si l'algorithme métaheuristique donne $P = [3, 1, 2]$, alors, les sous-séquences concaténées $A_3$, $A_1$et $A_2$sont affectées, réspéctivement, à la première, deuxième et troisième position de la longue séquence. Par conséquent, la séquence résultante est construite comme le montre la Figure 3-1.

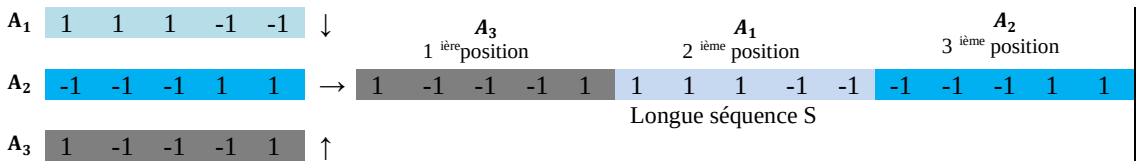



**Figure 3-1 Principe de génération du premier type de longue séquence pour $M = 3$ avec $P = [3, 1, 2]$**

### 3.2.2 Configuration des OLBSS basée sur l'entrelacement

Dans cette configuration, la longue séquence proposée S est construite à partir de la concaténation de $n$ ensembles de $M$ bits chacun. Ici, le $k^{ième}$ ensemble ($k = 1, \dots, n$) est obtenu en entrelaçant les $k^{ième}$ bits des $M$ sous-séquences. Les positions optimales de chaque bit $B_i$ ($i = 1, \dots M$) dans chaque ensemble sont trouvées en utilisant la méthode OEP ou les AGs similairement au cas de la concaténation. Par exemple, pour $M = 3$, si l'algorithme métaheuristique donne $P = [3, 1, 2]$, alors la séquence $S$ est construite comme le montre la Figure 3-2.

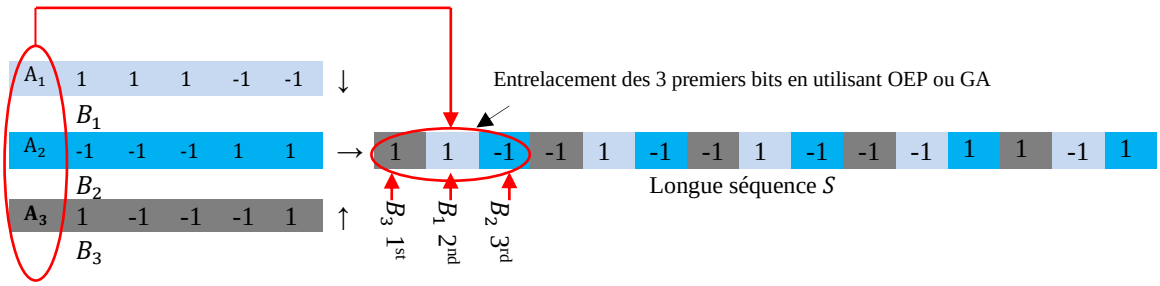



**Figure 3-2 Principe de génération du deuxième type de longue séquence pour $M = 3$ avec $P = [3, 1, 2]$.**

Le paramètre clé dans la génération des séquences OLBSS longues proposées est l'ensemble des sous-séquences binaires utilisées. En effet, ces dernières doivent appartenir à la même famille de codes. Plusieurs séquences courtes peuvent être utilisées. On peut citer comme exemples les sous-séquences de Walsh-Hadamard et celles de Gold. Le choix d'un tel ensemble des sous-séquences est basé sur leurs caractéristiques des fonctions FICs. Dans ce qui suit, on présente les caractéristiques, en termes de FAC, de deux types de sous-séquences qui peuvent être utilisées dans la génération des séquences OLBSS longues.

## 3.3 Caractéristiques de la FAC des séquences de Walsh-Hadamard et Gold

La FAC d'une séquence de Walsh-Hadamard de longueur 64 bits est montrée dans la Figure 3-3.

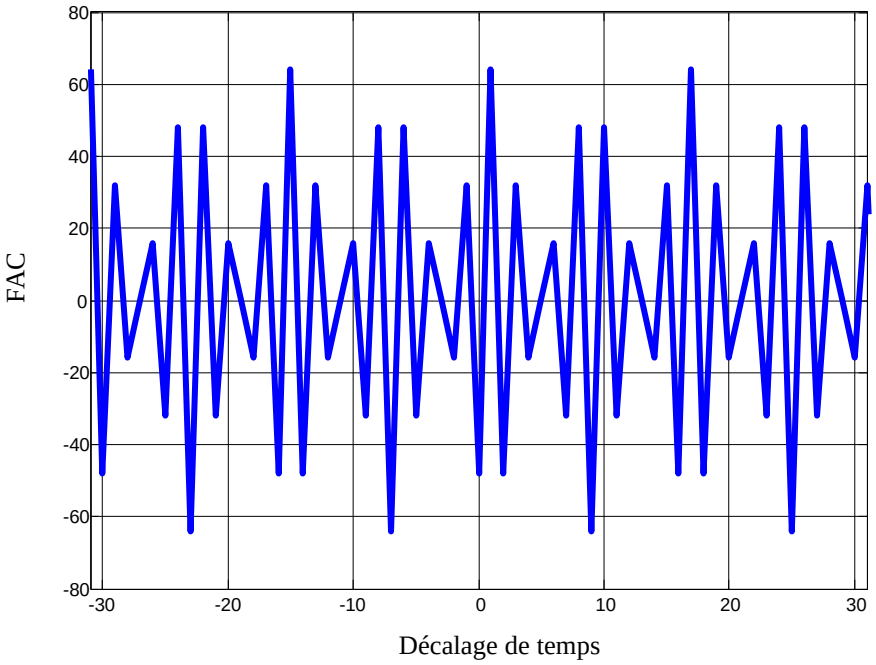



**Figure 3-3 FAC de la séquence de Walsh-Hadamard de longueur 64.**

On observe clairement, à partir de la Figure 3-3, que les amplitudes de tous les pics secondaires de la FAC de la séquence de Walsh-Hadamard sont très élevés. Malgré que les séquences de Walsh-Hadamard aient des FACs qui ne sont pas intéressantes, elles sont orthogonales et présentent par conséquent des fonctions FICs nulles. Les Figures 3-4 et 3-5 montrent les valeurs de la FAC pour les séquences de Gold et celles de Weil.

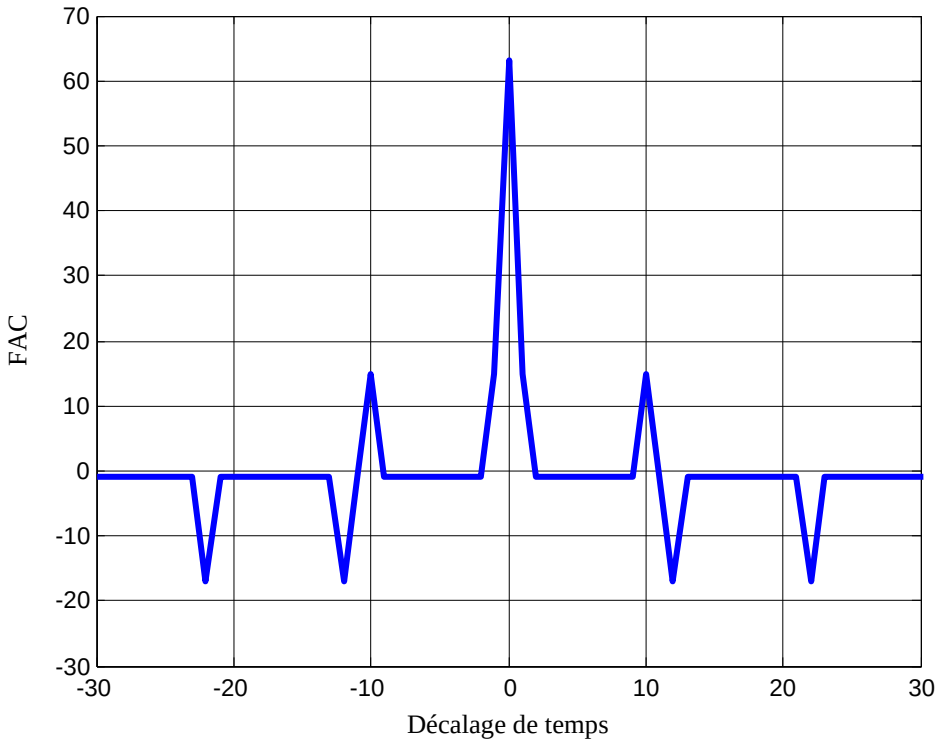



**Figure 3-4 Caractéristiques de la FAC de la séquence de Gold de longueur 63.**

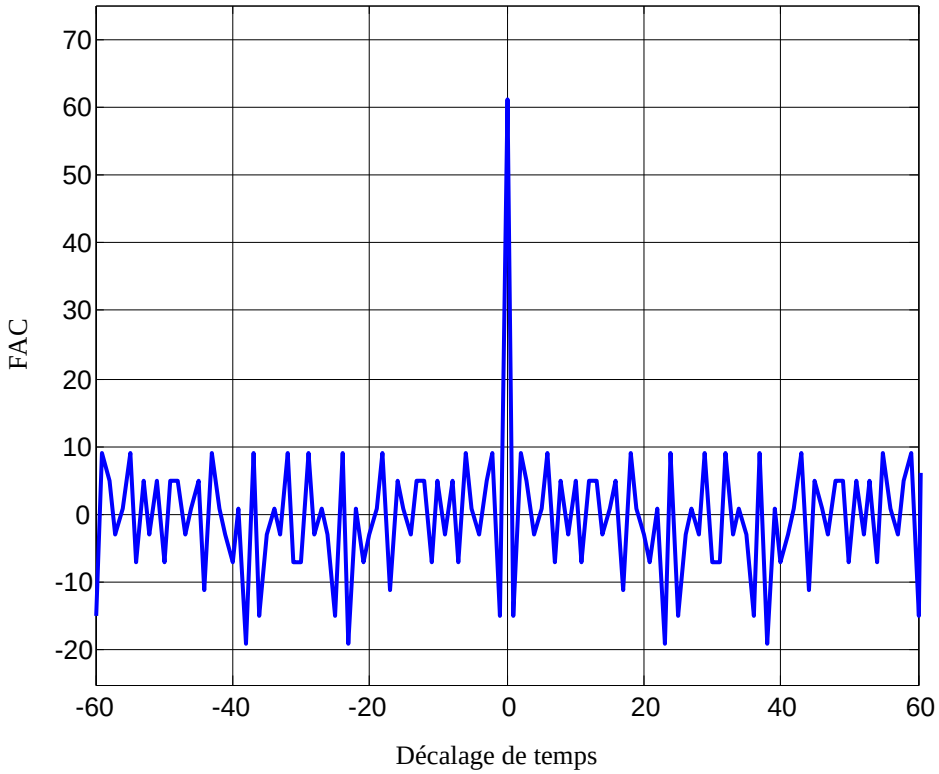



**Figure 3-5 Caractéristiques de la FAC de la séquence de Weil de longueur 61.**

Une comparaison directe des FACs des séquences de Walsh-Hadamard de longueur 64 et des séquences de Gold de longueur 63 indique clairement que ces dernières possèdent des caractéristiques importantes en termes de FAC. Par conséquent, nous pouvons dire que les séquences de Gold sont meilleures en comparaison avec les séquences Walsh-Hadamard en termes des propriétés de la FAC. En revanches, les séquences de Walsh-Hadamard sont nettement supérieures par rapport aux séquences de Gold vis-à-vis de la caractéristique de la FIC.

Les séquences de Weil ont des pics secondaires de la FAC qui sont plus faibles par rapport à ce que l'on observe pour les deux autres codes.

## 3.4 Optimisation des longues séquences par les AGs

Comme nous l'avons vu, les AGs reposent sur les deux principes du néodarwinisme. Ils empruntent d'ailleurs leur terminologie au modèle évolutionniste.

La Figure 3-6 présente le principe général des AGs [123].

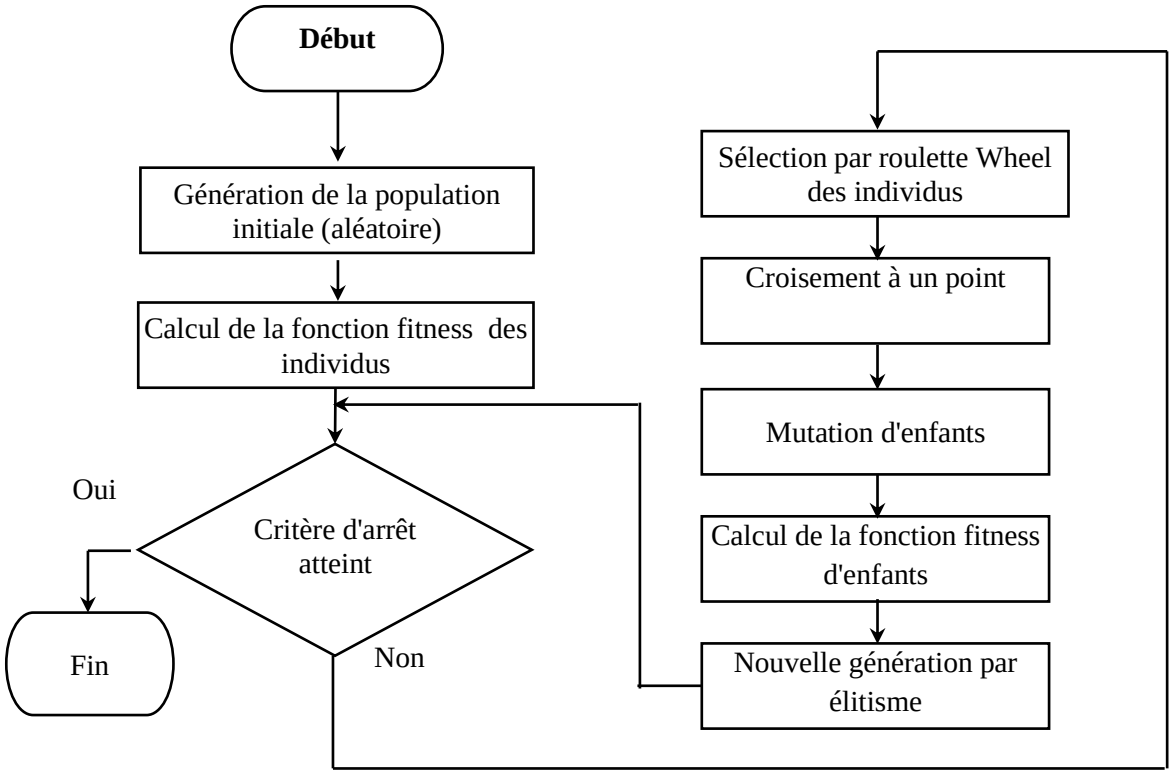



**Figure 3-6 Schéma général des AGs utilisés.**

Dans ce qui suit, les étapes de calcul par les AGs seront détaillées.

