

Politechnika Poznańska
Wydział Elektroniki i Telekomunikacji
Katedra Radiokomunikacji

Autoreferat rozprawy doktorskiej

Multidimensional Modulations in Modern Digital Communication Systems

Modulacje wielowymiarowe we współczesnych
systemach telekomunikacyjnych

Tomasz G. Markiewicz

Promotor: prof. dr hab. inż. Krzysztof Wesółowski

Poznań, 2017

Wprowadzenie

Modulacje wielowymiarowe (ang. *multidimensional modulations*, *MDM*) nie są powszechnie znaną w telekomunikacji techniką transmisyjną. Najprostszym sposobem na zapoznanie się z koncepcją modulacji wielowymiarowych jest ich porównanie z kwadraturową modulacją amplitudy (ang. *quadrature amplitude modulation*, *QAM*). W przypadku techniki QAM symbole należące do konstelacji sygnałów są rozmieszczone w dwuwymiarowej przestrzeni, tj. na płaszczyźnie zespolonej. Natomiast w przypadku modulacji wielowymiarowych najważniejszą różnicą jest to, iż symbole wybierane są spośród punktów należących do tzw. kraty (ang. *lattice*) i tworzą w ten sposób konstelację wielowymiarową (ang. *multidimensional constellation*, *MDC*). Nadmienić należy, iż istnieją również inne podejścia do tematu modulacji wielowymiarowych *vide* [24], [5] czy [23], gdzie dwa dodatkowe wymiary uzyskuje się niejako poprzez wykorzystywanie połówek funkcji sinus oraz cosinus.

Fakt iż konstelacje wielowymiarowe nie są specjalnie znane nie oznacza bynajmniej, że nie znajdują one zastosowania w technice. Wręcz przeciwnie, dostrzec można, iż zainteresowanie konstelacjami wielowymiarowymi niezaprzeczalnie wzrosło w ciągu ostatnich lat. Szczególne zainteresowanie można zaobserwować ze strony technik światłowodowych, gdzie modulacje wielowymiarowe wydają się być dobrym rozwiązaniem [3, 4, 9–11]. Modulacje wielowymiarowe cieszą się jednakże nie mniejszym zainteresowaniem ze strony technik typowo związanych z radiokomunikacją. Flagowym przykładem może być tu technika SCMA (ang. *sparse code multiple access*) [18, 19, 25]. W pracy [25] utrzymuje się, że ta technika pozwala na zwiększenie efektywności widmowej trzykrotnie. Interesujące wyniki uzyskano również w przypadku stosowania modulacji wielowymiarowych w połączeniu z techniką OFDM (ang. *orthogonal frequency division multiplexing*) [8]. Choć w pracy [8] pokazano, iż wykorzystanie modulacji wielowymiarowych może dawać zyski rzędu 6 dB i 8 dB względem modulacji QAM, to po weryfikacji metodologii przez autora niniejszej rozprawy okazało się, iż zakładane tamże zyski są na *nieco* tylko niższym poziomie, tj. 1 dB do 3 dB.

1.1 Cel i teza pracy

Celem rozprawy jest pokazanie, iż **budowa oraz odwzorowywanie bitów w symbole danych w przypadku konstelacji wielowymiarowych wraz z detekcją symboli wielowymiarowych może zostać zrealizowane w lepszy i bardziej efektywny sposób od metod znanych obecnie**. W związku z tym aspekty poruszone w rozprawie dotyczą kolejno:

- budowania konstelacji wielowymiarowych,
- odwzorowywania ciągów bitów w punkty konstelacji, oraz
- detekcji symboli wielowymiarowych.

Odpowiedzią na problem budowania konstelacji wielowymiarowych jest opracowanie takich algorytmów, które zapewnią minimalizację średniej energii przypadającej na punkt konstelacji oraz wyeliminują składową stałą sygnału. Pierwszy z algorytmów rozważanych w rozprawie, tzw. algorytm sklejający (ang. *gluing algorithm*) [12], wykorzystuje fakt, iż pewien rodzaj konstelacji wielowymiarowych może zostać zbudowany poprzez "sklejanie" ze sobą wielowymiarowych sympleksów. Metoda jest niestety ograniczona do jednej rodziny konstelacji wielowymiarowych. Ograniczenia tego nie posiada drugi z proponowanych w niniejszej rozprawie algorytmów – algorytm wykorzystujący dystrybucję pierwiastków kraty (ang. *root-distribution algorithm, RD-algorithm*), który wybiera punkty konstelacji leżące na powierzchni wielowymiarowych sfer.

Odwzorowywanie ciągów bitów w punkty konstelacji (zwane czasem etykietowaniem – ang. *labelling*) jest stosunkowo proste w przypadku konstelacji określonych na płaszczyźnie, np. tych, których topologia jest zbliżona do konstelacji QAM. Ten sam problem nie jest już jednak tak trywialny w przypadku większej liczby wymiarów oraz konstelacji, które diametralnie różnią się w swojej strukturze od tego, co jest znane na płaszczyźnie (zespolonej). W tym celu zaproponowano w pracy algorytm etykietowania takich konstelacji – tzw. algorytm etykietujący warstwami (ang. *layer labelling algorithm, LL-algorithm*). Algorytm ten dzieli całą konstelację na tzw. warstwy (ang. *layers*) a następnie tak stworzone struktury wykorzystuje do rekurencyjnego przypisywania etykiet bitowych poszczególnym punktom konstelacji wielowymiarowej.

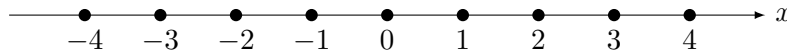
Ostatnim z algorytmów proponowanych w pracy jest algorytm służący do detekcji symboli wielowymiarowych. Proponowane rozwiązanie wykorzystuje podejście znane chociażby w informatyce pod nazwą "dziel i rządź" (łac. *divide et impera*). Algorytm ten, zwany dalej demodulatorem hipersześciennej (ang. *hipercube demodulator*), został porównany z rozwiązaniami znanymi z literatury, tj. dekoderym sferycznym (ang. *sphere decoder, SD*) oraz przeszukiwaniem wyczerpującym (ang. *exhaustive search*).

Kraty i konstelacje wielowymiarowe

Konstelacje wielowymiarowe wykorzystywane w pracy zostały zbudowane z wykorzystaniem krat. Choć sama idea wykorzystywania krat do budowy konstelacji jest w ogólności znana, to warto wyjaśnić dokładniej czym są kraty.

Najbardziej przystępna definicja kraty określa ją jako zbiór punktów w przestrzeni wielowymiarowej. Nie jest to jednak zbiór losowych punktów, ale samopowtarzalny i zachowujący pewną strukturę, nieskończony, lecz policzalny zbiór punktów z pewnej wielowymiarowej przestrzeni. Kraty oznaczamy symbolem X_n , gdzie X oznacza rodzinę (typ) kraty, natomiast indeks dolny, n , oznacza liczbę wymiarów przestrzeni, w której zdefiniowana jest dana krata.

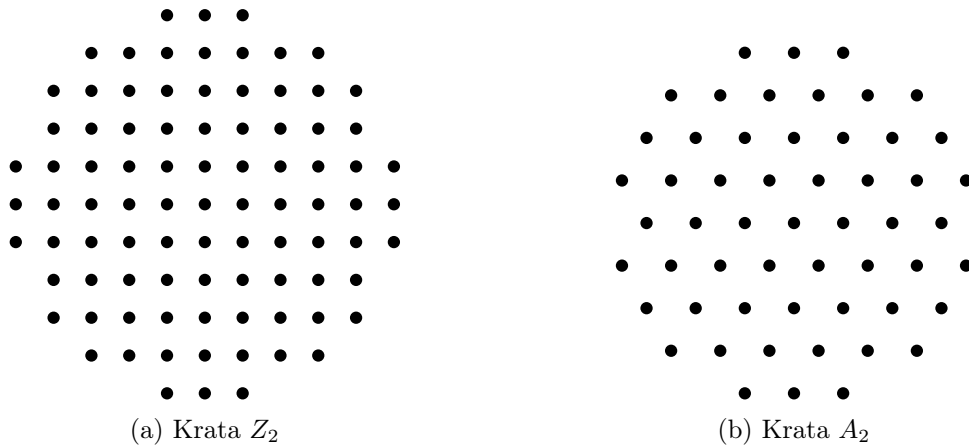
I tak najprostszym przykładem kraty może być krata Z_1 pokazana na rys. 2.1. Taka krata może być oczywiście uważana za podstawę do budowy konstelacji PAM (ang. *pulse amplitude modulation*). Krata ta to nic innego jak po prostu zbiór liczb całkowitych (oznaczany symbolem \mathbb{Z}). Alternatywnie można również zdefiniować kraty, które będą wielokrotnościami innych liczb, np. można określić kratę Z_1 , która będzie zawierała liczby ze zbioru $\sqrt{2}\mathbb{Z}$, tj. $\{0, \pm\sqrt{2}, \pm 2\sqrt{2}, \dots\}$.