### 3.4.1 Représentation des chromosomes

Le chromosome est représenté par le vecteur $P$ de $M$ variables. Il représente respectivement les valeurs des $M$ positions des sous-séquences ou des $M$ positions des bits $P_i$, selon le type d'OLBSS (concaténées ou entrelacées). En général, il peut être représenté par [123]:

$$Chromosome = P = [P_1, P_2, P_3, \dots, P_M] \quad 3\text{-}1$$

### 3.4.2 Création de la population initiale

La résolution d'un problème par les AGs débute avec la génération de la population initiale. Le choix de la population initiale est important pour le déroulement du programme. En effet, le choix de cette population peut rendre plus ou moins rapide la convergence vers l'optimum global.

Les AGs génèrent au préalable, d'une façon aléatoire, la population initiale $Npop$ dans l'espace de recherche. La population initiale est une matrice de nombres aléatoires uniformément distribués sur un intervalle bien déterminé. Ils sont donnés comme suit :

$$InitPop = round(rand(N_{pop}, M)) \quad 3\text{-}2$$

Où :

- $M$ est le nombre de sous-séquences
- La valeur de $Npop$ est généralement égale à $40$.

### 3.4.3 Fonction fitness

La qualité d'une solution représentée par un chromosome est donnée par la fonction de fitness (appelée aussi fonction d'adaptation, d'évaluation, objective, de coût …) représentant son potentiel. Elle est utilisée par les AGs pour trouver le meilleur individu. Elle associe donc une valeur à chaque individu de la population, permettant ainsi aux solutions de pouvoir être comparées entre elles.

#### a. Le choix de la fonction fitness

La mesure de la qualité d'une séquence $S$ de longueur $N$ se base généralement sur l'étude de son FAC périodique qui est définie par:

$$R_S(u) = \sum_{i=1}^{N} S(i)\, S(i-u) \quad 3\text{-}3$$

La dégradation des propriétés de la FAC a une relation directe sur le spectre de chaque séquence.

Afin de déterminer quantitativement l'importance de cette dégradation pour un ensemble de séquences donné, on utilise le $FM$ qui représente une mesure spectrale nécessaire pour évaluer les caractéristiques fréquentielles des séquences. Le $FM$, pour la séquence binaire $S$ de longueur $N$, a été défini dans la section 1.5.2 du chapitre 1.

Les séquences à faible $FM$ ont un spectre étroit et ne conviennent pas aux applications CDMA. Donc, la meilleure séquence binaire est celle pour laquelle ce facteur est le plus élevé possible ; puisque son spectre est plat, il est donc le plus proche aux caractéristiques du bruit.

Les Figures 3-7 et 3-8 représentent, en détail la relation entre le spectre et le $FM$.

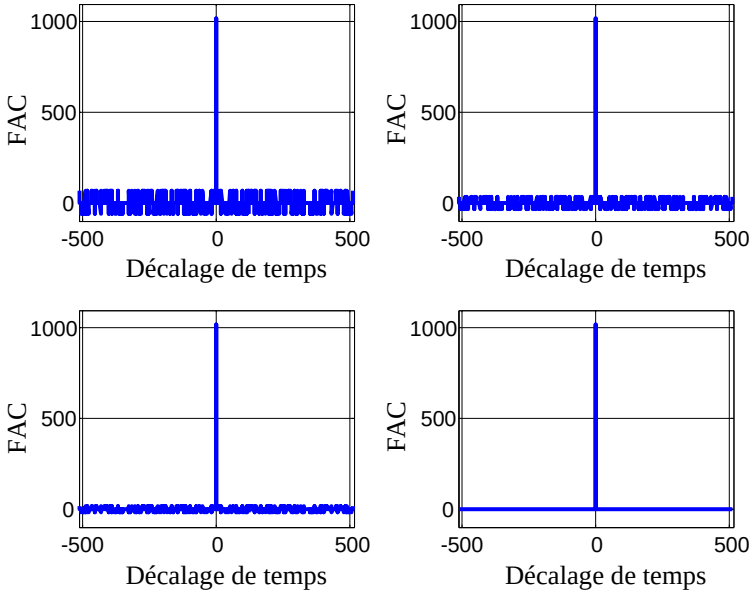



**Figure 3-7 Augmentation du FM.**

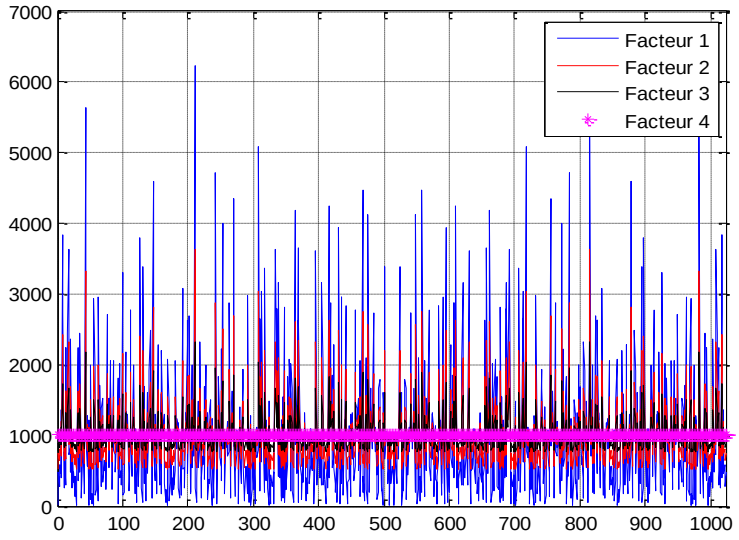



**Figure 3-8 Effet du FM sur la représentation spectrale.**

La réduction du niveau des pics secondaires entraîne une augmentation du $FM$. C'est pour cette raison que notre premier objectif, dans ce travail, étant la minimisation de la fonction $F$, qui est définie comme étant l'inverse du $FM$. La fonction fitness est donc la suivante:

$$F = \frac{1}{FM(S)} \qquad 3\text{-}4$$

**b. L'évaluation**

Deçà, la fonction fitness est calculée pour chaque chromosome dans les différentes étapes des AGs. Elle est donnée par:

$$Cost = F(chromosome) = F(P_1, P_2, P_3, \dots, P_M) \quad 3\text{-}5$$

### 3.4.4 Opérateurs génétiques

Afin de garantir le principe de la reproduction, il faut appliquer des opérateurs génétiques qui sont : la sélection, le croisement et la mutation [123].

**a. La sélection**

Elle permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais. Cet opérateur utilise la fonction fitness comme critère de sélection. Nous avons choisi la méthode de la roulette SRW qui donne les meilleurs résultats.

**b. Le croisement**

Le croisement, aussi nommé reproduction ou recombinaison, consiste à générer deux nouveaux chromosomes en recombinant l'information génétique de deux individus parents. II existe plusieurs méthodes pour effectuer la recombinaison génétique. Nous avons utilisé le plus simple et le plus connu qui est le croisement « à un seul point ».

**a. La mutation**

Le rôle de cet opérateur est de modifier aléatoirement, la valeur d'un composant de l'individu, avec une probabilité donnée. Cette probabilité est appelée le taux de mutation $P_m$ qui est généralement faible. Contrairement au croisement, la mutation est une opération qui implique un seul chromosome.

**b. Elitisme**

Utilisée plus récemment, l'élitisme est une idéologie qui fait en sorte d'assurer à chaque génération la survie des meilleurs individus de la population.

### 3.4.5 Critères d'arrêt

Les opérations génétiques précédentes seront exécutées autant de fois que le programme atteint un nombre maximum d'itérations « MaxIT » qu'il est généralement fixé à 1000 ou 2000 itérations en fonction de la longueur de la séquence optimisée. En effet, après un certain nombre d'itérations (généralement inférieure à 1000 ou inférieure à 2000) la séquence demeure inchangée ce qui en fait l'optimum. Ce dernier est ensuite mémorisé et l'algorithme est arrêté.

## 3.5 Optimisation des longues séquences à l'aide de la méthode OEP

La méthode OEP est semblable aux AGs vis-à-vis de la méthode de recherche qui est fondée sur la population et la recherche de la solution optimale en mettant à jour les générations. Cependant, sa stratégie de recherche est différente en comparaison avec les AGs.

La méthode OEP repose sur un ensemble de solutions ou particule disposées aléatoirement au début de l'algorithme. Chaque particule se situe à une position donnée dans l'espace de recherche et se déplace selon une certaine vitesse. Le mouvement de la particule est influencé par sa propre expérience et l'expérience des particules voisines. Les particules de l'essaim communiquent entre elles pour construire une solution à un problème, en s'appuyant sur leurs expériences collectives. La performance de chaque particule est mesurée au moyen d'une fonction fitness qui varie avec le problème d'optimisation.

Dans nos travaux, la méthode OEP est utilisée pour trouver les meilleures positions $P_{A_i}$ (première configuration des OLBSS) ou les meilleures positions $P_{B_i}$ (deuxième configuration des OLBSS). La fonction fitness est la même que dans le cas des AGs [123].

La position et la vitesse d'une particule, dans un espace de recherche à $M$ dimensions, sont définies par : $x_i = x_{i,1}, \dots, x_{i,M}$ et $v_i = v_{i,1}, \dots, v_{i,M}$ réspectivement. Chaque particule est caractérisée par sa meilleure position $pbest_i = pbest_{i,1}, \dots, pbest_{i,M}$ à l'itération $t$. La meilleure position atteinte par toutes les particules de l'essaim est notée $gbest = gbest_1, \dots, gbest_N$. La structure de base de la méthode OEP est illustrée sur le schéma de la Figure 3-9.

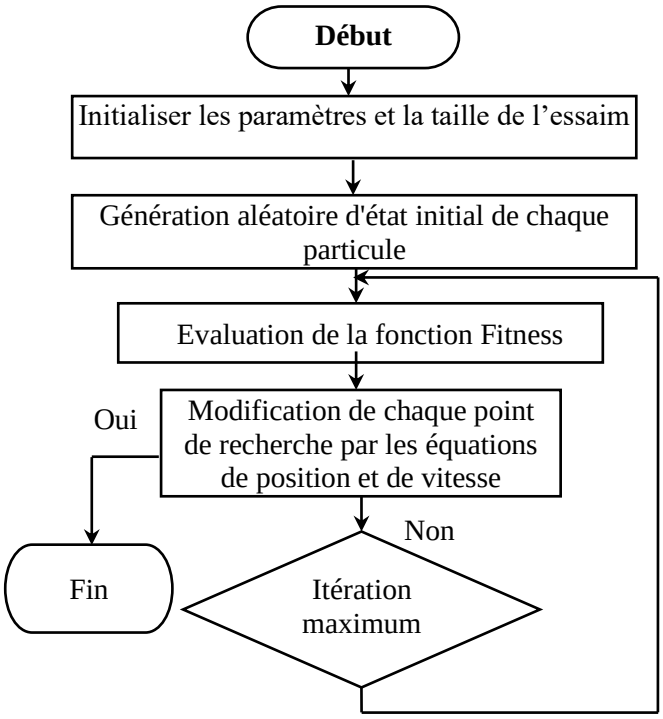



**Figure 3-9 Organigramme général de la méthode OEP**.

Les éléments de la méthode OEP implémentée peuvent être donnés comme suit :

### 3.5.1 Initialisation

Dans la population considérée ($c = 40$ particules), la $i^{ème}$ position de la particule est représentée par la $i^{ème}$ position de la sous-séquence $P_{A_i}$ pour la première configuration des OLBSS, ou par la $i^{ème}$ position de bit $P_{B_i}$ pour la seconde configuration des OLBSS ($i = 1, \dots, N_{pop}$). Les vitesses et les positions des particules sont initialisées aléatoirement [123].

### 3.5.2 Calcul de la valeur de fitness

La valeur de fitness de chaque particule est calculée par la même formule utilisée dans les AGs.

### 3.5.3 Mise à jour de la meilleure position $Pbest_i$

Pour chaque particule, on compare la valeur de fitness actuelle avec la valeur optimale locale. Si la valeur actuelle est meilleure, on remplace sa valeur de forme optimale par la valeur actuelle.

### 3.5.4 Mise à jour de *Gbest*

Pour chaque particule, si la valeur de fitness actuelle est meilleure que la valeur de fitness globale, on remplace le meilleur global par l'indice actuel dans le tableau des particules.

### 3.5.5 Mise à jour de la vitesse et de la position des particules

À l'itération t, après avoir trouvé les deux meilleures valeurs, la particule met à jour sa vitesse selon l'expression suivante:

$$v_{ij}^{t} = \chi\left[w * v_{ij}^{t-1} + c_1 r_1\left(Pbest_{ij}^{t-1} - x_{ij}^{t-1}\right) + c_2 r_2\left(Gbest_{ij}^{t-1} - x_{ij}^{t-1}\right)\right] \quad 3\text{-}6$$

La position de chaque particule, à la prochaine étape, est ensuite évaluée comme la somme de sa position actuelle et de la vitesse obtenue par l'Equation 3-7.

$$x_{ij}^{t} = x_{ij}^{t-1} + v_{ij}^{t} \quad 3\text{-}7$$

Où :

- $w$ est une constante appelée facteur d'inertie ;
- $v_{ij}^{t-1}$ est la vitesse de la particule $i$ dans la dimension $j$ à l'instant $t-1$;
- $c_1$ et $c_2$ sont des constantes d'accélération positives qui servent à ajuster la contribution des composantes cognitives et sociales, respectivement ;
- $r_1$ et $r_2$ sont des nombres aléatoires uniformément distribués dans l'intervalle [0, 1];
- $Pbest_{ij}^{t-1}$ est la meilleure position personnelle de la particule $i$ dans la dimension $j$ à l'instant $t-1$;
- $Gbest_{ij}^{t-1}$ est la meilleure position globale de la particule $i$ dans la dimension $j$ à l'instant $t-1$;
- $x_{ij}^{t-1}$ est la position de la particule $i$ dans la dimension $j$ à l'instant $t-1$;
- $\chi$ est le facteur de convergence, donné par:

$$\chi = (2\,k)/|2 - \phi - \sqrt{(\phi\text{^}2 - 4\phi)}| \quad 3\text{-}8$$

$\phi = \phi_1 + \phi_2 \geq 4$ ; $\phi_1$, $\phi_2$ et $k$ sont des facteurs de convergence, avec $k \in [0,1]$.

Dans la référence [121], de nombreux tests sont menés pour déterminer les valeurs optimales de $\phi_1$et $\phi_2$. En général, on utilise $\phi$ = 4,1 et $\phi_1 = \phi_2$, ce qui donne un coefficient $\chi = 0{,}7298844$.

Dans la référence [122] ont introduit un nouveau paramètre $V_{max}$ qui va permettre de contrôler l'éclatement du système i.e. que les particules se déplacent très rapidement d'une

région à une autre dans l'espace de recherche. Dans ce cas, il est possible que ce déplacement provoque les particules à sortir de l'espace de recherche.

$$v_{ij} \in [-V_{max}, V_{max}]$$

### 3.5.6 Critères d'arrêt

Le critère d'arrêt peut être différent suivant le problème posé. Dans ce travail, nous définissons un nombre maximum d'itération (égale à 1000 ou 2000 itérations) qui est fonction de la taille de la séquence optimisée [123].

Pour comparer entre ces deux méthodes, les paramètres fixés initialement sont identiques en termes de nombre de générations maximales et du nombre de population (essaim), les autres paramètres sont propres à chacune de ces méthodes (Tableau 3-1).

Notons que des valeurs différentes des paramètres peuvent conduire à des résultats meilleurs ou moins bons, selon le problème. Dans notre cas, après plusieurs expériences, nous avons choisi les paramètres indiqués dans le Tableau 1-3 [123].

**Tableau 3-1 Paramètres de réglages d'OEP et des AGs.**

| **Paramètres de l'OEP** | | **Paramètres des AGs** | |
|---|---|---|---|
| La taille de la population | *40* | La taille de la population | 40 |
| Nombre maximal de générations | *1000* | Nombre maximal de générations | 1000/2000 |
| Vitesse maximale | *30* | Croisement | à un seul point |
| Constantes d'accélération $(\boldsymbol{c_1}, \boldsymbol{c_2})$ | *1* | Taux de croisement | 1 |
| facteurs de convergence | $K = 1$, $\phi_1 = \phi_2 = 2.05$ | Mutation | Uniform |
| | | Taux de mutation | 0.02 |

## 3.6 Démonstration mathématique de la méthode proposée

Comme nous l'avons déjà dit, les sous-séquences de Walsh-Hadamard sont parmi les structures orthogonales les plus simples à construire.

Supposons que nous souhaitons réaliser une longue séquence optimisée à partir d'un ensemble de 8 sous-séquences de Hadamard de 8 bits chacune. La longue séquence initiale est donnée par le vecteur suivant [123]:

$$S_L = [\boldsymbol{p}\ -\boldsymbol{p}\ \ \boldsymbol{p}\ \ -\boldsymbol{p}\ \ \boldsymbol{p}\ -\boldsymbol{p}\ \ \ \boldsymbol{p}\ -\boldsymbol{p}\ \ \boldsymbol{p}\ \ \boldsymbol{p}\ -\boldsymbol{p}\ \ -\boldsymbol{p}\ \ \boldsymbol{p}\ \ \boldsymbol{p}\ \ -\boldsymbol{p}\ \ -\boldsymbol{p}\ldots..\boldsymbol{p}\ \ \boldsymbol{p}\ \ \boldsymbol{p}\ \ \boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}] \quad 3\text{-}9$$

1ère sous-séquence "$S_{L1}$" 2ème sous-séquence "$S_{L2}$" ….. 8ème sous-séquence"$S_{L8}$"

Dans cette Equation, $p$ représente la polarité positive du bit et $-p$ la polarité négative.

Redéfinissons d'abord la fonction de corrélation et les critères d'orthogonalité. Soit $Had_1$ et $Had_2$ deux sous-séquences de Hadamard distinctes, alors leurs FAC et FIC numériques sont respectivement données comme suit:

$$ACF_{Had_1}(m) = \sum_{n=0}^{N-1} Had_1(n)Had_1(n-m) \quad 3\text{-}10$$

$$ACF_{Had_2}(m) = \sum_{n=0}^{N-1} Had_1(n)Had_2(n-m) \quad 3\text{-}11$$

$$CCF_{Had_1Had_2}(m) = \sum_{n=0}^{N-1} Had_1(n)Had_2(n-m) \quad 3\text{-}12$$

Pour que les deux séquences $\mathrm{Had}_1$ et $\mathrm{Had}_2$ soient orthogonales, elles doivent satisfaire la condition donnée par:

$$\sum_{n=0}^{N-1} Had_1(n)Had_2(n) = 0 \quad 3\text{-}13$$

Maintenant, si l'on considère le cas de la concaténation, la construction de la longue séquence proposée est obtenue en optimisant les positions des sous-séquences dans la longue séquence initiale donnée par l'équation 3-9 en utilisant les AGs ou la méthode OEP. Cette optimisation est basée principalement sur la minimisation des amplitudes des pics secondaires situés dans les côtés gauche et droit du pic central de la FAC de la longue séquence initiale. La m$^{ième}$ amplitude du pic secondaire (à partir du pic central) est obtenue en décalant $S_L$ circulairement d'une valeur égale à $m$, à droite ($m > 0$) ou à gauche ($m < 0$), et en faisant le produit, élément par élément, entre $S_L$ et sa version décalée, puis en sommant les résultats obtenus.

Considérons maintenant les cas qui correspondent à des valeurs intéressantes de la valeur de décalage $m$.

- **1er cas : pour $m$ = 0**

Dans ce cas, on obtient la valeur maximale de la FAC correspondant à l'amplitude du pic central (c'est-à-dire l'énergie maximale), qui est égale à 64 (8 × 8).

- **2ème cas : pour $m$ un multiple de la longueur de la sous-séquence de Hadamard (c'est-à-dire que $m$ est un multiple de 8) [123]**

Dans ce cas, la valeur de la FAC est égale à zéro. Ceci est simplement dû à la propriété d'orthogonalité des sous-séquences de Hadamard illustrée dans l'Equation 3-13.

- **3ème cas : pour $m > 0$ et m n'est pas un multiple de 8 [123]**

On note ici, que les valeurs de décalage de $m = 1$ à $m = 63$ correspondent, respectivement, aux pics secondaires (de 1er au 63ième) dont les amplitudes peuvent être calculées comme illustré dans l'exemple suivant :

Prenons par exemple $m = 3$. La version décalée circulairement de $S_L$ est donnée comme suit:

$$SS_L = [\underbrace{-\boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}\ \ \boldsymbol{p}\ -\boldsymbol{p}\ \ \boldsymbol{p}\ \ -\boldsymbol{p}\ \ \boldsymbol{p}}_{1^{\text{iére}}\text{ “8 éléments” } (SS_{L1})}/\underbrace{-\boldsymbol{p}\ \ \boldsymbol{p}\ -\boldsymbol{p}\ \ \boldsymbol{p}\ \ \boldsymbol{p}\ -\boldsymbol{p}\ \ -\boldsymbol{p}\ \ \boldsymbol{p}}_{2^{\text{ieme}}\text{ “8 éléments” } (SS_{L2})}/\ \boldsymbol{p}\ \ -\boldsymbol{p}\ \ -\boldsymbol{p}\ldots\boldsymbol{p}\ \ \boldsymbol{p}\ \ \boldsymbol{p}\ \ \boldsymbol{p}\ -\boldsymbol{p}\,] \quad 3\text{-}14$$

Maintenant, si nous réalisons le produit, élément par élément, de $S_{\mathrm{L1}}$ par sa version décalée $SS_{\mathrm{L1}}$, comme illustré dans les Equations 3-15 et 3-16, respectivement, nous obtenons le résultat donné dans l'Equation 3-17.

$$S_{L1} = [\ \boldsymbol{p}\ \ -\boldsymbol{p}\ \ \ \boldsymbol{p}\ \ -\boldsymbol{p}\ \ \ \boldsymbol{p}\ \ -\boldsymbol{p}\ \ \ \boldsymbol{p}\ \ -\boldsymbol{p}\,] \quad 3\text{-}15$$

$$SS_{L1} = [-\boldsymbol{p}\ \ -\boldsymbol{p}\ \ -\boldsymbol{p}\ \ \ \boldsymbol{p}\ \ -\boldsymbol{p}\ \ \ \boldsymbol{p}\ \ -\boldsymbol{p}\ \ \ \boldsymbol{p}\,] \quad 3\text{-}16$$

$$S_{L1} \times SS_{L1} = [-\boldsymbol{p}\ \ \ \boldsymbol{p}\ \ -\boldsymbol{p}\ \ -\boldsymbol{p}\ \ -\boldsymbol{p}\ \ -\boldsymbol{p}\ \ -\boldsymbol{p}\ \ -\boldsymbol{p}] \quad 3\text{-}17$$

La séquence décalée dans l'Equation 3-16 se compose de deux ensembles d'éléments ; le premier ensemble correspond aux trois derniers éléments de $S_{L8}$ et le second correspond aux cinq premiers éléments de $S_{L1}$. Dans le vecteur de produits résultant, donné par

l'Equation 3-17, nous avons sept éléments de polarités négatives et un seul élément de polarité positive. Ainsi, la somme de ces éléments donne la valeur « -6 » qui est très proche (en valeur absolue) de la taille de la sous-séquence de Hadamard (8) et donc indésirable.

Ensuite, si nous réalisons le produit, élément par élément, de $S_{L2}$ avec sa version décalée $SS_{L2}$, donnée, réspéctivement, par les Equations 3-18 et 3-19, nous obtenons le vecteur produit résultant donné dans l'Equation 3-20. Ici, les nombres de polarités négatives et positives sont égaux. Par conséquent, la somme des éléments vectoriels est nulle, ce qui est très bénéfique pour l'optimisation de notre longue séquence.

$$S_{L2} = [\ \boldsymbol{p} \quad \boldsymbol{p} \ -\boldsymbol{p} \ -\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \ -\boldsymbol{p} \ -\boldsymbol{p}\ ] \tag{3-18}$$

$$SS_{L2} = [-\boldsymbol{p} \quad \boldsymbol{p} \ -\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \ -\boldsymbol{p} \ -\boldsymbol{p} \quad \boldsymbol{p}] \tag{3-19}$$

$$S_{L2} \times SS_{L2} = [-\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \ -\boldsymbol{p} \quad \boldsymbol{p} \quad -\boldsymbol{p} \quad \boldsymbol{p} \ -\boldsymbol{p}] \tag{3-20}$$

- **4éme cas : pour $m < 0$ et $m$ n'est pas un multiple de 8[123]**

Prenons comme exemple le cas de $m = -2$, la version décalée circulairement de $S_L$est donnée comme suit:

$$SS_L = [\ \underbrace{\boldsymbol{p} \ -\boldsymbol{p} \ \boldsymbol{p} \ \boldsymbol{p} \ -\boldsymbol{p} \ -\boldsymbol{p} \ \boldsymbol{p} \ \boldsymbol{p}}_{SS_{L1}} / \underbrace{-\boldsymbol{p} \ -\boldsymbol{p} \ldots\ldots \boldsymbol{p} \ \boldsymbol{p} \ \boldsymbol{p} \ \boldsymbol{p} \ -\boldsymbol{p} \ -\boldsymbol{p} / -\boldsymbol{p} \ -\boldsymbol{p} \ \boldsymbol{p} \ -\boldsymbol{p} \ \boldsymbol{p} - \boldsymbol{p} \ \boldsymbol{p} \ -\boldsymbol{p}}_{SS_{L8}}\ ] \tag{3-21}$$

En utilisant le même raisonnement que précédemment, $S_{L1}$, $SS_{L1}$, $S_{L8}$ et $SS_{L8}$ sont respectivement donnés par les équations 3-22, 3-23, 3-24 et 3-25.

$$S_{L1} = [\ \boldsymbol{p} \quad -\boldsymbol{p} \quad \boldsymbol{p} \ -\boldsymbol{p} \quad \boldsymbol{p} \quad -\boldsymbol{p} \quad \boldsymbol{p} \quad -\boldsymbol{p}\ ] \tag{3-22}$$

$$SS_{L1} = [\ \boldsymbol{p} \ -\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \quad -\boldsymbol{p} \ -\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p}\ ] \tag{3-23}$$

$$S_{L8} = [\ \boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \ -\boldsymbol{p} \ -\boldsymbol{p} \ -\boldsymbol{p} \ -\boldsymbol{p}] \tag{3-24}$$

$$SS_{L8} = [\ -\boldsymbol{p} \ -\boldsymbol{p} \quad \boldsymbol{p} \ -\boldsymbol{p} \quad \boldsymbol{p} \quad -\boldsymbol{p} \quad \boldsymbol{p} \ -\boldsymbol{p}] \tag{3-25}$$

Ensuite, les vecteurs des produits suivants sont calculés et sont donnés par :

$$S_{L1} \times SS_{L1} = [\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \quad -\boldsymbol{p} \quad -\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \quad -\boldsymbol{p}\ ] \tag{3-26}$$

$$S_{L8} \times SS_{L8} = [-\boldsymbol{p} \quad -\boldsymbol{p} \quad \boldsymbol{p} \quad -\boldsymbol{p} \quad -\boldsymbol{p} \quad \boldsymbol{p} \quad -\boldsymbol{p} \quad \boldsymbol{p}\,] \qquad 3\text{-}27$$

Les sommes des éléments des vecteurs, donnés par les équations 3-26 et 3-27 sont respectivement égales à 2 et -2. Par conséquent, les deux valeurs obtenues sont très bénéfiques pour l'optimisation de notre longue séquence. D'après les exemples présentés, nous voyons que, pour certaines valeurs du décalage $m$, on peut trouver des valeurs petites ou nulles des sommes des produits entre $S_L$ et sa version décalée $SS_L$.

Maintenant si, pour chaque valeur de $m$, on fait la même chose pour chaque couple de sous-séquences $S_{Li}$ et $SS_{Li}$ $(1 \leq i \leq 8)$ alors, la qualité de $S_L$ dépendra évidemment des résultats des produits $S_{L1} \times SS_{L1}, S_{L2} \times SS_{L2}, \ldots$ et $S_{L8} \times SS_{L8}$ pour toutes les valeurs de $m$. De plus, comme chaque $SS_{Li}$ $(1 \leq i \leq 8)$ est composée de deux sous-séquences adjacentes, la valeur de $S_{Li} \times SS_{Li}$ $(1 \leq i \leq 8)$ change chaque fois que la disposition des 8 sous-séquences est modifiée. Par conséquent, pour améliorer la qualité globale de la longue séquence initiale, nous devons tester autant d'arrangements différents de sous-séquences que possible et choisir ceux qui assurent les valeurs de pics latéraux les plus basses et, par conséquent, les plus appropriées. Ce processus d'optimisation est réalisé par l'utilisation des AGs ou par la méthode OEP.

## 3.7 Résultats de simulation

Des simulations sont effectuées pour évaluer les performances des OLBSS proposées. Pour cette raison cinq scénarios ont été donnés.