Rysunek 2.1: Krata Z_1

Innym istotnym rodzajem kraty jest krata Z_2 , której fragment został pokazany na rys. 2.2a. Można zauważyć, że taka krata jest określona przez zbiór \mathbb{Z}^2 , a punkty z niej wybrane mogą służyć do budowy konstelacji QAM. Jeszcze innym rodzajem kraty jest krata A_2 ukazana na rys. 2.2b. Krata ta zawiera punkty leżące na wierzchołkach trójkątów równobocznych i jest ona wykorzystywana do budowy konstelacji TQAM (ang. *triangular quadrature amplitude modulation*) [15, 21]. Krata A_2 jest czasem również nazywana kratą sześciokątną (ang. *hexagonal lattice*).

Zobrazowanie krat w przypadku, gdy te są zdefiniowane w wyższych wymiarach, jest nieco trudniejsze i wymaga zapoznania z pojęciem diagramów *Dynkina*. W przypadku kraty A_2 diagram taki pokazano na rys. 2.3. Każdy z węzłów tego diagramu można w przybliżeniu uważać za związany z jednym z wektorów bazowych, na których rozpięta jest krata. W przypadku kraty A_2 mamy do czynienia z dwoma wektorami, które nachylone są do siebie pod kątem $\pi/3$ (por. rys. 2.2b). Fakt ten zobrazowany jest na rys. 2.3 w taki sposób, iż dwa węzły połączone są jedną krawędzią. W ogólności jeśli liczba krawędzi wynosi r , to kąt między wektorami bazowymi kraty wynosi $\pi/(2+r)$. Można z tego wywnioskować, iż diagram Dynkina dla kraty Z_2 będzie składał się jedynie z dwóch węzłów nie połączonych żadną krawędzią, ponieważ w przypadku kraty Z_2 kąt między wektorami bazowymi wynosi $\pi/2$, czyli $r = 0$, więc na takim diagramie nie będzie żadnej krawędzi.



Rysunek 2.2: Kraty A_2 oraz Z_2 .

Aby pokazać jak wygląda czterowymiarowa krata A_4 , spójrzmy na rys. 2.4. Można zauważyć, że rozpięta jest ona przez cztery wektory (cztery węzły na diagramie Dynkina kraty A_4). Dodatkowo możemy zauważyć, że kąt pomiędzy trzema parami wektorów wynosi $\pi/3$ (trzy węzły połączone są ze sobą za pomocą pojedynczej linii).



Rysunek 2.3: Diagram Dynkina kraty A_2 .



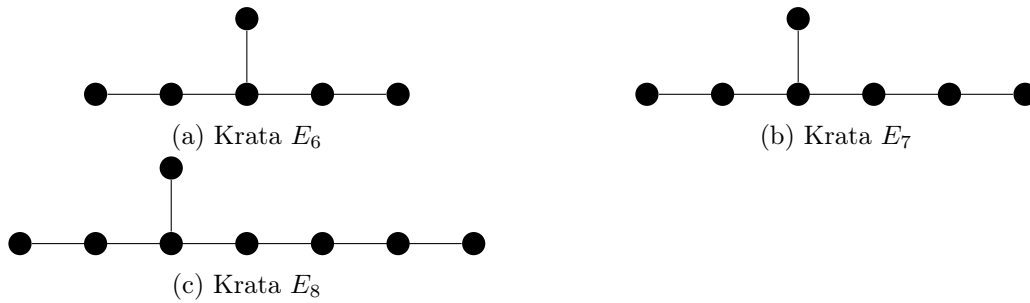
Rysunek 2.4: Diagram Dynkina kraty A_4 .

W rozprawie badano konstelacje tworzone na podstawie krat A_n , D_n , E_n oraz Z_n . Diagramy Dynkina dla krat A_n oraz D_n zostały pokazane na rys. 2.5. Kraty E_n (litera ‘E’ pochodzi od angielskiego słowa *exceptional* – niezwykły) z uwagi na swoją osobliwość określone są jedynie dla wymiarów 6, 7 oraz 8, w związku z czym diagramy Dynkina jedynie dla tych wymiarów zostały pokazane na rys. 2.6. Warto ponadto zauważyć, że niedawno opublikowano pracę, która udowadnia, iż krata E_8 określa najgęstsze upakowanie w ośmiu wymiarach [26]. Diagramy Dynkina dla krat Z_n składają się jedynie z n węzłów, dlatego też zostały pominięte.

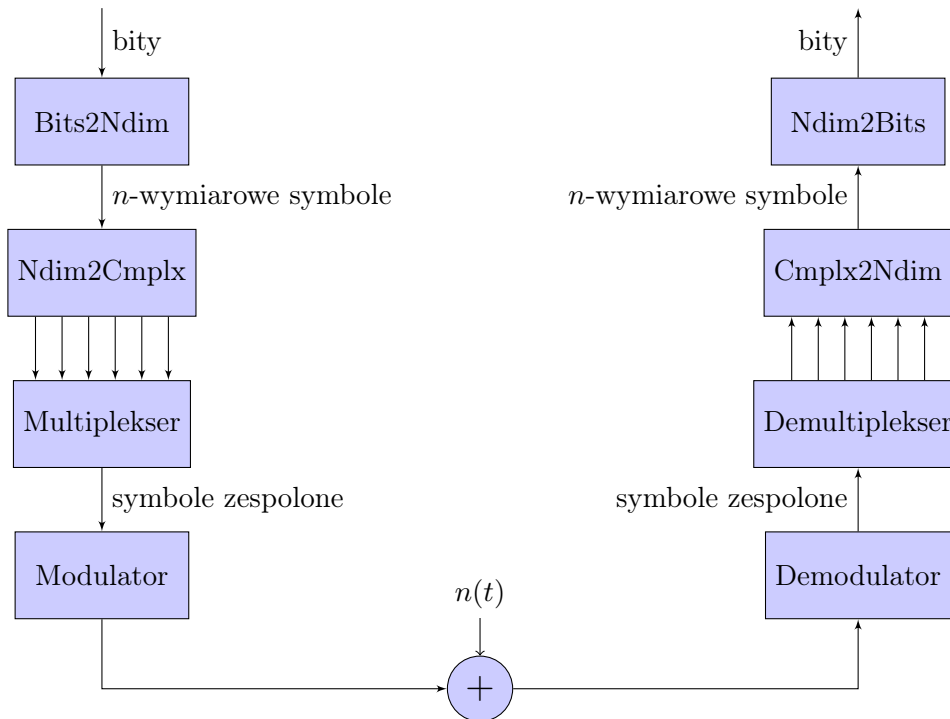
Na rys. 2.7 pokazano przykład systemu wykorzystującego modulacje wielowymiarowe. Elementem, który odróżnia taki system od systemu tradycyjnego jest nieco bardziej złożony sposób odwzorowywania bitów w symbole danych. W pierwszym etapie bity odwzorowywane są w symbole wielowymiarowe, tj. punkty w przestrzeni wielowymiarowej. Odpowiada za to element ‘Bits2Ndim’. Następnie każdy symbol wielowymiarowy zamieniany jest na odpowiednią liczbę symboli zespolonych (moduł ‘Ndim2Cmplx’). Przykładowo, czterowymiarowy punkt (a, b, c, d) może zostać zamieniony na dwa symbole zespolone $z_1 = a + jb$ oraz $z_2 = c + jd$. W przypadku nieparzystej liczby wymiarów można dokonać podobnego odwzorowania, przy czym w takim wypadku dwa symbole n wymiarowe zostają zamienione na n



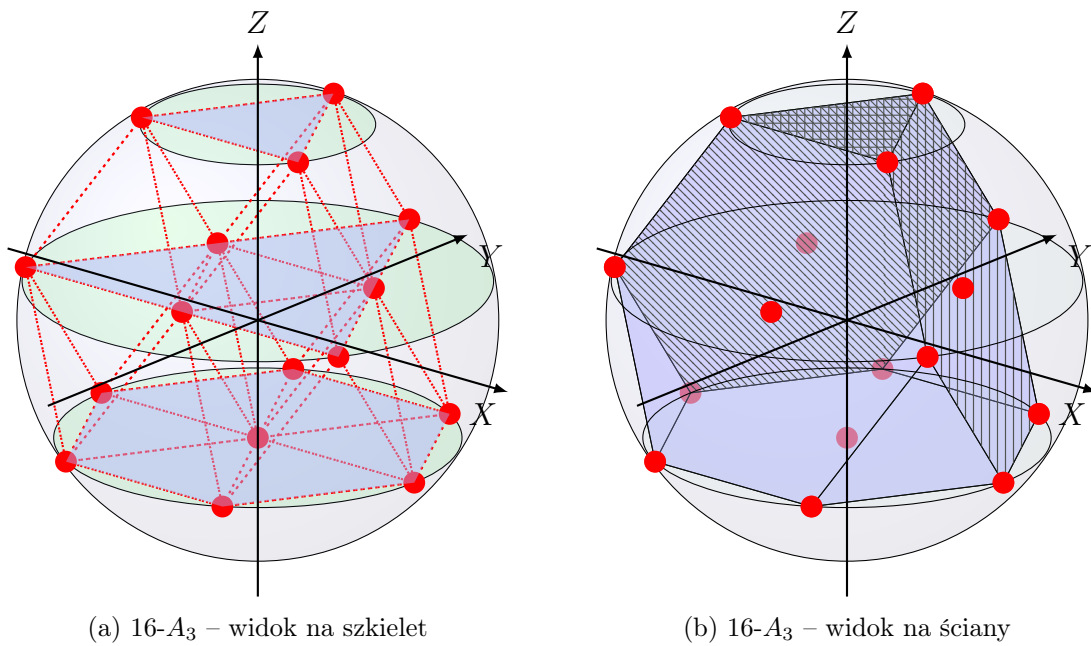
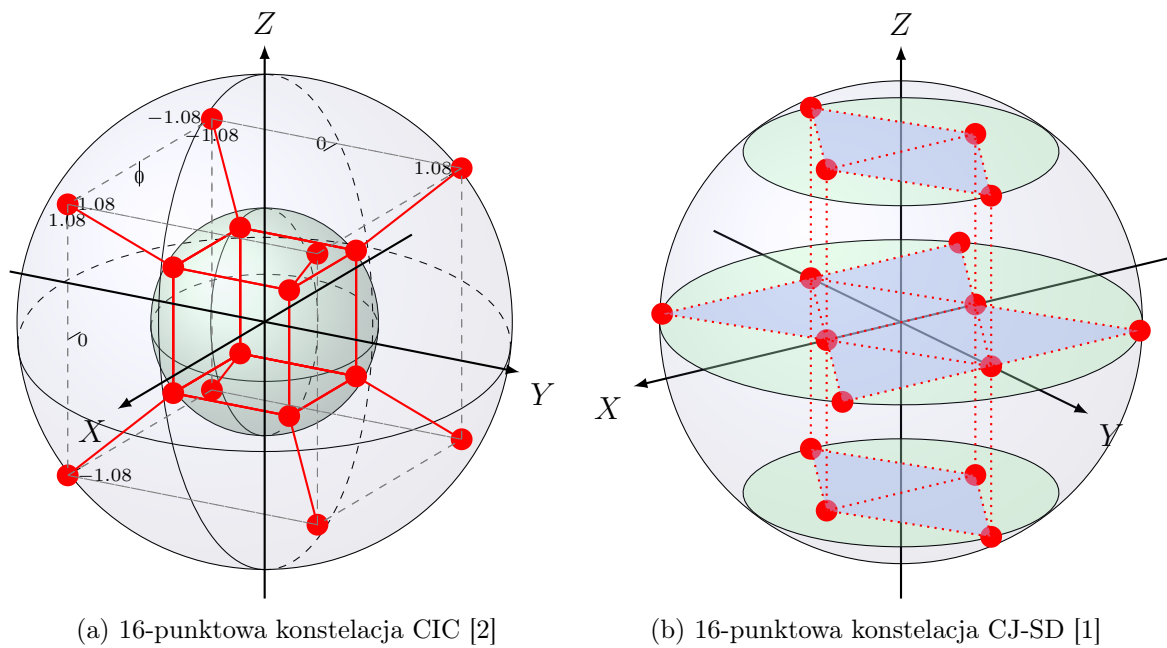
Rysunek 2.5: Diagramy Dynkina krat A_n oraz D_n .

Rysunek 2.6: Diagramy Dynkin dla krat E_6 , E_7 oraz E_8 .

symboli zespolonych. Kolejnym elementem takiego systemu jest multiplekser, który szereguje w czasie wszystkie symbole zespolone wychodzące z elementu 'Ndim2Cmplx'. Wreszcie powstałe w ten sposób symbole zespolone przekazywane są do modulatora, a następnie przesyłane przez kanał AWGN (ang. *additive white Gaussian noise*). Procedura odbiorcza jest realizowana w sposób analogiczny. Należy zauważyć, że pokazany przykład jest bardzo prosty, jednakże nic nie stoi na przeszkodzie, aby pokazany system przystosować do współpracy np. z techniką OFDM.

Rysunek 2.7: Prosty model systemu n -wymiarowego

Przykładowa konstelacja zbudowana za pomocą zaproponowanego w pracy algorytmu RD (ang. *root distribution*) jest pokazana na rys. 2.8. Na rys. 2.8a przedstawiono widok *szkieletu* konstelacji $16-A_3$, tj. 16-punktowej konstelacji zbudowanej na bazie kraty A_3 . Konstelacja ta składa się z trzech warstw, które kolejno zawierają 3, 6 oraz 7 punktów. Przerywane linie na rys. 2.8a opisują odległość minimalną pomiędzy punktami konstelacji. W tym wypadku wynosi ona 1. Na rys. 2.8b pokazano tę samą konstelację ($16-A_3$), lecz z widoku na wielościan. Konstelację tę można również rozumieć jako sumę zbioru punktów położonych na wierzchołkach czworoscianu ściętego (rys. 2.8b) oraz punktów leżących w centrum sześciokątnych ścian tejże bryły.

Rysunek 2.8: Konstelacja 16- A_3 zbudowana za pomocą algorytmu RD.

Rysunek 2.9: Konstelacje 16-CIC oraz 16-CJ-SD.

Konstelację tę porównano z konstelacjami zaproponowanymi w [2] oraz [1]. Obie konstelacje przedstawiono na rys. 2.9. Konstelacja 16-CIC (ang. *cube-in-cube* – sześcian w sześcianie) została zaproponowana przez Chena [2], natomiast konstelacja 16-CJ-SD (ang. *Chen-Jiang scheme D* – Chen-Jiang schemat D) jest to jedna z siedmiu konstelacji rozważanych w [1]. Konstelacja 16-CJ-SD charakteryzowała się najniższą średnią energią przypadającą na trójwymiarowy symbol spośród wszystkich konstelacji rozważanych w tej pracy.

Średnia energia przypadająca na symbol wielowymiarowy w przypadku konstelacji 16-CIC to