### 2.5.1 Premier scénario

Dans le premier scénario [123], on construit une nouvelle classe d'OLBSS de longueur 1024 bits par la concaténation (premier type) ou l'entrelacement (deuxième type) de 32 sous-séquences de Walsh-Hadamard de 32 bits chacune. Puis, en compare l'OLBSS obtenue avec une séquence de Gold traditionnelle de même longueur. Les techniques d'optimisation par les AGs et la méthode OEP sont utilisées séparément. Les paramètres de simulation, pour les AGs et la méthode OEP, sont donnés dans le Tableau 3-1.

Les variations de la fonction fitness, correspondant à l'optimisation de la séquence OLBSS de longueur 1024 bits, pour les quatre cas de construction de sous-séquences (entrelacées / concaténées et de méthodes AGs / OEP), sont présentées dans la Figure 3-10.

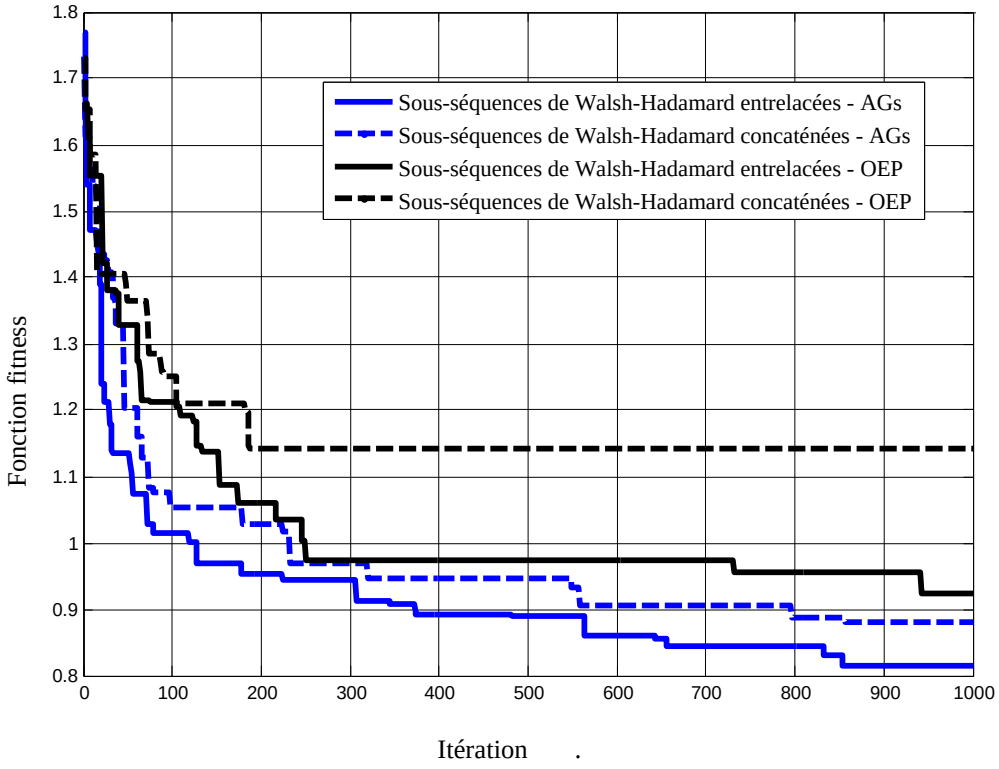



**Figure 3-10 Fonctions fitness pour les OLBSS de 1024 bits, composée chacune de 32 sous-séquences de Walsh-Hadamard de 32 bits chacune (concaténées/entrelacées) optimisées par les AGs et la méthode OEP.**

On observe, à partir de la Figure 3-10, que les valeurs des fonctions fitness correspondant à l'application des AGs sont plus faibles que celles utilisant la méthode OEP pour toutes les générations. Donc, les séquences OLBSS obtenues par l'application des AGs présentent de meilleures performances par rapport aux OLBSS obtenues par l'application de la méthode OEP. En effet, la meilleure valeur de fitness, dans le cas des sous-séquences entrelacées (AGs), est de 0,8145 et elle apparait dans la 854$^{ème}$ itération. Cette valeur est de 0,8811 pour les sous-séquences concaténées (AGs). Pour la méthode OEP, la fonction fitness pour les sous-séquences entrelacées possède la valeur la plus faible qui est de 0,9243. Cette valeur apparait dans la 942$^{ème}$ itération.

Pour confirmer la supériorité des performances des OLBSS construites par les sous-séquences de Walsh-Hadamard, comme la montre la Figure 3-11, leurs FAC sont comparées à

celle du code de Gold de même longueur et à celle de l'OLBSS initiale proposée (avant optimisation).

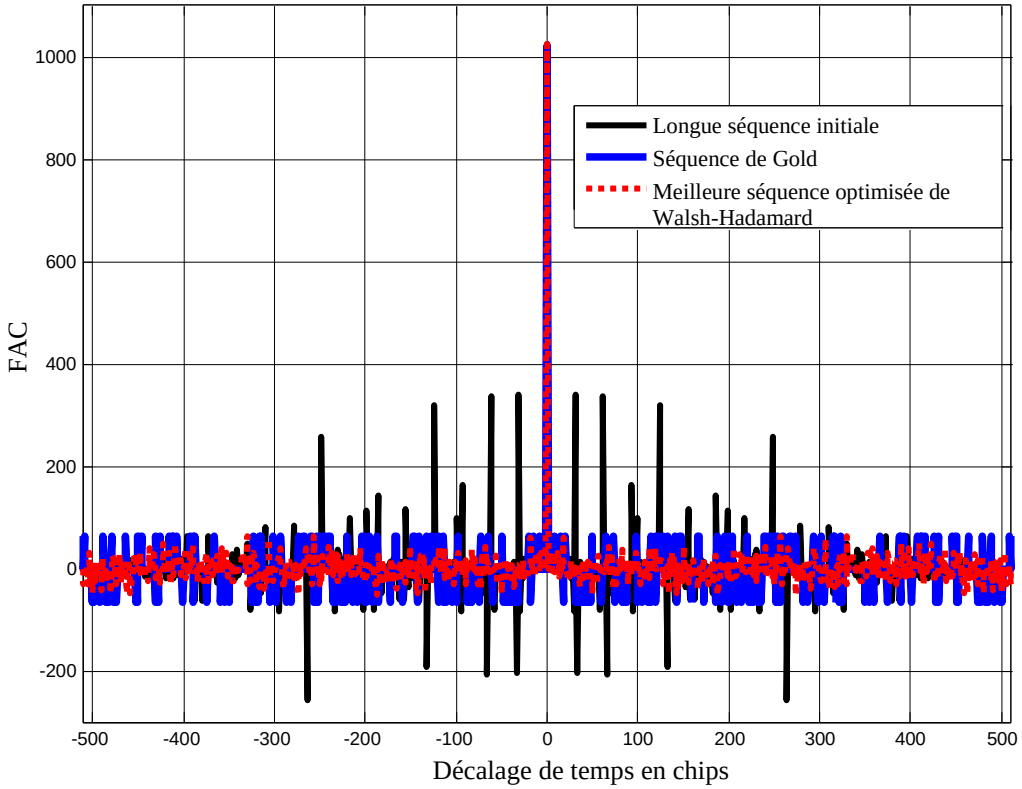



**Figure 3-11 Comparaison entre la FAC de l'OLBSS composée par les sous-séquences de Walsh-Hadamard et celles de la séquence de Gold et la séquence proposée avant optimisation.**

De plus, dans la Figure 3-12, nous comparons les histogrammes des valeurs des FACs sans pics centraux pour la séquence OLBSS optimisée et une autre séquence aléatoire.

A partir de la Figure 3-11, on peut observer que l'OLBSS construite par des sous-séquences de Walsh-Hadamard, présente les meilleures caractéristiques de la FAC par rapport à la séquence de Gold originale étant donné qu'elle présente une diminution significative des niveaux des pics secondaires ayant des amplitudes élevées. Ce résultat peut être facilement expliqué, à partir de la Figure 3-12, où l'histogramme des valeurs des FACs sans pic central de la séquence aléatoire est plus étalé par rapport à l'OLBSS. En effet, la majorité des pics secondaires, ayant des amplitudes supérieures à 60, sont complètement éliminés.

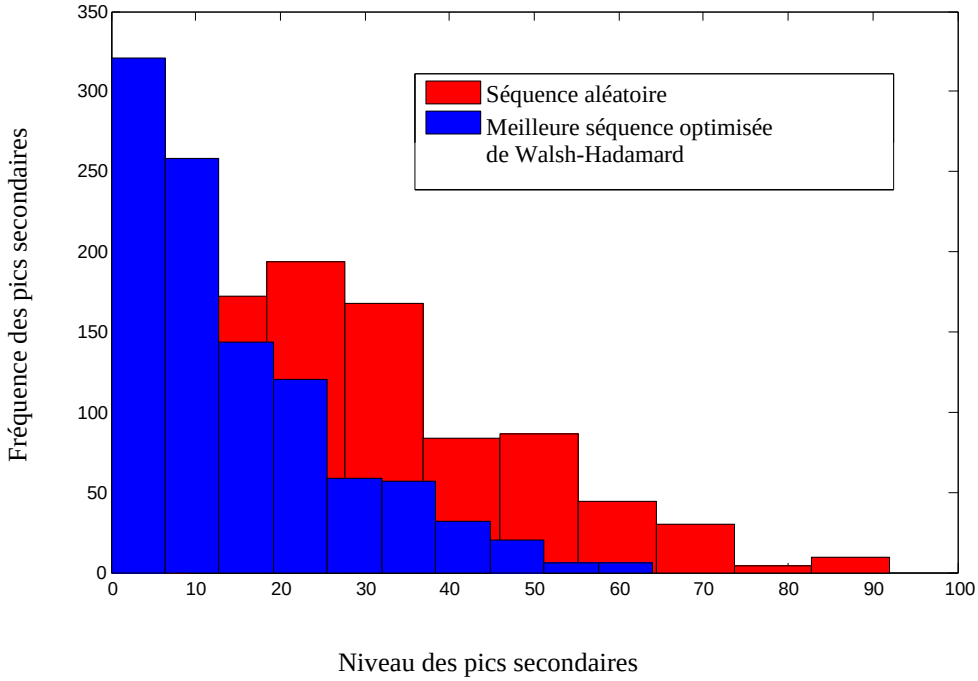



**Figure 3-12 Comparaison entre l'histogramme des valeurs de la FAC sans pic central de l'OLBSS composée des sous-séquences de Walsh-Hadamard et celui des valeurs de la FAC sans pic central d'une séquence aléatoire.**

### 2.5.2 Deuxième scénario

Dans le deuxième scénario, la même expérience précédente est répétée mais cette fois-ci avec l'utilisation de sous-séquences de même longueur appartenant à la famille de Gold. Par conséquent, ces dernières sont concaténées ou entrelacées pour construire une longue séquence qui est optimisée en utilisant les AGs ou la méthode OEP. Les fonctions fitness sont dérivées pour les mêmes combinaisons que celles du premier scénario. Ici, les paramètres de simulation sont les mêmes que ceux donnés dans le Tableau 3-1.

Un nombre de 33 sous-séquences de Gold, de 31 bits chacune, est utilisé pour générer une séquence de longueur 1023 bits [123].

La Figure 3-13 illustre la comparaison des évolutions des fonctions fitness données par les différents algorithmes lors du processus d'optimisation [123].

La Figure 3-14 illustre la comparaison entre les fonctions fitness de l'OLBSS composée des sous-séquences de Walsh-Hadamard (entrelacées) optimisée par les AGs et l'OLBSS composée des sous-séquences de Gold (concaténées) optimisée par les AGs [123].

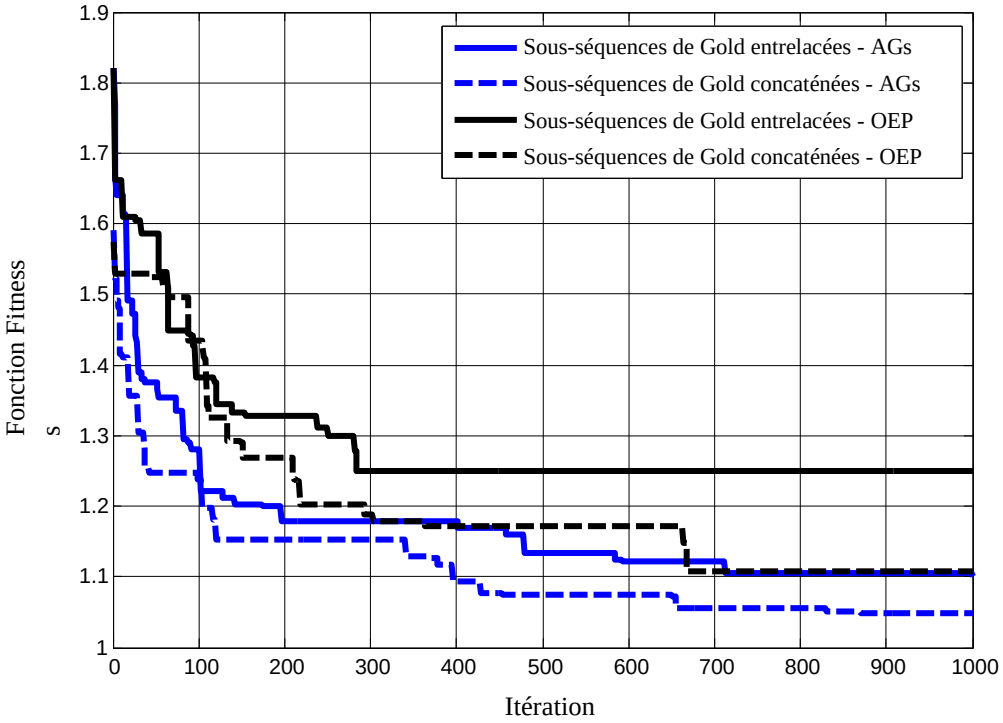



**Figure 3-13 Fonctions fitness pour les OLBSS de 1023 bits, composée chacune de 33 sous-séquences de Gold de 31 bits chacune (concaténées/entrelacées) optimisées par les AGs et la méthode OEP.**