$$\mathcal{E}_{ave}^{CIC} \approx 2.11603, \quad (2.1)$$

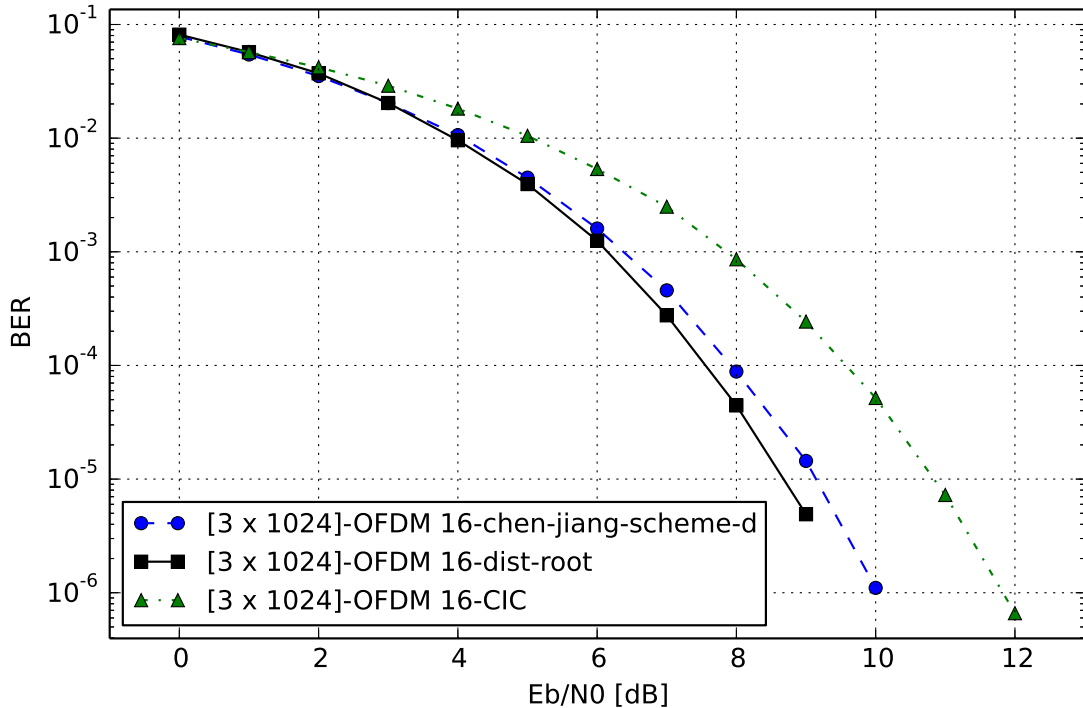
w przypadku konstelacji 16-CJ-SD jest to

$$\mathcal{E}_{ave}^{CJ-SD} = \frac{21}{16} = 1.3125, \quad (2.2)$$

natomiast w przypadku proponowanej konstelacji zbudowanej za pomocą algorytmu RD wynosi ona

$$\mathcal{E}_{ave}^{RD} = \frac{18}{16} = 1.125. \quad (2.3)$$

Dzięki obniżeniu średniej energii przypadającej na punkt konstelacji otrzymano zysk ok. 0.45 dB w porównaniu z konstelacją 16-CJ-SD oraz ok. 2.45 dB względem konstelacji 16-CIC. Wyniki te uzyskano na drodze symulacyjnej badając system 3D-OFDM [8, 17] wykorzystujący 1024 podnośne. Ponadto zaproponowany algorytm RD sprawuje się równie dobrze w przypadku konstelacji dwuwymiarowych [15].



Rysunek 2.10: BER w funkcji E_b/N_0 dla systemu 3D-OFDM składającego się z 1024 podnośnych.

Etykietowanie konstelacji

Następnym zaproponowanym w pracy algorytmem był warstwowy algorytm etykietowania konstelacji (ang. *layer labelling algorithm, LL-algorithm*) [16]. Do oceny jakości etykietowania wykorzystano średnią odległość Hamminga konstelacji. Średnia odległość Hamminga konstelacji \mathcal{C} określona jest następującą zależnością.

$$d_H^{ave}(\mathcal{C}) = \frac{1}{N} \sum_{\mathbf{p} \in \mathcal{C}} d_H^{ave}(\mathbf{p}) = \frac{1}{2N} \sum_{\mathbf{p} \in \mathcal{C}} \sum_{\mathbf{p}' \in N(\mathbf{p})} \frac{d_H(\mathbf{p}, \mathbf{p}')}{|N(\mathbf{p})|}, \quad (3.1)$$

gdzie $N(\mathbf{p})$ jest zbiorem punktów złożonym z sąsiadów punktu \mathbf{p} a $|\cdot|$ oznacza liczbę elementów danego zbioru.

Algorytm warstwowy porównano z innymi sposobami etykietowania, tj. z algorytmem losowym, algorytmem wykorzystującym porządek leksykograficzny, algorytmem zachłannym wykorzystującym system pierwiastkowy kraty oraz zbiór wektorów bazowych. Działanie algorytmu losowego jest samo wytłumaczalne. Algorytm leksykograficzny wykorzystywał tzw. porządek leksykograficzny punktów konstelacji w celu przyporządkowania etykiet bitowych kolejnym punktom. Na rys. 3.1a pokazano, iż punkty w porządku leksykograficznym etykietowane są kolejno ciągami bitów: 00000, 00001, 00010, 00011, etc. Wadą tego rozwiązania jest to, iż niektóre etykiety bitowe zawsze będą przyporządkowane w sposób nieoptymalny. Można to dostrzec na rys. 3.1a, iż odległość Hamminga pomiędzy punktem położonym w centrum (o etykiecie 00011) wynosi 3 do jednego z sąsiadów (punkt o etykiecie 00100). Przyczynia się to do nieoptymalnego rozmieszczenia etykiet bitowych w całej konstelacji. Kolejnym algorytmem jest algorytm zachłanny wykorzystujący system pierwiastkowy kraty. Dla porządku warto wyjaśnić, iż system pierwiastkowy kraty składa się z wektorów, których początek leży w pewnym punkcie kraty natomiast końce znajdują się w najbliższych punktach sąsiednich*. Na rys. 3.1b punkt w centrum otrzymuje etykietę 00000, natomiast etykiety punktów sąsiednich wybierane są w taki sposób, aby minimalizować metrykę

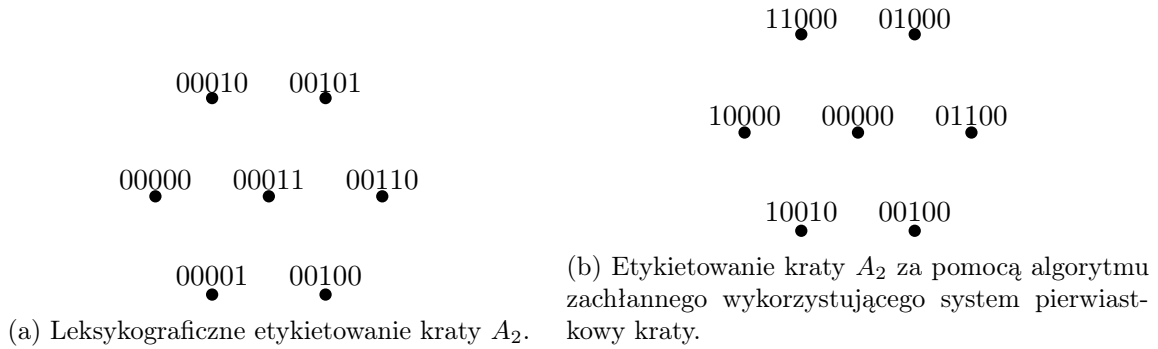
$$H(\mathbf{s}) = \sum_{\mathbf{s}' \in L(\mathbf{s})} d_H(\mathbf{s}, \mathbf{s}'), \quad (3.2)$$

gdzie $L(\mathbf{s})$ jest zbiorem punktów sąsiednich dla punktu \mathbf{s} mających już przyporządkowane etykiety bitowe. Minimalizacja tej metryki jest dokonywana poprzez przydzielanie etykiet kolejnym punktom \mathbf{p} w następujący sposób:

$$\mathbf{p} \mapsto \arg \min_{\mathbf{b} \in \mathcal{B}'} d_H \left(\mathbf{b}, \sum_{\mathbf{v} \in \Phi} l(\mathbf{p} + \mathbf{v}) \right), \quad (3.3)$$

gdzie \mathcal{B}' jest zbiorem jeszcze nie wykorzystanych etykiet bitowych, Φ oznacza system pierwiastkowy kraty, natomiast funkcja $l(\mathbf{q})$ wyznacza etykietę bitową przypisaną punktowi \mathbf{q} ,

*W ogólności nie jest to prawdą, jednakże na potrzeby tego przykładu taka definicja wydaje się być wystarczającą.

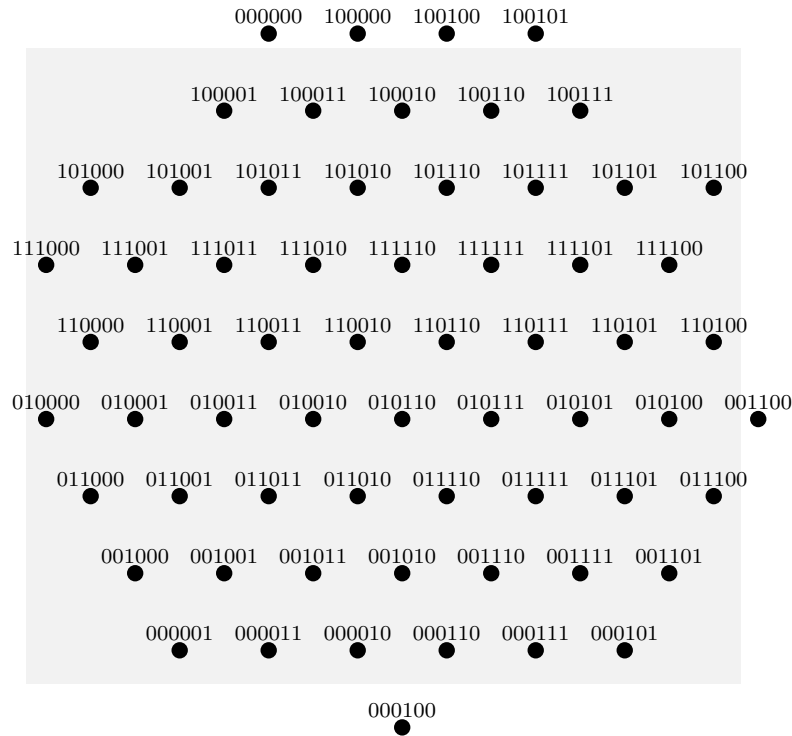


Rysunek 3.1: Przykładowe algorytmy etykietowania.

lub 0, jeśli punkt \mathbf{q} nie posiada jeszcze przypisanej żadnej etykiety. Na podobnej zasadzie działa algorytm zachłanny wykorzystujący zbiór wektorów bazowych. Różnica polega jedynie na tym, iż w przypadku kraty A_2 algorytm *zwiadzałby* kratę nie idąc we wszystkich sześciu wybranych kierunkach, lecz jedynie czterech i to takich, które tworzą parami wzajemnie przeciwstawne wektory.

Najłatwiej wy tłumaczyć działanie algorytmu LL posługując się bardzo prostym przykładem. Na rys. 3.2 przedstawiono przykład konstelacji $64-A_2$, dla której etykiety bitowe zostały przypisane za pomocą algorytmu LL. W początkowej fazie algorytm ten wyznacza zakres punktów, w którym będzie się on poruszał. Ponieważ konstelację przedstawioną na rys. 3.2 można podzielić na 10 wierszy a algorytm *potrafi* etykietować konstelację, gdy liczba wierszy (lub też warstw) jest potęgą dwójki, to najpierw należy wybrać warstwy (wiersze), które będą etykietowane jako pierwsze. Na rys. 3.2 są to te wiersze, które leżą wewnątrz zacienionego obszaru. Ponieważ liczba *dostępnych* bitów wynosi 6 ($2^6 = 64$ punkty konstelacji), to algorytm przeznacza po 3 bity na każdy wymiar. Następnie algorytm przypisuje trzem pierwszym bitom kolejnych warstw etykiety w kodzie Graya oraz w ten sam sposób trzem ostatnim bitom wewnątrz każdej z warstw. Dodatkowo algorytm *dba* o to, aby tak przypisywane etykiety były odpowiednio przesunięte w taki sposób, aby minimalizować średnią odległość Hamminga pomiędzy warstwami. Punkty, którym nie zostaną przypisane etykiety bitowe po realizacji wyżej opisanego procesu, są etykietowane bądź to wykorzystując przeszukiwanie wyczerpujące w celu znalezienia optymalnego przyporządkowania etykiet, bądź używając do tego celu algorytmu zachłannego wykorzystującego system pierwiastkowy kraty. Jeśli liczba punktów pozostałych do etykietowania jest większa niż 9, to wtedy zawsze wykorzystywany jest algorytm zachłanny wykorzystujący system pierwiastkowy kraty. Warto zauważyć, że efektem końcowym etykietowania konstelacji QAM za pomocą algorytmu warstwowego będzie również optymalne przypisanie etykiet bitowych wszystkim punktom konstelacji. Wreszcie wyjściową konstelację wraz z etykietami bitowymi poddano heurystycznej optymalizacji wykorzystującej algorytm genetyczny w celu sprawdzenia, czy można poprawić otrzymaną wartość średniej odległości Hamminga w konstelacji.

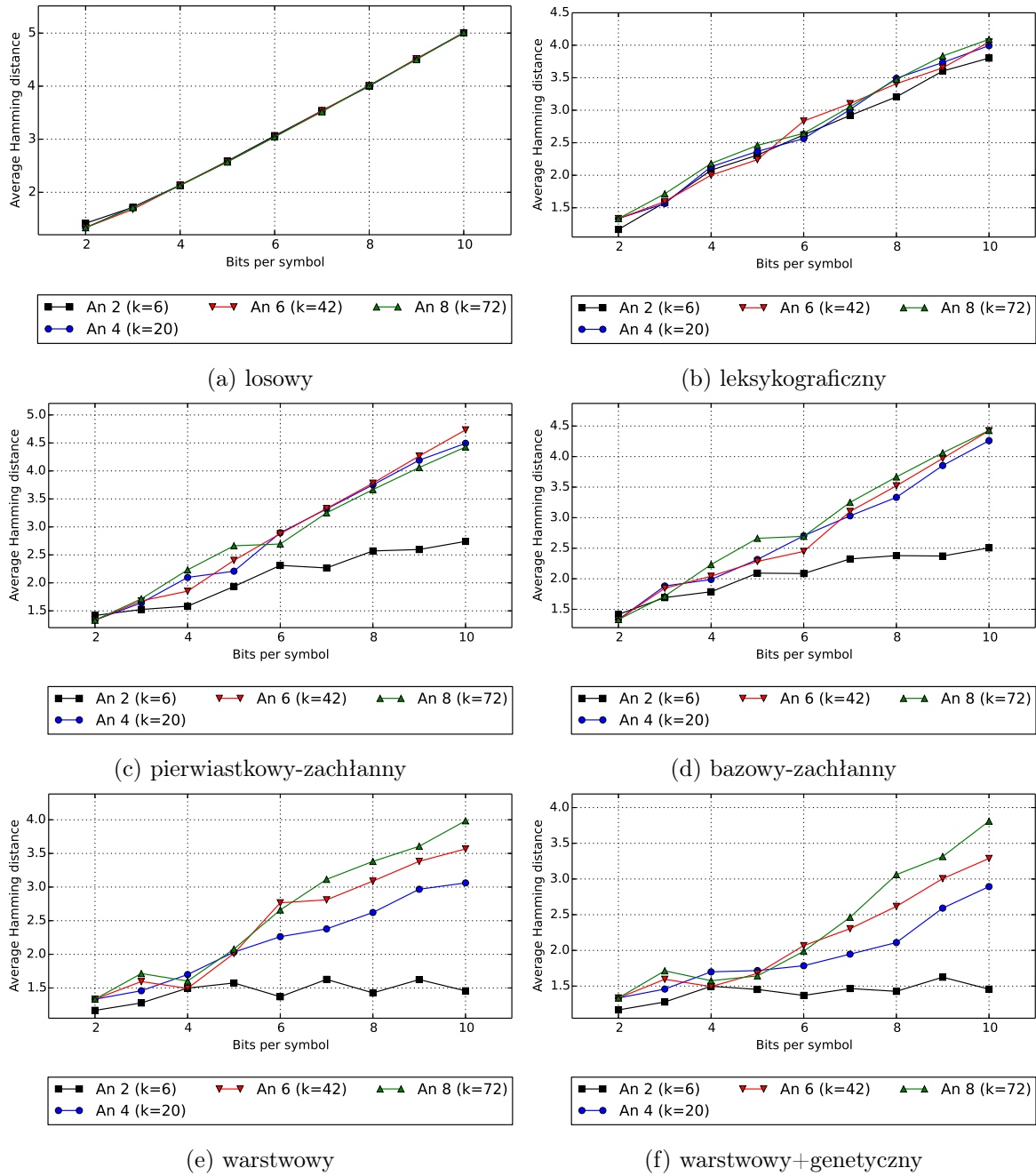
Średnią odległość Hamminga dla konstelacji bazujących na kratach A_n przy wykorzystaniu różnych algorytmów etykietowania pokazano na rys. 3.3. W przypadku algorytmu losowego (rys. 3.3a) wyniki są dość przewidywalne. Otóż średnia odległość Hamminga konstelacji jest zależna od liczby bitów i wynosi (z dość dobrą dokładnością) $b/2$, gdzie b jest liczbą bitów przypadających na punkt konstelacji. Nieco tylko lepsze wyniki uzyskano w przypadku algorytmu wykorzystującego porządek leksykograficzny (rys. 3.3b). Natomiast w przypadku algorytmów zachłannych wykorzystujących system pierwiastkowy kraty lub zbiór jej wektorów bazowych (rys. 3.3c oraz rys. 3.3d) widać pogorszenie w przypadku konstelacji o wymiarach większych niż 2, tj. 4, 6 oraz 8. W przypadku konstelacji bazujących na kracie A_2 (w literaturze zwanych również TQAM), widać dość znaczne poprawienie średniej odle-