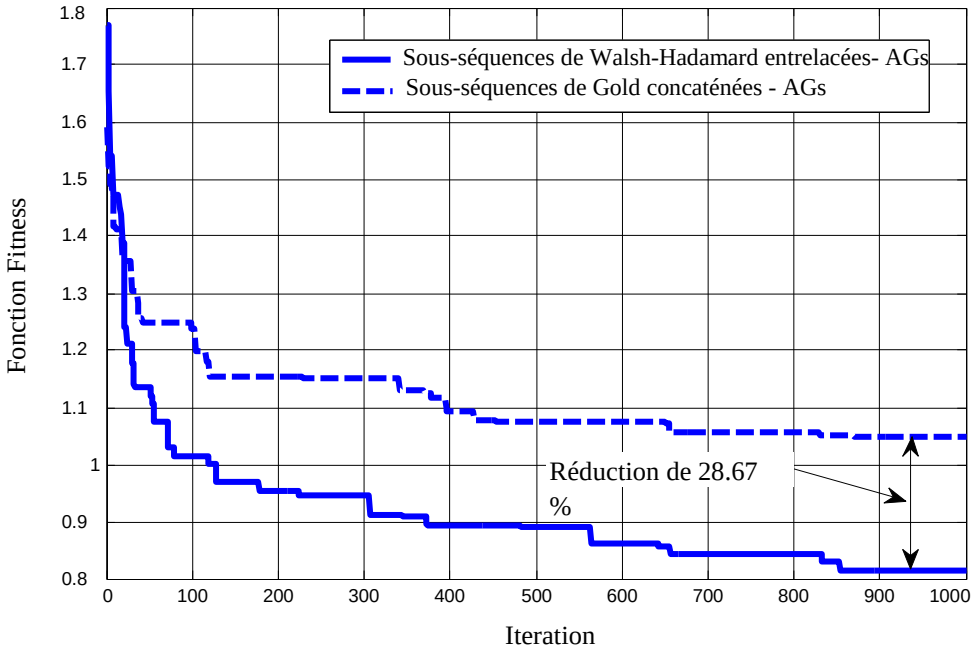



**Figure 3-14 Comparaison entre les fonctions fitness de l'OLBSS composée des sous-séquences de Walsh-Hadamard (entrelacées) optimisée par les AG et l'OLBSS composée des sous-séquences de Gold (concaténées) optimisée par les AG.**

Comme la montre la Figure 3-13, on peut observer que la meilleure valeur de la fonction fitness, correspondant à l'OLBSS construite à partir des sous-séquences de Gold concaténées et optimisée par les AG, est de 1,051. Elle apparait dans la 836$^{ème}$ itération. Par rapport au cas de Walsh-Hadamard, comme le montre la Figure 3-14, cette valeur correspond

à une augmentation de 28,67%. En effet, les sous-séquences de Walsh-Hadamard présentent de meilleures performances en termes de FIC ce qui a une influence directe sur la FAC de l'OLBSS construite.

### 2.5.3 Troisième scénario

Dans le troisième scénario, nous étudions l'effet du nombre de sous-séquences utilisées pour construire l'OLBSS. Pour cette raison, nous construisons deux différentes longues séquences OLBSS de longueur 1024 bits chacune et qui sont dérivées de sous-séquences Walsh-Hadamard entrelacées. Ici, on utilise les AGs pour l'optimisation de la séquence construite.

Pour la première longue séquence, on utilise 32 sous-séquences de 32 bits chacune, tandis que pour la deuxième, on utilise 16 sous-séquences de 64 bits chacune. L'évolution de la fonction fitness, correspondant à chaque cas, est illustrée sur la Figure 3-15 [123].

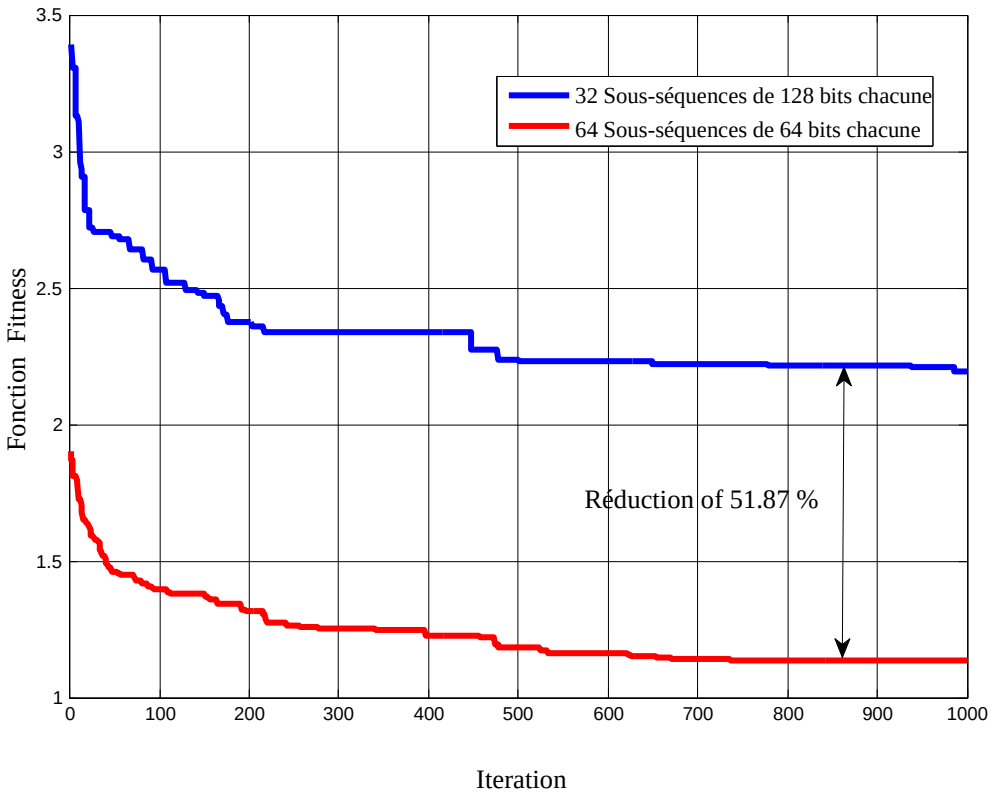



**Figure 3-15 Comparaison entre les fonctions fitness correspondant aux deux meilleures OLBSS (Walsh-Hadamard /entrelacé- AG) de longueur 4096 bits chacune composées respectivement de 64 sous-séquences de 64 bits chacune et de 32 sous-séquences de 128 bits chacune.**

On observe, à partir de la Figure 3-15, que pour une longueur d'OLBSS fixe donnée, lorsque le nombre de sous-séquences est augmenté de moitié, il y a une diminution d'environ 51,87% des valeurs de la fonction fitness et donc, les performances d'OLBSS, en termes de

réduction des niveaux de pics secondaires, deviennent supérieures. En conséquence, un choix judicieux des longueurs des sous-séquences doit être considéré comme un paramètre intéressant peut améliorer la qualité des OLBSS.

En plus de générer des OLBSS de longueur moyenne comme cela a été fait plus haut, nous pouvons également utiliser notre méthode proposée, comme illustré dans le scénario suivant, pour générer et optimiser des séquences plus longues avec de meilleures performances.

### 2.5.4 Quatrième scénario

Dans ce scénario, nous utilisons le processus d'entrelacement des sous-séquences de Walsh-Hadamard plus courtes en conjonction avec les AGs pour générer une OLBSS plus longue de taille 8192 bits. La comparaison de la FAC de l'OLBSS obtenue avec celles des séquences aléatoires et Gold de même longueur est illustrée sur la Figure 3-16 [123].

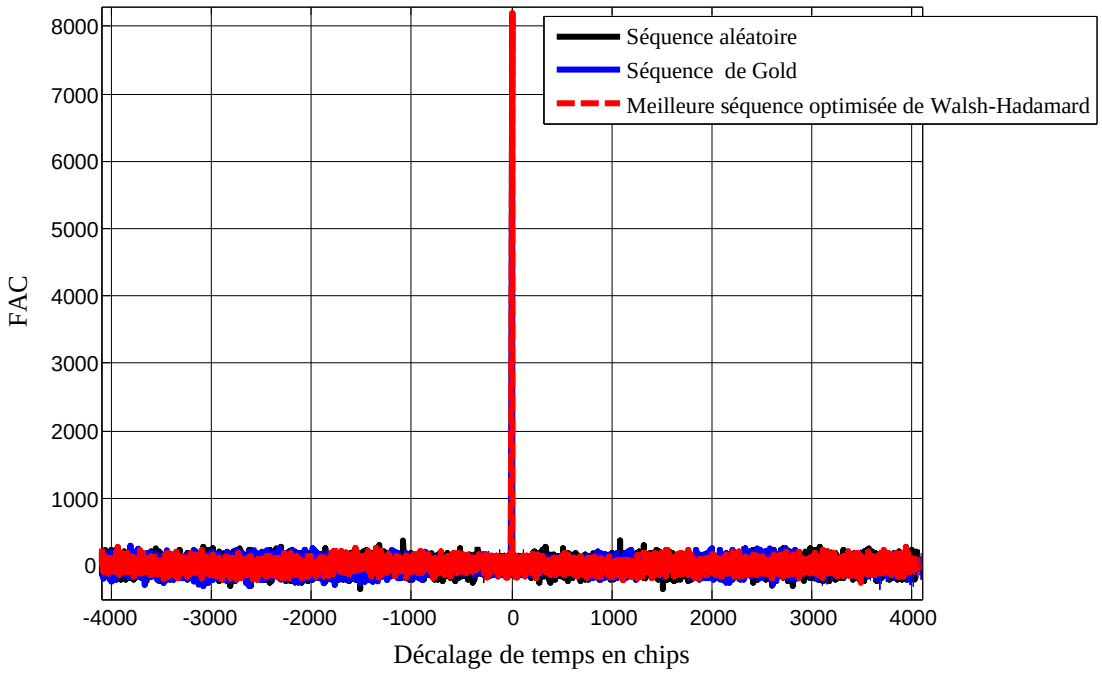



**Figure 3-16 Comparaison entre les FAC des meilleures OLBSS (Walsh-Hadamard/ entrelacé -AGs) de longueur 8192 bits et celles de Gold et aléatoire de même longueur.**

Comme illustré dans cette Figure, nous observons directement la différence. En fait, l'OLBSS, obtenue à partir de l'entrelacement de 64 sous-séquences de longueur 128 chacune, présente une FAC avec un niveau réduit des pics secondaires. Ceci peut être confirmé par la Figure 3-17 où les histogrammes des séquences aléatoire et de Gold sont plus étalés en comparaison avec celui de l'OLBSS.

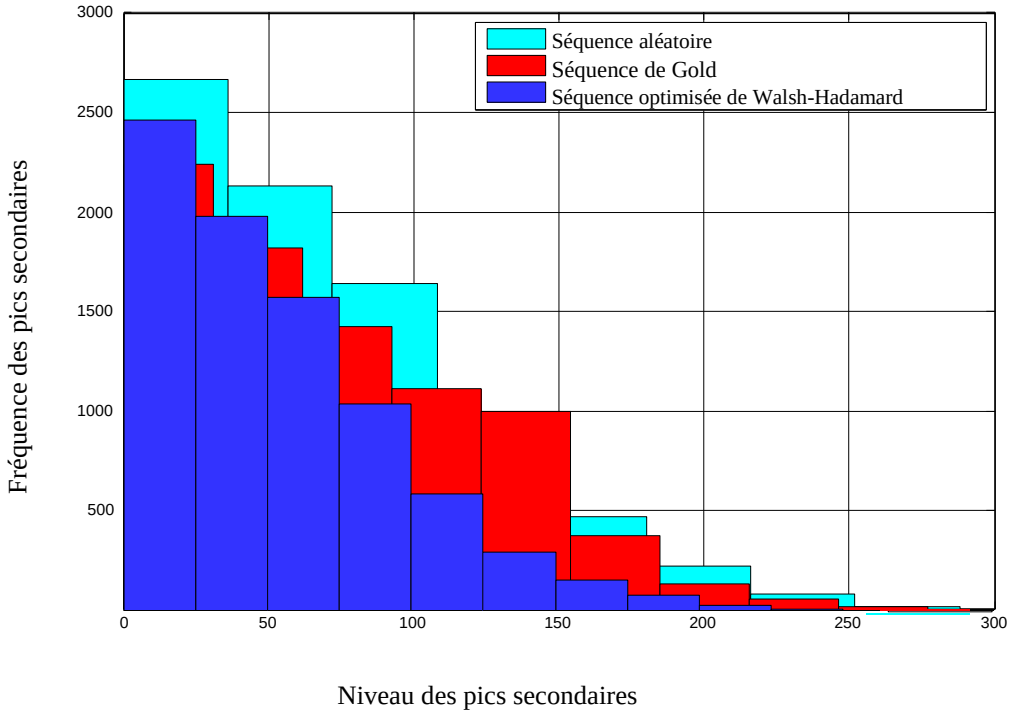



**Figure 3-17 Comparaison entre l'histogramme de la FAC sans pic central de l'OLBSS de longueur 8192 bits composée par des sous-séquences de Walsh-Hadamard et ceux des FACs sans pic central des séquences aléatoire et de Gold de même longueur.**

La Figure 3-18 illustre, l'évolution de la fonction fitness donnée par les AGs pour l'OLBSS de taille 8192 bits lors du processus d'optimisation [123].

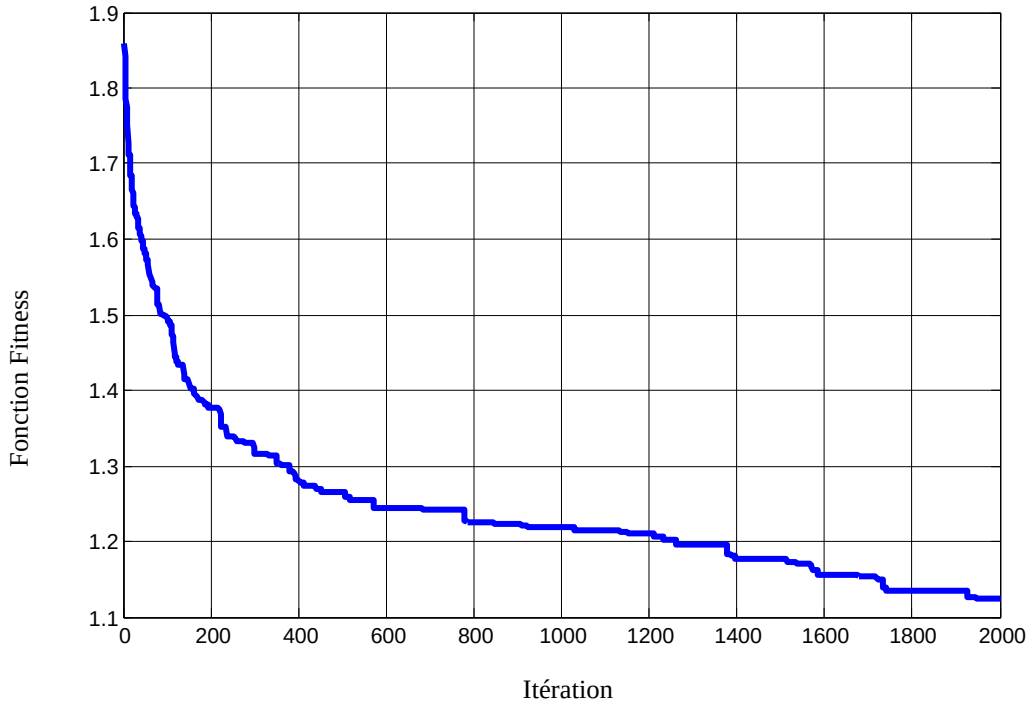



**Figure 3-18 Evolution de la fonction fitness pour l'optimisation, par les AGs, d'une séquence de longueur 8192 bits composée de 64 sous-séquences de Walsh-Hadamard entrelacées de 128 bits chacune.**

Comme le montre cette Figure, on peut observer que la meilleure valeur de la fonction fitness est égale à 1,125. Elle apparait dans la 1948ème itération. On note ici qu'en raison de la grande longueur de la séquence obtenue, nous avons utilisé 2000 itérations dans le processus d'optimisation.