Rysunek 3.2: konstelacja $64-A_2$ z etykietami bitowymi przyporządkowanymi za pomocą algorytmu LL

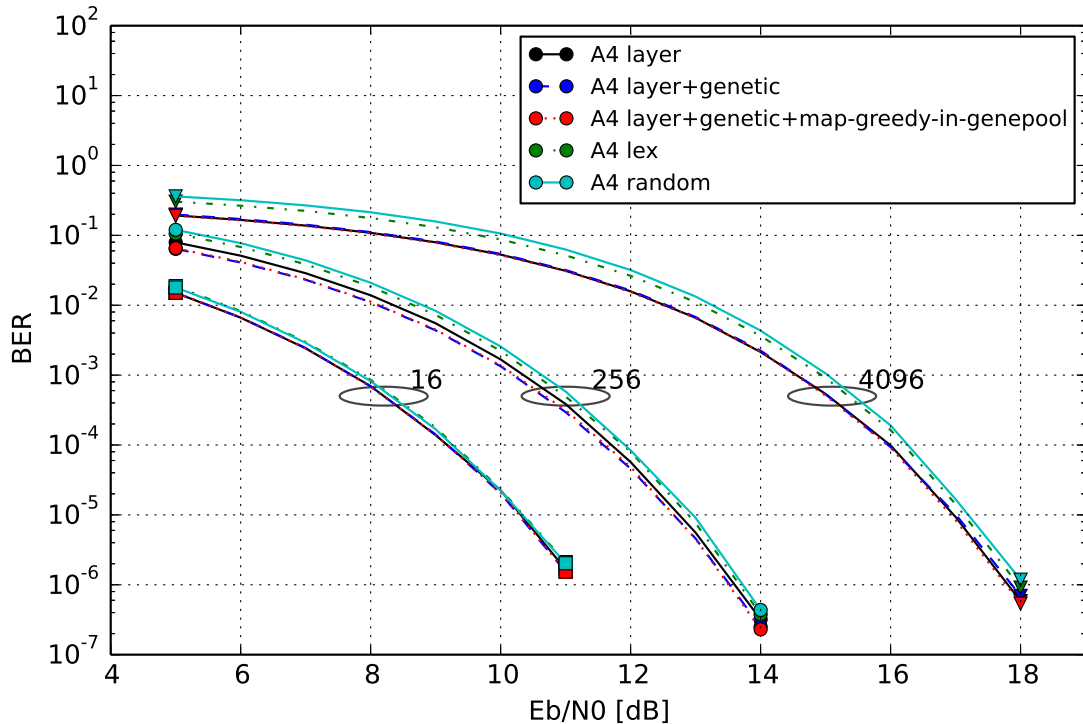
głości Hamminga w funkcji liczby bitów na symbol. Od poprzednich wyników wyróżnia się zaproponowany w pracy algorytm warstwowy (algorytm LL). Na rys. 3.3e widać, iż zastosowanie tego algorytmu poprawia w widoczny sposób średnią odległość Hamminga konstelacji dla wymiarów 2, 4 oraz 6. W przypadku konstelacji 8-wymiarowych widać nieznacznie tylko poprawę względem algorytmu leksykograficznego (rys. 3.3b). Dodatkowo w pracy zbadano możliwość poprawy średniej odległości Hamminga konstelacji za pomocą algorytmu genetycznego (rys. 3.3f). W tym wypadku otrzymano pewną poprawę względem *czystego* algorytmu warstwowego. Jest to widoczne zwłaszcza w przypadku konstelacji o niższej liczbie bitów na symbol, choć także dla 8-wymiarowej konstelacji złożonej z 1024 punktów (10 bitów na symbol) widać spadek średniej odległości Hamminga konstelacji z 4.0 do ok. 3.8. Na koniec warto jeszcze zaznaczyć, że jakość etykietowania (w sensie średniej odległości Hamminga konstelacji) uzyskana za pomocą algorytmu warstwowego (LL) jest także marginalnie lepsza w przypadku konstelacji 64-TQAM (rys. 3.2) niż etykietowania znane obecnie w literaturze [22], co potwierdza między innymi publikacja wynikająca z badań prowadzonych podczas studiów doktoranckich [15].

Na rys. 3.4 pokazano jak średnia odległość Hamminga konstelacji przekłada się na poziom bitowej stopy błędów (ang. *bit error rate*, *BER*) w przypadku konstelacji bazujących na kracie A_4 . Na wspomnianym rysunku można z łatwością zauważyć, iż najgorszą bitową stopę błędów przy danym stosunku sygnału do szumu (ang. *signal-to-noise ratio*, *SNR*) daje etykietowanie losowe (na rysunku oznaczone jako *random*). Kolejnym co do jakości jest algorytm wykorzystujący porządek leksykograficzny (na rysunku oznaczony jako *lex*) – poprawa w stosunku do algorytmu losowego jest w tym wypadku znikoma. Algorytm warstwowy (na rysunku oznaczony jako *layer*) oraz warstwowy poprawiony przez algorytm genetyczny (na rysunku oznaczony jako *layer+genetic*) dają najlepsze wyniki (poprawa o ok. 0.45 dB względem losowego przyporządkowywania etykiet bitowych). W przypadku konstelacji $256-A_4$ dodanie algorytmu genetycznego poprawia w pewnym stopniu jakość dekodowania względem sytuacji, gdy algorytm genetyczny nie jest zastosowany. Na rys. 3.4 widoczny



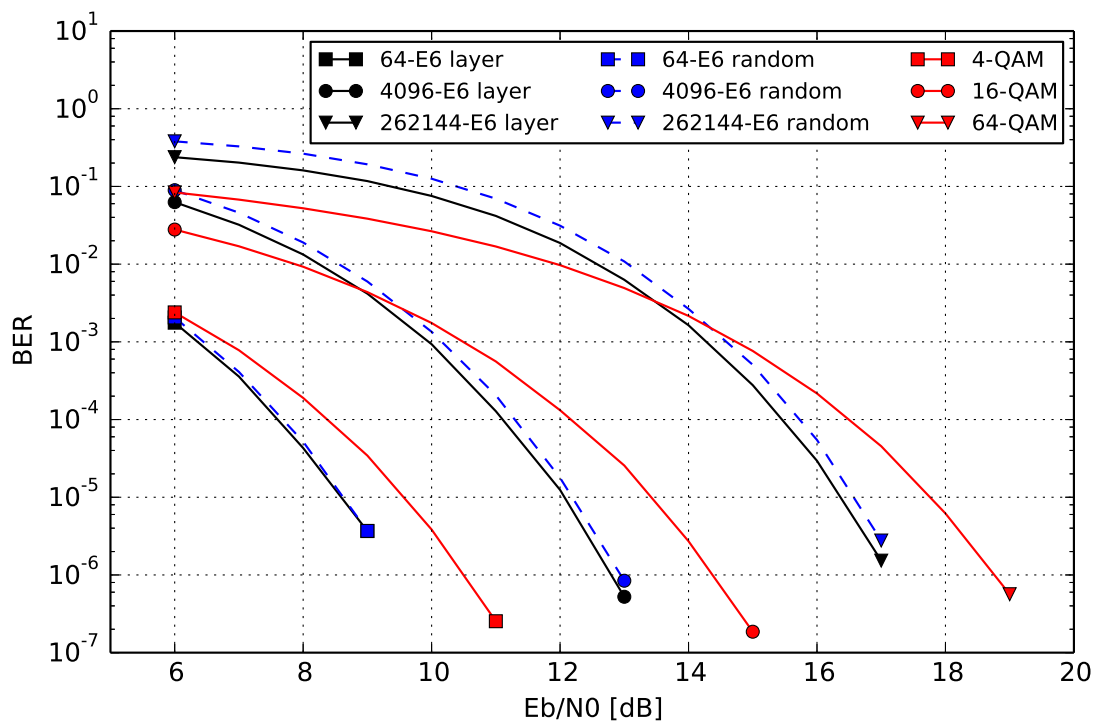
Rysunek 3.3: Średnia odległość Hamminga dla konstelacji zbudowanych na bazie kraty A_n (różne algorytmy etykietowania, k – liczba punktów sąsiednich). Wymiarowość kraty/konstelacji podano w legendach pod rysunkami.

jest także dodatkowy algorytm, tj. *layer+genetic+map-greedy-in-genepool*. Różnica między nim a algorytmem *layer+genetic* wynika jedynie z przygotowywania wstępnej puli osobników (genów) dla algorytmu genetycznego. W przypadku algorytmu oznaczonego na rys. 3.4 jako *layer+genetic+map-greedy-in-genepool*, do pierwszego pokolenia osobników dodano także etykietowanie będące wynikiem zastosowania algorytmu zachłannego wykorzystującego system pierwiastkowy kraty. Widać jednak, iż dodanie dodatkowego osobnika do początkowej puli genów nie poprawiło otrzymanych wyników.



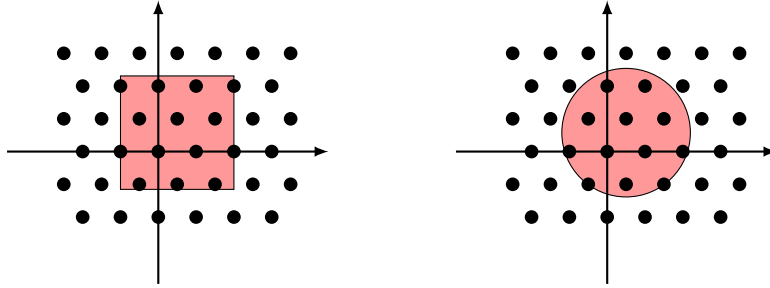
Rysunek 3.4: BER w funkcji E_b/N_0 dla konstelacji 16-, 256- oraz 4096- A_4 etykietowanej za pomocą różnych algorytmów.

Porównanie konstelacji wielowymiarowych z konstelacją QAM zostało przedstawione na rys. 3.5. Do porównania wybrano konstelacje $64-E_6$, $4096-E_6$ oraz $262144-E_6$. (Diagram Dynkina dla kraty E_6 pokazano wcześniej na rys. 2.6a.) Rozmiary konstelacji 6-wymiarowych dobrano w taki sposób, aby przepływność bitowa sygnałów wykorzystujących odwzorowania QAM oraz konstelacje wielowymiarowe była identyczna, tj. konstelacja $64-E_6$ odpowiada konstelacji 4-QAM, $4096-E_6$ – 16-QAM, etc. Na rys. 3.5 widać, iż niezależnie od rozmiaru konstelacji wykorzystanie konstelacji wielowymiarowych daje zysk od ok. 1 dB do *circa* 1.6 dB w zależności od rozmiaru konstelacji. Dodatkowo warto zauważyć, iż zastosowanie algorytmu warstwowego do etykietowania konstelacji pozwala uzyskać taką samą bitową stopę błędów przy stosunku E_b/N_0 niższym o ok. 0.1 dB do 0.2 dB.

Rysunek 3.5: Porównanie konstelacji 64/4096/262144- E_6 oraz 4/16/64-QAM.

Detekcja symboli wielowymiarowych

Rozwiązaniem stosowanym w literaturze do detekcji symboli wielowymiarowych jest tzw. dekodery sferyczny (ang. *sphere decoder*, *SD*) [20,27]. W rozprawie zaproponowano alternatywny algorytm detekcji – demodulator hipersześcienny. Różnice pomiędzy demodulatorem hipersześciennym a dekodery sferycznym pokazano na rys. 4.1. Jak widać, w przypadku dekodera sferycznego obszarem poszukiwań jest wnętrze okręgu (w przypadku wielowymiarowym jest to hipersfera). Demodulator hipersześcienny poszukuje natomiast najbliższego symbolu do symbolu odebranego wewnątrz kwadratu (ponownie w przypadku wielowymiarowym jest to hipersześcian).



Rysunek 4.1: Obszary w których odbywa się poszukiwanie estymaty nadanego symbolu; demodulator hipersześcienny (po lewej) oraz dekodery sferyczny (po prawej).

Zaproponowany algorytm demodulacji podzielony jest na dwie fazy. Pierwsza z nich, tzw. faza inicjacji, musi zostać wykonana jedynie raz dla każdej rozważanej konstelacji i związana jest ona z kwantyzacją hipersześcienną. Można ją przeprowadzić poza odbiornikiem, a wyniki jej obliczeń umieścić bezpośrednio w pamięci nieulotnej urządzenia odbiorczego. Druga faza to demodulacja właściwa. Podczas fazy inicjacji obszar konstelacji dzielony jest przez hipersześciany o boku równym d_{min} , tj. minimalnej odległości pomiędzy dwoma punktami konstelacji. Otrzymany w ten sposób podział wyznacza zbiory \mathcal{G}_i , które zawierają punkty konstelacji zawierające się wewnątrz kolejnych hipersześcianów. Po pokryciu całej konstelacji tworzony jest kolejny zbiór punktów w następujący sposób

$$\bigwedge_{\mathcal{G}_i} \mathcal{H}_i = \bigcup_{\mathcal{G}_k \in \mathcal{G}_i^{adj}} \mathcal{G}_k, \quad (4.1)$$

gdzie \mathcal{G}_i^{adj} jest zbiorem zawierającym punkty należące do \mathcal{G}_i oraz do jego $3^n - 1$ przyległych hipersześcianów, gdzie n jest liczbą wymiarów konstelacji.

W pracy porównano złożoność zaproponowanego rozwiązania, tj. demodulatora hipersześciennego, dekodera sferycznego oraz metody wykorzystującej przeszukiwanie wyczerpujące. W założeniu wszystkie te rozwiązania można uważać za inne formy detekcji ML (ang. *ma-*

ximum likelihood). Do oceny złożoności posłużono się liczbą operacji zmiennoprzecinkowych (ang. *floating-point operations, FLOPs*).

W pracy [7] złożoność dekodera sferycznego definiowana jest w następujący sposób

$$C_{SD} = \sum_i^n N_i^{S(d)} f_p(i), \quad (4.2)$$

gdzie $N_i^{S(d)}$ to liczba punktów w i -wymiarowej sferze o promieniu d , natomiast $f_p(i) = 2i + 11$ odpowiada liczbie operacji zmiennoprzecinkowych (ang. *floating point operations, flops*) wykonywanych przez algorytm w każdym rozpatrywanym wymiarze i . Przy założeniu, że sygnał jest zniekształcony jedynie poprzez dodanie próbki szumu, to wtedy złożoność dekodera sferycznego można oszacować w następujący sposób [6, 7, 13, 14]:

$$C_{SD} = \sum_i^n \frac{(\pi)^{i/2}}{\Gamma(1 + i/2) \sqrt{i+1}} f_p(i), \quad (4.3)$$

gdzie $f_p(i) = 2i + 11$ określa liczbę elementarnych operacji zmiennoprzecinkowych (mnożenia, dodawania, etc.) na wymiar i , które muszą zostać wykonane podczas *wizytowania* przez algorytm każdego sprawdzanego punktu. Taka definicja złożoności odpowiada liczbie operacji zmiennoprzecinkowych wymaganych do demodulacji symbolu n -wymiarowego.

W związku z tym, iż zaproponowany algorytm może znaleźć rozwiązanie zarówno w hipersześcianach \mathcal{H}_i jak i przeszukując jedynie \mathcal{G}_i , to najwłaściwsze wydaje się podanie dwóch złożoności, tj. dolnej C_{HD}^L oraz górnej C_{HD}^U . Są one zdefiniowane w następujący sposób

$$\begin{aligned} C_{HD}^U &= 3n + \frac{V(C, 3d_{min})}{V(A_n^{Vor})} f_p^{HD}(n) \\ C_{HD}^L &= 3n + \frac{V(C, d_{min})}{V(A_n^{Vor})} f_p^{HD}(n), \end{aligned} \quad (4.4)$$