### 2.5.5 Cinquième scénario

Dans ce dernier scénario, l'efficacité de la méthode proposée est testée en faisant varier la longueur de la séquence générée. A cet effet, un nombre significatif de séquences de longueurs différentes a été considéré. Ces dernières, ont été optimisées en utilisant des sous-séquences de Walsh-Hadamard ou de Gold en conjonction avec les AGs. Les séquences obtenues sont ensuite comparées, respectivement, avec des séquences aléatoires de mêmes tailles. Pour renforcer nos résultats, certaines séquences optimisées ont également été comparées avec leurs séquences de Gold ou de Weil correspondantes. Les résultats des comparaisons sont présentés dans les Tableaux 3-2 et 3-3. Notons que la plupart des séquences considérées dans les deux tableaux sont des séquences utilisées dans les applications GNSS, et leurs longueurs sont généralement choisies pour être une puissance entière de deux (c'est-à-dire de la forme $2^n$) pour faciliter le calcul à base des fonctions FFT.

**Tableau 3-2 Comparaison, à plusieurs échelles, entre certaines séquences optimisées et des séquences aléatoires.**

| Longueur de la séquence proposée | Type de sous-séquence | Nombre de sous-séquences | Longueur de la sous-séquence | Valeur de *F* | | Taux d'amélioration en% |
|---|---|---|---|---|---|---|
| | | | | OLBSS proposée | Séquence aléatoire | |
| **512** | Walsh | 16 | 32 | **0,64** | 2,16 | 70,37 |
| **1024** | Walsh | 32 | 32 | **0,81** | 2,08 | 61,06 |
| **4096** | Walsh | 64 | 64 | **0,96** | 2,07 | 53,62 |
| **5115** | Gold | 5 | 1023 | **1.89** | 2.10 | 10,22 |
| **10230** | Gold | 10 | 1023 | **1,88** | 2.13 | 11,74 |
| **16384** | Walsh | 128 | 128 | **1,24** | 1,99 | 37,69 |
| **65536** | Walsh | 256 | 256 | **1,48** | 2,01 | 26,37 |
| **262144** | Walsh | 512 | 512 | **1,65** | 2 | 17,50 |
| **262143** | Gold | 513 | 511 | **1.7475** | 2.0019 | 13 |

D'après le Tableau 3-2, l'étude comparative, basée sur la valeur du facteur F, montre que toutes les séquences aléatoires de différentes tailles ont presque la même valeur de F (environ 2). Cependant, les séquences optimisées, quelles que soient leurs tailles, ont des valeurs du FM beaucoup plus petites. Cela prouve la validité de notre méthode proposée pour

générer des séquences avec n'importe quelle longueur. En fait, le taux d'amélioration des séquences optimisées varie de 11,74% (pour les séquences très longues) à 70,37% pour les séquences courtes. De plus, contrairement aux séquences aléatoires, les bits +1 et -1 dans les séquences proposées (si l'on exclut la première et la dernière ligne de la matrice d'Hadamard du processus d'optimisation) sont bien équilibrés puisque la différence entre leurs nombres est nulle quelle que soit la taille de la séquence optimisée.

**Tableau 3-3 Comparaison entre certaines séquences optimisées et des séquences de Gold ou de Weil à plusieurs échelles.**

| **Longueur de la séquence proposée** | **Type de sous-séquence** | **Nombre de sous-séquence** | **Longueur de la sous-séquence** | **Type et longueur de séquence traditionnelle** | **Valeur de *F*** | | **Taux d'amélioration en%** |
|---|---|---|---|---|---|---|---|
| | | | | | **OLBSS proposée** | **Séquence de Gold ou de Weil** | |
| **512** | Walsh | 16 | 32 | Gold / 511 | **0,64** | 2,11 | 69,67 |
| **1023** | Gold | 33 | 31 | Gold / 1023 | **1,05** | 1,88 | 44,15 |
| **1024** | Walsh | 32 | 32 | Gold / 1023 | **0,81** | 1,88 | 56,91 |
| **4096** | Walsh | 64 | 64 | Gold / 4095 | **0,96** | 2,11 | 54,50 |
| **10230** | Gold | 10 | 1023 | Weil / 10230 | **1,88** | 2,43 | 22,63 |

A partir du Tableau 3-3, l'étude comparative, basée sur le même critère F, montre que quelles que soient leurs tailles, les séquences proposées ont des valeurs de F qui sont bien inférieures à celles des séquences de Gold ou de Weil. En effet, le taux d'amélioration des séquences optimisées varie de 22,63% (pour les séquences très longues) à 69,67% pour les séquences courtes.

## 3.8 Conclusion

Dans ce chapitre, une méthode efficace, pour générer de longues séquences d'étalement optimisées, avec de propriétés de corrélation améliorées, est proposée. Dans ces conditions, une séquence longue initiale est générée à partir d'un certain nombre de sous-séquences binaires plus courtes et de même longueur appartenant à la famille des séquences de Walsh-Hadamard ou de Gold. Pour améliorer les performances des séquences générées, les AGs ou la méthode OEP ont été utilisés afin d'optimiser les caractéristiques de leurs FAC. Deçà, le processus de construction de la séquence longue, effectué soit par concaténation ou entrelacement des sous-séquences, donne naissance à deux types d'OLBSS.

Selon les résultats obtenus, les nouvelles OLBSS, composées de séquences de Walsh-Hadamard ou de Gold, ont de bonnes propriétés de la FAC en comparaison avec les séquences aléatoires et de Gold originales de même tailles. De plus, l'étude comparative a montré que l'OLBSS, basée sur l'optimisation par les AGs et les sous-séquences de Walsh-Hadamard entrelacées, présente les meilleures performances.

Concernant les OLBSS très longues, les résultats de simulation ont montré que l'utilisation d'un nombre plus grand de sous-séquences a pour effet l'amélioration des performances. En effet, lorsque le nombre de sous-séquences augmente de moitié, il y a une diminution d'environ 51,87% des valeurs de la fonction fitness et donc, l'OLBSS obtenue présente de meilleures performances en termes de réduction des niveaux des pics secondaires dans la FAC.

Finalement, alors que les séquences OLBSS optimisées, quelles que soient leurs tailles, présentent des valeurs beaucoup plus petites du facteur F, toutes les séquences aléatoires ont presque la même valeur de F (environ 2) pour n'importe quelle taille de la séquence. Ceci démontre la validité de notre méthode proposée pour générer des séquences avec n'importe quelle longueur. En effet, d'après les résultats obtenus, le taux d'amélioration varie de 11,74% à 70,37%. En comparaison avec les séquences de Gold ou de Weil, les séquences OLBSS optimisées montrent un taux d'amélioration qui varie de 22,63% à 69,67%.

# Conclusion générale

Dans les systèmes de communication CDMA, afin de récupérer les informations proportionnelles à chaque utilisateur, il est important que les signaux des différents utilisateurs soient les plus décorrélés possible les uns des autres. Pour cela, le choix des codes d'étalement est une étape primordiale durant l'opération de la préparation d'une chaîne de transmission. Ce choix dépend principalement de plusieurs paramètres cruciaux plus particulièrement la taille de la famille de séquences qu'on peut générer, les performances en termes de FAC et de FIC et les longueurs des séquences générées.

Concernant le premier paramètre, dans les systèmes à base de CDMA, le nombre de séquences disponibles doit être suffisamment grand pour autoriser l'accès au canal de transmission à un maximum d'usagers. Pour le $2^{ème}$ paramètre, indiquant les caractéristiques énergétiques, une valeur faible de la FIC est très importante pour permettre d'éliminer les IAM dues à la transmission simultanée de plusieurs utilisateurs. De plus, une bonne propriété de la FAC est importante pour assurer une meilleure synchronisation des séquences en réception et éliminer par conséquent les interférences dues aux trajets multiples. Quant au troisième paramètre, relatif à la sécurité des communications, des séquences de codes assez longues que possible donne des performances supérieures.

Une recherche approfondie des techniques de génération des séquences PN avec les propriétés souhaitées nous a permis de proposer une méthode efficace permettant de générer deux types de séquences longues ayant de bonnes propriétés de la FAC. En effet, deux nouvelles classes d'OLBSS ont été construites à partir d'un certain nombre de sous-séquences binaires plus courtes de même longueur appartenant à la même famille. Dans la première classe, les sous-séquences ont été concaténées, tandis que dans la seconde elles ont été entrelacées. Deux algorithmes différents ont été utilisés, à savoir les AGs et la méthode OEP, pour sélectionner la position optimale de chaque sous-séquence dans la première classe et celle du premier bit de chaque sous-séquence dans la seconde classe.

Selon les résultats obtenus, les nouvelles OLBSS, composées des sous-séquences de Walsh-Hadamard ou sous-séquences de Gold présentent de bonnes propriétés de la FAC par rapport celles aléatoires et de Gold originales de mêmes longueurs. De plus, l'étude comparative entre les deux classes a montré que les OLBSS composées des sous-séquences de Walsh-Hadamard entrelacées présentent les meilleures performances. En ce qui concerne les méthodes d'optimisation, les résultats obtenus ont démontré que les AGs sont meilleurs pour la sélection des positions des sous-séquences ou des premiers bits des sous-séquences dans les séquences longues optimisées. En outre, les séquences les plus efficaces, parmi les meilleures OLBSS obtenues, sont celles construites à partir d'un plus grand nombre de sous-séquences. Subséquemment, un choix judicieux de la longueur de la sous-séquence doit être considéré comme étant le paramètre le plus intéressant pouvant influencer la qualité des OLBSS.

Finalement, une étude comparative multi-échelles a démontré que les séquences OLBSS construites par les sous-séquences de Walsh-Hadamard ou de Gold et optimisées par les AGs, quelles que soient leurs tailles, présentent les meilleures performances en comparaison avec celles des celles aléatoires, de Gold ou de Weil de mêmes tailles. Ceci prouve la validité de notre méthode proposée pour générer des séquences optimisées avec n'importe quelle taille.

# Perspective

Comme perspective, nous proposons aux futurs Doctorants de faire des recherches et des études avancées afin de proposer des méthodes permettant de générer un nombre assez grand des séquences de même famille optimisées par rapport à d'autres propriétés incluant le paramètre de FIC pour l'utilisation dans les systèmes de communications.

# Références Bibliographiques

[1] Sourour E, Atarius R, Khayrallah A. CDMA system using Quasi-orthogonal codes. U.S. Patent 7,133,353, 2006.

[2] Viterbi A J. CDMA Principles of Spread Spectrum. Addison-Wesley, 1995.

[3] Pickholtz R, Schilling D, Milstein L. Theory of Spread-Spectrum Communications - A Tutorial. Navigation. IEEE Transactions on Communications, 1982, 30(5): 855-884.

[4] Zepeng Z, Jinfeng C, Lei Y, Zhiyao Y. Correlation Functions of Perfect Binary Sequences of Same Lengths. Wuhan University Journal of Natural Sciences, 2019, 24 (3): 201-204

[5] Kedia D. Comparative Analysis of Peak Correlation Characteristics of Non-Orthogonal Spreading Codes for Wireless Systems. International Journal of Distributed and Parallel Systems (IJDPS), 2012, 3(3): 63-74.

[6] Sarwate D V, Pursley M B. Crossorrelation Properties of Pseudo Random and Related Sequences. Proceedings of the IEEE, 1980, 68(5): 593-619.

[7] Golomb S W. Shift Register Sequences. San Francisco: Holden-Day, 1967.

[8] Gold R. Optimal Binary Sequences for Spread Spectrum Multiplexing. IEEE Transactions on Information Theory, 1967, 13(4): 619-621.

[9] Kasami T. Weight Distribution Formula for Some Class of Cyclic Codes. Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1966.

[10] Rushanan J J. Weil Sequences: A Family of Binary Sequences with Good Correlation Properties IEEE International Symposium on Information Theory, Seattle, WA, USA, 2006.

[11] Golomb S, Scholtz R. Generalized Barker Sequences. IEEE Transactions on Information Theory, 1965, 11(4): 533-537.

[12] Fan P Z, Suehiro N, Kuroyanagi N, Deng X M. Class of Binary Sequences with Zero Correlation Zone. Electronics Letters, 1999, 35(10): 777-779.

[13] De Bruijn N G. A combinatorial problem. Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam, 1946, 49(2): 758-764.

[14] Golomb S. W. Shift Register Sequences: Secure and Limited-Access Code Generators, Efficiency Code Generators, Prescribed Property Generators, Mathematical Model. (Third Revised Edition) World Scientific, 2017.

[15] Rouabah K, Attia S, Harba R, Ravier P, Boukerma S. Optimized Method for Generating and Acquiring GPS Gold Codes. International Journal of Antennas and Propagation, 2015.

[16] Wallner S. Avila-Rodriguez J-A, Hein G W, Galileo E1 OS and GPS L1C Pseudo Random Noise.

Codes - Requirements, Generation, Optimization and Comparison. Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS), 2007.

[17] Dinan E. H, Jabbari B. Spreading Codes for Direct Sequence CDMA and Wideband CDMA Cellular Networks." IEEE Communications Magazine, 1998, 36(9): 48-54.

[18] Kedia D, Duhan M, Maskara S L. Evaluation of Correlation Properties of Orthogonal Spreading Codes for CDMA Wireless Mobile Communication. IEEE 2nd International Advance Computing Conference (IACC), 2010, Patiala, India.

[19] Adachi F, Sawahashi M, Okawa K. Tree-Structured Generation of Orthogonal Spreading Codes with Different Lengths for Forward Link Of DS-CDMA Mobile Radio. Electronics Letters, 1997, 33(1): 27 – 28.

[20] Golay M J E. Complementary series. IRE Transactions on Information Theory, 1961, 7(2): 82-87.

[21] ETSI. The ETSI UMTS Terrestrial Radio Access (UTRA) ITU-R R0TT Candidate Submission, 1998.

[22] Association of Radio Industries and Businesses (ARIB). Japan's Proposal for Candidate Radio Transmission Technology on IMT-2000: W-CDMA, 1998.

[23] CATT. TD-SCDMA Radio Transmission Technology for IMT-2000 Candidate submission, 1998.

[24] TIA/EIA-95B. Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Systems. Baseline Version, 1997.

[25] Telecommunications Industry Association (TIA). The CDMA2000 ITU-R RTT Candidate Submission, 1998.

[26] Olsen J, Scholtz R, Welch L. Bent-Function Sequences. IEEE Transactions on Information Theory, 1982, 28(6): 858-864.

[27] No J-S, Kumar P V. A New Family of Binary Pseudorandom Sequences Having Optimal Periodic Correlation Properties and Large Linear Span. IEEE Transactions on Information Theory, 1989, 35 (2): 371-379

[28] Scholtz R, Welch L.GMW Sequences. IEEE Transactions on Information Theory, May 1984, 30(3): 548-553.

[29] Frank R L, Zadoff S A. Phase Shift Pulse Codes With Good Periodic Correlation Properties. IRE Transactions on Information Theory, 1962, 381-382.

[30] Holland J H. Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The University of Michigan Press, 1975.

[31] Goldberg D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, 1989.

[32] Kennedy J, Eberhart R. Particle Swarm Optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, Australia, 1995.

[33] Eberhart, Shi Y. Particle Swarm Optimization: Developments, Applications and Resources.

Proceedings of the Congress on Evolutionary Computation; Seoul, South Korea, South Korea, 2001.

[34] Schltz R.A. The origins of spread spectrum communications. IEEE Transactions on Communications, 1982, ( COM-30): 822-854.

[35] Someswar G M, Rao T P S C, Chigurukota D R. Global Navigation Satellite Systems and Their Applications. International Journal of Software and Web Sciences (IJSWS), 2013, 3(1): 17-23.

[36] Hofmann-Wellenhof B, Lichtenegger H, Wasle E. GNSS Global Navigation Satellite Systems; GPS, Glonass, Galileo & More. Springer-Verlag Wien, 2007.

[37] Shannon C E. A Mathematical Theory of Communication. Bell System Technical Journal, 1948: 379-423, 623-656

[38] Dixon R C. Spread Spectrum Systems. Second édition, 1986.

[39] Glisic S, B. Vucetic. Spread spectrum CDMA systems for wireless communications. Artech House, 1997.

[40] Nobilet S. Etude et optimisation des techniques MC-CDMA pour les futures générations de systèmes de communications hertziennes. PhD thesis, Institut National des Sciences Appliquées, Rennes, octobre 2003.

[41] Chen H H. The Next Generation CDMA Technologies. England: John Wiley & Sons, Ltd. 2007.

[42] Simon M K, Omura J K, Scholtz R A, Levitt B K. Spread Spectrum Communications Handbook. McGraw-Hill International Editions, 2002.

[43] Karim M R., Sarraf M. W-CDMA and CDMA2000 for 3G Mobile Net-works. McGrauw-Hill Telecom Professional, 2002.

[44] Garg V K. IS-95 CDMA and CDMA2000. Cellular/PCS Systems Implementation. Prentice Hall, 2000.

[45] Gold R. Maximal Recursive Sequences with 3-valued Recursive Cross-correlation Functions. IEEE Transactions on Information Theory, 1968:154-156.

[46] Fan P, Darnell M. Sequence Design for Communication Applications. U.K.:R esearch Studies Press,1996.

[47] Donelan H, Farrell T. O. Methods for Generating Sets of Orthogonal Sequences. Electronics Letters, 1999, 35 (18):1537–1538.

[48] Tachikawa S I. Recent Spreading Codes for Spread Spectrum Communication Systems. Electronics and Communications in Japan, 1992, 75(6): 41-49.

[49] Tseng C, Liu C. Complementary Sets of Sequences. IEEE Transactions on Information Theory, 1972, IT –18(5): 644–652.

[50] Ding C, Hesseseth T, Shan W. On the Linear Complexity of Legendre Sequences. IEEE Transactions on Information Theory, 1998, 44(3): 1276-1278.

[51] Lu H, Niu R. Generation Method of GPS L1C Codes Based on Quadratic Reciprocity law. Journal of Systems Engineering and Electronics, 2013, 24(2): 189-195.

[52] Fan P, Suehiro N, Kuroyanagi N, Deng X M. Class of Binary Sequences with Zero Correlation Zone.

Electronics Letters, 1999, 32(10): 777-779.

[53] Maeda T, Kanemoto S, Hayashi T. A Novel Class of Binary Zero-Correlation Zone Sequence Sets. IEEE Region 10 Conference, 2010: 708-711.

[54] Hayashi T. A Class of Ternary Sequence Sets having a Zero-Correlation Zone for Even and Odd Correlation Functions. IEEE International Symposium on Information Theory, Yokohama, Japan, 2003.

[55] Golomb S W. Shift Register Sequences. Aegean Park Press, 1982.

[56] Sawada J, Williams A, Wong D. A surprisingly simple de Bruijn sequence construction. Discrete Math. 2016, 339: 127–131.

[57] Fredricksen H, Maiorana J, Necklaces of beads in k colors and k-ary de Bruijn sequences, Discrete Math, 1978,23:207–210.

[58] Ford L. A cyclic arrangement of M-tuples, Report No. P-1071, Rand Corporation, Santa Monica, California, 1957.

[59] Martin M H., A problem in arrangements, Bull. Amer. Math. Soc, 1934,40 (12): 859–864

[60] Fredricksen H. A survey of full length nonlinear shift registers cycle algorithms. SIAM review, 1982, 24 (2):195–221.

[61] Alhakim A. A Simple Combinatorial Algorithm for De Bruijn Sequences, Amer. Math. Monthly, 2010,117 (8):728–732.

[62] Golay M J E. A Class of Finite Binary Sequences with Alternate Autocorrelation Values Equal to Zero. IEEE Transactions on Information Theory, 1972, IT-18(449):450.

[63] Land A H, Doig A G. An automatic method of solving discrete programming problems. Econometrica: Journal of the Econometric Society, 1960, 28(3): 497–520.

[64] Bellman R: Dynamic Programming. Princeton University Press, 1957.

[65] Baptiste P, Le Pape C, Nuijten W. Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems, Springer, 2001.

[66] Kantorovich L V. Mathematical Methods of Organizing and Planning Production. Management Science, 196, 06 (4):366–422.

[67] Glover F. Future Paths for Integer Programming and Links to Artifocial Intelligence. Computers and Operations Research, 1986, 13: 533-549.

[68] Osman I H, Laporte G. Metaheuristics : A bibliography. Annals of Operations Research, 1996, 63:513–623.

[69] Kirkpatrick S, Gelatt S, Vecchi M. Optimization by Simualted Annealing, Science, 1983, 220(4598): 671-680.

[70] Cerny V. Thermodynamical Approach to the Traveling Saleman Problem: An Efficient Simulation Algorithm. Journal of Optimization Theory and Applications, 1985, 45:41–51.

[71] Bäck T. Evolutionary Algorithms in Theory and Practice. Oxford University Press, 1995

[72] Darwin C. The Origin of Species by Means of Natural Selection. Mentor Reprint, New York, 1859.

[73] Koza JR. Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems. Stanford University Computer Science Department STAN-CS-90-1314, 1990.

[74] Rechenberg I. Evolution Strategy: Optimization of Technical Systems by Means of Biological Evolution. Fromman-Holzboog, 1973, 104:15–16.

[75] Fogel L J, Owens A J, Walsh M J. Articial Intelligence through Simulated Evolution.John Wiley & sons, 1966.

[76] Angeline P. Comparing subtree crossover with macromutation. International Conference on Evolutionary Programming, Evolutionary Programming VI, Berlin. Springer, 1997:101-111.

[77] Fogel D B, Fogel L J. An introduction to evolutionary programming. European Conference on Artificial Evolution, 1995:21-33.

[78] Dorigo M, Maniezzo V, Colorni A. Distributed Optimization by Ant Colonies", Proceedings of ECA L91 –European Conference on Artificia l Life, Elsevier Publishing, 1991: 134-142.

[79] Moscato P, Norman M. A competition- cooperative approach to complex combinatorial search. Technical Report 790, Caltech Concurrent Computational Program, 1989.

[80] Lindner J. Binary Sequences up to Length 40 with Best Possible Autocorrelation Function. Electronics Letters, 1975, 11: 507.

[81] Baden J, Cohen M. Optimal Peak Sidelobe Filters for Biphase Pulse Compression. In proceedings of IEEE conference Radar, 1990: 249–252.

[82] Coxson G, Russo J. Efficient Exhaustive Search for Optimal-Peak-Sidelobe Binary Codes. IEEE transactions on aerospace and electronic systems, Jan. 2005, 41(1): 302–308.

[83] Leukhin A, Potehin E. Binary Sequences with Minimum Peak Side-Lobe Level up to Length 68. 2012, arXiv: 1212. 4930.

[84] Leukhin A , Potekhin E N. Optimal Peak Sidelobe Level Sequences up to Length 74. In proceedings of IEEE conference Radar, 2013: 495–498.

[85] Leukhin A, Potekhin E N. Exhaustive Search For Optimal Minimum Peak Sidelobe Binary Sequences up to Length 80. In proceedings of IEEE conference. Sequences Appl, 2014: 157–169.

[86] Leukhin A, Potekhin E N. A Bernasconi model for constructing ground-state spin systems and optimal binary sequences. Journal of physics conference series, 2015, 613(1).

[87] Leukhin A N, Parsaev N V, Bezrodnyi V I, Kokovihina N A. The Exhaustive Search for Optimum Minimum Peak Sidelobe Binary Sequences. Bulletin of the Russian Academy of Sciences Physics, 2017, 81(5): 575-578.

[88] Nunn C J, Coxson G E. Best-Known Autocorrelation Peak Sidelobe Levels for Binary Codes of Length 71 to 105. IEEE Transactions on Aerospace and Electronic Systems, 2008, 44 (1): 392–395.

[89] Dimitrov M, Baitcheva T, Nikolov N. Efficient Generation of Low Autocorrelation Binary Sequences.

IEEE Signal Processing Letters, 2020, 27: 341-345.

[90] Bose A, Soltanalian M. Constructing Binary Sequences With Good Correlation Properties: An Efficient Analytical-Computational Interplay. IEEE Transactions on Signal Processing, 2018, 66 (11): 2998-3007.

[91] Wai Ho M; Ke-Lin D; Wei Hsiang W. New evolutionary search for long low autocorrelation binary sequences. IEEE Transactions on Aerospace and Electronic Systems, 2015, 51(1): 290 – 303.

[92] Mertens S. Exhaustive Search for Low-Autocorrelation Binary Sequences. Journal of Physics A: mathematical and General, 1996, 29(18): 473–481.

[93] Wang S. Efficient heuristic method of search for binary sequences with good aperiodic autocorrelations. Electronics Letters, 2008, 44 (12):731–732.

[94] Zhang N, Golomb SW. Polyphase sequence with low autocorrelations. IEEE Transactions on Information Theory, 1993, 39(3):1085–1089.

[95] Li J, Stoica P, Zheng X. Signal synthesis and receiver design for MIMO radar imaging. IEEE Trans.Signal Process. 2008, 56:3959–3968.

[96] Stoica P, He H, Li J. New Algorithms for Designing Unimodular Sequences with Good Correlation Properties. IEEE Transactions on Signal Processing, 2009, 57(4): 1415-1425.

[97] Stoica P, He H, Li J. On designing sequences with impulse-like periodic correlation. IEEE Signal Processing Letters, 2009, 16(8): 703–706.

[98] Bose A, Mohammadi N, Soltanalian M. Designing Signals With Good Correlation and Distribution Properties. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.

[99] Soltanalian M, Naghsh M M, Stoica P. A Fast Algorithm for Designing Complementary Sets of Sequences. Signal Process, 2013, 93: 2096-2102.

[100] Lin R, Soltanalian M, Tang B, Li J. Efficient Design of Binary Sequences with Low Autocorrelation Sidelobes. IEEE Transactions on Signal Processing, 2019, 67(24): 6397- 6410.

[101] Soltanalian M, Stoica P. Computational design of sequences with good correlation properties. IEEE Transactions on Signal processing, 2012, 60 (5):2180–2193.

[102] Zhao L, Song J, Babu P, Palomar D P. A Unified Framework for Low Autocorrelation Sequence Design via Majorization-Minimization. IEEE Transactions on Signal, 2017, 65(2): 438–453.

[103] Alaee-Kerahroodi M, Modarres-Hashemi M, Naghsh M M. Designing Sets of Binary Sequences for MIMO Radar Systems. IEEE Transactions on Signal Processing, 2019, 67(13):3347-3360.

[104] Lunelli L. Tabelli di sequenze (+1,−1) con autocorrelazione troncata non mag-giore di 2. Politecnico di Milano, 1965.

[105] Littlewood J E. Some Problems in Real and Complex Analysis. Heath Mathematical Monographs. D.C. Heath and Company, Massachusetts, 1968.

[106] Golay M J E. The merit factor of long low autocorrelation binary sequences. IEEE Trans. Inform.

Theory, 1982, IT-28:543–549.

[107] Mertens S, Bauke H. Ground States of the Bernasconi Model with Open Boundary Conditions. Online. Available: <http://odysseus.nat.uni-magdeburg.de/~mertens/bernasconi/open.dat>, November 2004.

[108] Packebusch T, Mertens S. Low Autocorrelation Binary Sequences. Journal of Physics A Mathematical and Theoretical, 2016, 49(16).

[109] Golay M J E. The merit factor of legendre sequences. IEEE Trans. Inform. Theory, 1983, IT-29:934-936.

[110] Hoholdt T, Jensen H E. Determination of the merit factor of Legendre sequences. IEEE Trans. Inform. Theory, 1988, 34:161-164.

[111] Kristiansen R A, Parker M G. Binary Sequences With merit factor ≥ 6.3. IEEE Trans Theory, 2004, 50(12):3385-3389.

[112] Green D H, Green P R. Modified Jacobi sequences. IEE Proceedings. Computers and Digital Techniques, 2000, 147(4): 241–251

[113] Suribabu Naick B, Rajesh Kumar P. An Algorithmic Approach towards Construction of Long Binary Sequences using Modified Jacobi Sequences. International Journal of Computer Applications, 2015, 120(1) : 8-15.

[114] Ding C, Helleseth T. On Cyclotomic Generator of Order R. Inf. Process Lett, 1998, 66 (1): 21–25.

[115] Ukil A. Low Autocorrelation Binary Sequences: Number Theory-Based Analysis for Minimum Energy Level, Barker codes, Digit. Signal Process. 20 (2) (2010) 483-495.

[116] Ukil A. On Asymptotic Merit Factor of Low Autocorrelation Binary Sequences. In Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE, 2015: 004738–004741.

[117] Meng R, Yan T. New Constructions of Binary Interleaved Sequences with Low Autocorrelation. International Journal of Network Security, 2017, 19(4): 546-550.

[118] Yan T, Gong G. Some Notes on Constructions of Binary Sequences with Optimal Autocorrelation. 2014.

[119] Ren J. Design of Long Period Pseudo-Random Sequences from the Addition of M -Sequences over Fp. EURASIP Journal on Wireless Communications and Networking, 2004, 1: 12-18.

[120] Sarayloo M, Gambi E, Spinsante S. A New Approach to Sequence Construction with Good Correlation by Particle Swarm Optimization. Journal of Communications Software and Systems, 2015, 11(3): 127-135.

[121] Clerc M, Kennedy J. The Particle Swarm: Explosion, Stability, and Convergence in Multi-Dimensional Complex Space. IEEE Transactions on Evolutionary Computation, 2002, 6: 58-73.

[122] Eberhart R C, Shi Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. Proceedings of the 6thIEEE Congress on Evolutionary Computation, 2000: 84-88, IEEE Press.

[123] Boukerma S, Rouabeh K, Mezaache S , Atia S.Efficient method for constructing optimized long binary

spreading sequences.International Journal of Communication Systems - Wiley;34:e4719, 2021. https://doi.org/10.1002/dac.4719

# Annexe A

Le Tableau 1 représente les connexions de rétroaction correspondant aux différentes longueurs des codes PN à longueur maximale.

**Tableau 1 Connexions de rétroaction correspondant aux différentes longueurs des codes PN à longueur maximale**

| Taille du LFSR N | Longueur du code $N = 2^n - 1$ | Les connexions de rétroaction de m-séquence | La taille de la famille du code |
|---|---|---|---|
| 2 | 3 | [2,1] | 1 |
| 3 | 7 | [3,1] | 2 |
| 4 | 15 | [4,1] | 2 |
| 5 | 31 | [5, 2], [5, 4, 3, 2], [5, 4, 2, 1] | 6 |
| 6 | 63 | [6, 1], [6, 5, 2, 1] [6, 5, 3, 2] | 6 |
| 7 | 127 | [7, 3], [7, 3, 2, 1]<br>[7, 3, 2, 1], [7, 5, 4, 3, 2,1] | 18 |
| 8 | 255 | [8, 7, 6, 5, 2, 1],[8, 7, 6, 1] | 16 |
| 9 | 511 | [9, 4], [9, 6, 4, 3]<br>[9, 6, 4, 3], [9, 8, 4, 1] | 48 |
| 10 | 1023 | [10, 9, 8, 7, 6, 5, 4, 3], [10, 9, 7, 6,<br>4,1] ,[10,8,5,1],[10,7, 6, 4, 2, 1]<br>[10, 9, 8, 7, 6, 5, 4, 3,1], [10, 9 7, 6, 4,1] | 176 |

Le Tableau 2 montre les paires préférées de m-séquences nécessaires pour générer les codes de Gold.

**Tableau 2 Paires de m-séquences préférées utilisées pour générer les codes de Gold et leurs valeurs de FAC et de FIC correspondantes**

| Taille du registre LFSR n | Longueur du code $N = 2^n - 1$ | Paires préférées des m-séquences |
|---|---|---|
| 5 | 31 | [5, 2], [5, 4, 3, 2] |
| 6 | 63 | [6, 1], [6, 5, 2, 1] |
| 7 | 127 | [7, 3], [7, 3, 2, 1]<br>[7, 3, 2, 1], [7, 5, 4, 3, 2,1] |
| 8 | 255 | [8, 7, 6, 5, 2, 1],[8, 7, 6, 1] |
| 9 | 511 | [9, 4], [9, 6, 4, 3]<br>[9, 6, 4, 3], [9, 8, 4, 1] |
| 10 | 1023 | [10, 9, 8, 7, 6, 5, 4, 3], [10, 9, 7, 6, 4,1]<br>[10,8,5,1],[10,7, 6, 4, 2, 1]<br>[10, 9, 8, 7, 6, 5, 4, 3,1], [10, 9 7, 6, 4,1] |

# Annexe B

**Construction des Séquences ZCZ binaires**

- **1$^{ère}$ méthode - Construction de Fan et al :**

On obtient un ensemble de séquences ZCZ $F^{(1)}$ de ZCZ (8, 4, 1) à partir de la matrice de Hadamard, MOCS basique $F^{(0)}$. La matrice de démarrage MOCS est choisie comme suit :

$$F^{(0)} = \begin{bmatrix} + & + \\ + & - \end{bmatrix}$$

Pour la première itération (n = 1), en utilisant l'Equation 1-27, un ensemble de séquences ZCZ $F^{(1)}$ est obtenu comme suit :

$$F^{(1)} = \begin{bmatrix} + & + & + & + & - & + & - & + \\ + & + & - & - & - & + & + & - \\ - & + & - & + & + & + & + & + \\ - & + & + & - & + & + & - & - \end{bmatrix}$$

Les lignes de $F^{(1)}$ sont des séquences MOCS. Pour une matrice ZCZ plus grande notamment ZCZ (32, 8, 2), on applique l'Equation 1-27 sur $F^{(1)}$.

- **2$^{ème}$ méthode - Construction de Maeda et al :**

L'exemple suivant détaille la génération d'une famille des séquences ZCZ de taille $K = 4$. Ici, la longueur de chaque séquence est $N = 16$. De plus, $Z_0$ = 1. Dans ce cas, la séquence est nommée ZCZ (16, 4, 1).

Pour se faire, on démarre avec une matrice de Hadamard d'ordre $N = 2$ donnée ci-dessous :

$$H_2 = \begin{bmatrix} + & + \\ + & - \end{bmatrix}$$

La matrice de démarrage ZCZ est obtenue à l'aide de l'Equation 1-28 comme suit :

$$F = \begin{bmatrix} F^1 \\ F^2 \end{bmatrix} = \begin{bmatrix} - & - & + & + & + & + & + & + \\ - & + & + & + & - & + & + & - \\ + & + & + & - & - & - & + & + \\ + & - & + & - & + & + & + & - \end{bmatrix}$$

Et l'ensemble des séquences ZCZ est obtenu en utilisant l'Equation 1-29 comme suit :

$$Z^{(0)} = \begin{bmatrix} - & + & - & + & + & + & + & + & + & - & + & - & + & + & + \\ - & + & + & - & + & + & - & - & + & - & - & + & + & + & - \\ - & - & - & - & + & - & + & - & + & + & + & + & + & - & + \\ - & - & + & + & + & - & - & + & + & + & - & - & + & - & - \end{bmatrix}$$

Les lignes de $Z^{(0)}$ sont des séquences ZCZ. Si une matrice ZCZ plus grande $Z^{(1)}$ nommée ZCZ (32, 4, 3) est nécessaire, on applique l'Equation 1-30 sur la matrice $Z^{(0)}$

**Séquences ZCZ ternaires**

- **1$^{ère}$ méthode - construction de Hayashi's :**

Génération d'un ensemble de séquences de taille $K = 8$, de longueur $N = 20$ chacune et de zone zéro $Z_0 = 1$.

On démare d'une matrice de Hadamard d'ordre $N = 4$ donnée comme suit :

$$H_4 = \begin{bmatrix} + & + & + & + \\ + & - & + & - \\ + & + & - & - \\ + & - & - & + \end{bmatrix}$$

L'ensemble $F$ est obtenu à l'aide de l'Equation 1-37 comme suit :

$$F = \begin{bmatrix} F^1 \\ F^2 \end{bmatrix} = \begin{bmatrix} + & + & + & +0 & + & + & 0 \\ + & - & + & -0 & + & - & 0 \\ + & + & - & -0 & + & + & 0 \\ + & - & - & +0 & + & - & 0 \\ + & + & + & +0 & - & - & 0 \\ + & - & + & -0 & - & + & 0 \\ + & + & - & -0 & - & + & 0 \\ + & - & - & +0 & - & + & 0 \end{bmatrix}$$

En utilisant la technique d'entrelacement de l'Equation 1-38, une séquence ZCZ, nommée ZCZ (20, 8, 1) peut être obtenue. Elle est donnée comme suit :

$$Z^{(0)} = \begin{array}{l} + + + + + + + + 00 + - + - + - + -00 \\ + + - - + + - - 00 + - - + + - - +00 \\ + + + + - - - - 00 + - + - - + - +00 \\ + + - - - - + + 00 + - - + - + + -00 \\ + - + - + - + - 00 + + + + + + + +00 \\ + - - + + - - + 00 + + - - + + - -00 \\ + - + - - + - + 00 + + + + - - - -00 \\ + - - + - + + - 00 + + - - - - + +00 \end{array}$$

## Résumé

Les travaux de recherche présentés dans cette thèse ont pour but l'étude et l'optimisation des séquences longues utilisées dans les systèmes d'accès multiple. Initialement, nous réalisons une étude des différentes méthodes de génération des séquences PN utilisées dans les systèmes DS-SS. De plus, nous effectuons une comparaison des séquences d'étalement étudiées en s'appuyant plus particulièrement sur les propriétés de la corrélation. Ensuite, nous présentons le principe et les résultats de l'application de notre méthode proposée pour générer deux types de longues séquences d'étalement binaires optimisées OLBSS avec de bonnes propriétés de la fonction d'autocorrélation (FAC). Ici, le premier type proposé est construit à partir de sous-séquences binaires concaténées appartenant à la même famille de codes, telles que les sous-séquences de Walsh-Hadamard et les sous-séquences de Gold, ayants de bonnes propriétés de la fonction d'inter-corrélation (FIC). Le deuxième type utilise les mêmes sous-séquences mais qui sont plutôt entrelacées. Ici, le nombre et les tailles des sous-séquences sont liés à la taille choisie de la longue séquence finale construite et aux performances souhaitées. L'optimisation des OLBSS est réalisée à l'aide de deux techniques d'optimisation, à savoir les algorithmes génétiques (AGs) et l'optimisation par essaims de particules (OEP). Les résultats de simulations obtenus, basés sur l'outil Matlab, ont montré que les nouvelles OLBSS composées de sous-séquences de Walsh-Hadamard ou de Gold ont de bonnes propriétés de la FAC en comparaison avec les séquences aléatoires et celles de Gold originales de même longueur. De plus, l'étude comparative a montré que les OLBSS, obtenues en utilisant les AGs et les sous-séquences de Walsh-Hadamard entrelacées, présentent les meilleures performances.

**Mots clés :** PN, DS-SS, OLBSS, FAC, FIC, Walsh-Hadamard, Gold, AGs, OEP, Entrelacées, Concaténées.

## Abstract

This thesis focuses on the study and optimization of sequences used in CDMA systems. Firstly, we present a study of the methods of generating different pseudo-random PN sequences used in DS-SS systems. The basic properties including autocorrelation and cross-correlation of these sequences are compared and analyzed. In addition, we carry out a comparison of the studied spreading sequences by relying more particularly on the properties of the correlation. Next, we present the principle and the results of the application of our proposed method used to generate two types of OLBSS with good ACF properties. At this point, the first type proposed is constructed from the concatenated binary subsequences belonging to the same code family, such as Walsh-Hadamard subsequences and Gold subsequences, having good properties of the CCF. The second type uses the same sub-sequences but which are rather interlaced. Now, the number and the sizes of the subsequences are related to the preferred size of the final long constructed sequence and the desired performance. The optimization of theses OLBSS sequences is achieved using two optimization techniques, namely GA and PSO. The obtained simulation results, based on Matlab tool, showed that the new OLBSS, composed of Walsh-Hadamard or Gold subsequences, have good ACF properties in comparison with the original random and Gold sequences of same length. In addition, the comparative study demonstrated that the OLBSS, obtained using the GA optimization algorithm and interlaced Walsh-Hadamard subsequences, have the best performance.

**Key words:** PN code, DS-SS, OLBSS, ACF, CCF, Hadamard, Gold, GA, PSO, interlaced, concatenated.

**ملخص**

يهدف العمل البحثي المقدم في هذه الأطروحة إلى دراسة وتحسين التسلسلات الطويلة المستخدمة في أنظمة الوصول المتعددة. في البداية، نجري دراسة للطرق المختلفة لخلق تسلسل PN العشوائي الزائف المستخدمة في أنظمة DS-SS. بالإضافة إلى ذلك، قمنا بإجراء مقارنة بين تسلسلات الانتشار التي تمت دراستها من خلال الاعتماد بشكل خاص على خصائص الارتباط. بعد ذلك، نقدم مبدأ ونتائج تطبيق طريقتنا المقترحة لإنشاء نوعين من متواليات الانتشار الثنائي الطويلة المُحسَّنة OLBSS مع خصائص جيدة لوظيفة الارتباط التلقائي FAC. هنا، تم إنشاء النوع الأول المقترح من تكرارات ثنائية متسلسلة لاحقة تنتمي إلى نفس عائلة الكود، مثل سلاسل Walsh-Hadamard و Gold، والتي لها خصائص جيدة فيما يخص وظيفة الترابط FIC. النوع الثاني يستخدم نفس التسلسلات الفرعية ولكنها متشابكة إلى حد ما. هنا، يرتبط عدد وأطوال السلاسل اللاحقة بالطول المختار للتسلسل الطويل النهائي الذي تم إنشاؤه و النتائج المرجوة. يتم إجراء تحسين OLBSS باستخدام طريقتين، وهما الخوارزميات الجينية َAGs وتحسين سرب الجسيمات (OEP). أظهرت نتائج المحاكاة التي تم الحصول عليها ، بناءً على أداة Matlab ، أن سلاسل OLBSS الجديدة المكونة من Walsh-Hadamard أو Gold اللاحقة لها خصائص FAC جيدة مقارنة بالتسلسلات العشوائية و Gold الأصلية من نفس الطول. بالإضافة إلى ذلك ، أوضحت الدراسة المقارنة أن OLBSS ، التي تم الحصول عليها باستخدام خوارزمية تحسين َAGs ونتائج Walsh-Hadamard المتشابكة ، لديها أفضل أداء.

**الكلمات المفتاحية:** الرموز العشوائية الزائفة PN، أنظمة DS-SS (للتسلسل المباشر الطيف المنتشر)، تسلسل الانتشار الثنائي الطويل المُحسَّن (OLBSS)، دالة الارتباط الذاتية (FAC)، دالة الارتباط المتبادل (FIC)، التسلسلات الثنائية القصيرة Walsh-Hadamard أو Gold، الخوارزميات الجينية (AGs) ،خوارزمية عناصر السرب (OEP)، متسلسلة، متشابكة.