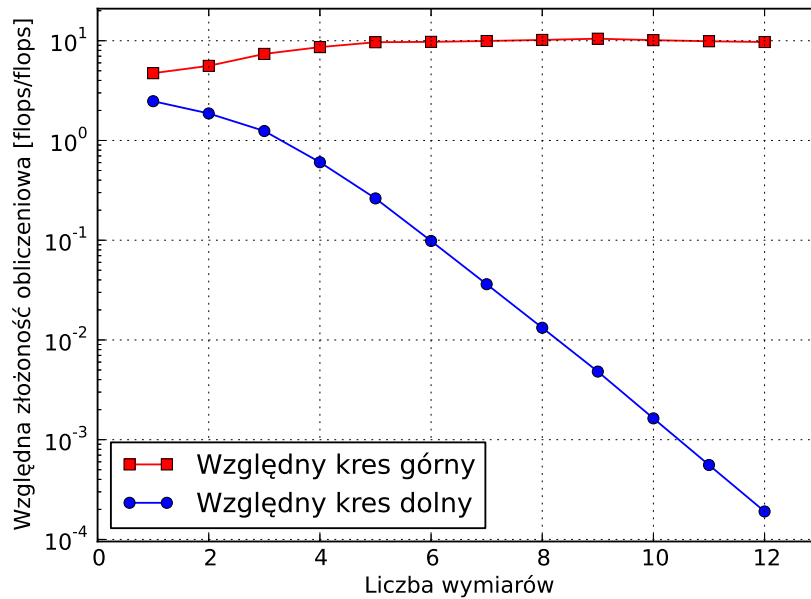
gdzie $V(C, a)$ i $V(A_n^{Vor})$ to odpowiednio objętość hipersześcianu o boku długości a oraz objętość komórki Woronoja kraty A_n . Czynniki $f_p^{HD}(n)$ opisuje liczbę operacji zmiennoprzecinkowych wymaganych do demodulacji jednego n -wymiarowego symbolu i jest on równy

$$f_p^{HD}(n) = 1.5n + 1. \quad (4.5)$$

Składnik $3n$ w (4.4) odpowiada liczbie operacji potrzebnej do wyszukania właściwego hipersześcianu w tablicy LUT (ang. *look-up table*).

Stosunek złożoności dekodera sferycznego do demodulatora hipersześciennego pokazano na rys. 4.2. Względny kres górny jest zdefiniowany jako C_{SD}/C_{HP}^L , natomiast względny kres dolny jako C_{SD}/C_{HP}^U . Względny kres górny (dolny) opisuje, ile razy złożoność demodulatora hipersześciennego jest lepsza (gorsza) od złożoności dekodera sferycznego. Z rys. 4.2 wynika, iż zaproponowany algorytm może mieć w najlepszym wypadku złożoność dziesięciokrotnie mniejszą niż dekodery sferyczne.

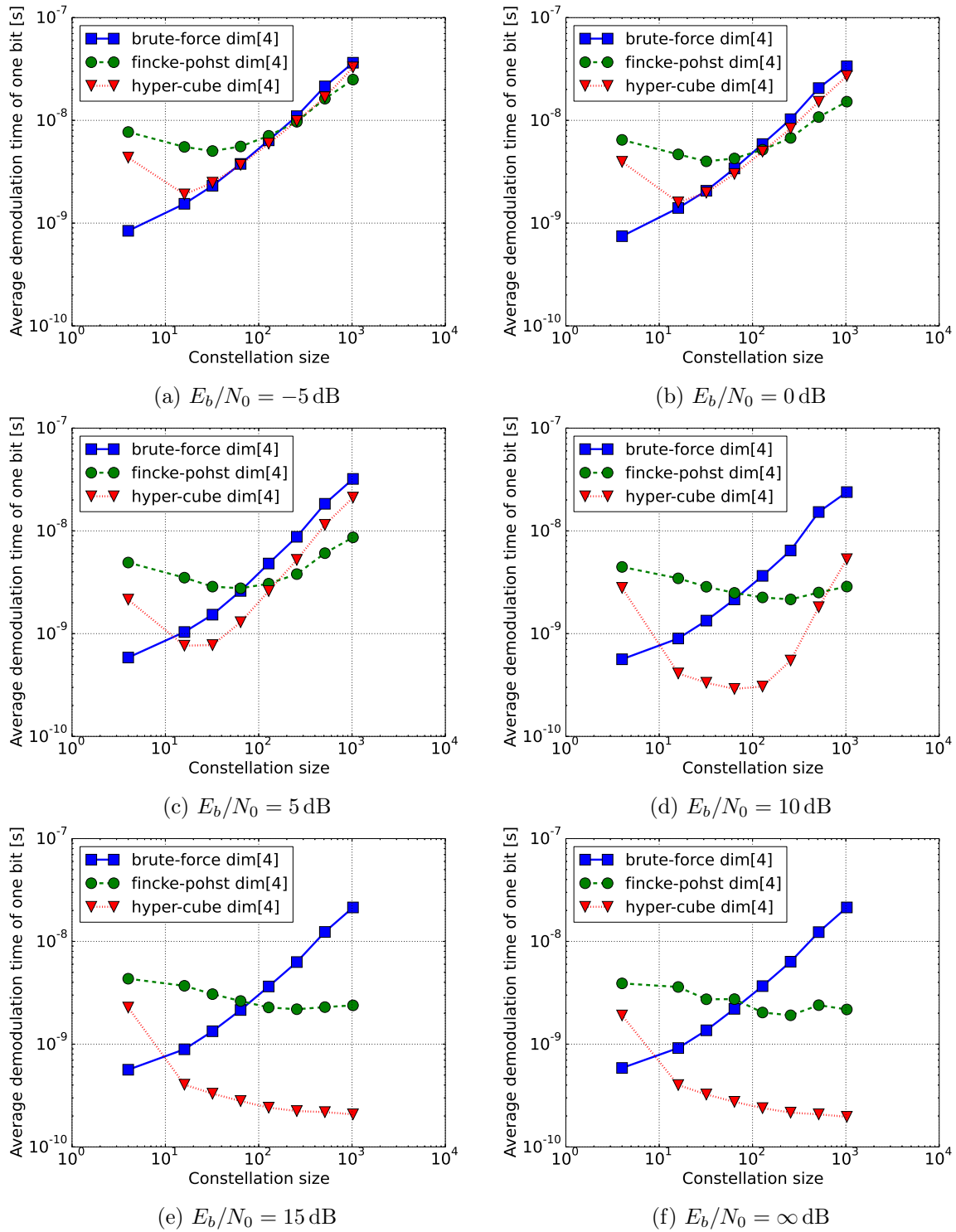
Na rys. 4.3 przedstawiono średni czas demodulacji pojedynczego bitu dla konstelacji wielowymiarowej bazującej na kracie A_4 . Należy jednak zauważyć, iż wyniki przedstawione na rys. 4.3 nie mogą zostać użyte do określenia maksymalnej przepustowości systemu wynikającej z szybkości działania odbiornika. Jest to spowodowane nie tylko tym, iż wyniki przedstawione na rys. 4.3 zostały otrzymane po przekazaniu wskaźnika `-O0` do kompilatora, która wyłącza optymalizację kodu wynikowego podczas kompilacji, lecz głównie z tego powodu, iż typowo implementacja odbiornika jest realizowana sprzętowo. Gwoli ścisłości przebadano również przypadek, gdy najwyższy poziom optymalizacji w kompilatorze GCC był włączona, tj. przekazano `-O3` do kompilatora. W związku z powyższym wyniki przedstawione



Rysunek 4.2: Stosunek złożoności algorytmu Fincke-Pohsta i algorytmu hipersześciennego.

na rys. 4.3 powinny być traktowane jedynie w kontekście wzajemnych relacji pomiędzy sobą. Warto zauważyć również, że wyniki uzyskane za pomocą symulacji komputerowej (rys. 4.3) potwierdzają wyniki teoretyczne (rys. 4.2) wskazujące, iż demodulator hipersześcienne może być w pewnych sytuacjach nie więcej niż dziesięciokrotnie szybszy od dekodera. Rozważono następujące konstelacje: $4-A_4$, $16-A_4$, $32-A_4$, $64-A_4$, $128-A_4$, $256-A_4$, $512-A_4$ oraz $1024-A_4$.

Na rys. 4.3a oraz 4.3b pokazano średni czas demodulacji pojedynczego bitu, gdy stosunek energii przypadającej na bit do gęstości widmowej mocy szumu wynosił odpowiednio -5 dB oraz 0 dB. W takim wypadku zarówno demodulator hipersześcienne (na rysunkach oznaczony jako "hyper-cube" jak i dekodery sferyczne (na rysunkach oznaczony jako "fincke-pohst") mają podobne średnie czasy detekcji pojedynczego bitu. W przypadku, gdy rozmiar konstelacji był mniejszy niż 128 punktów, to proponowane rozwiązania okazywało się być szybsze niż dekodery sferyczne. Jednakże dla tak niskich wartości E_b/N_0 oba rozwiązania cechowały się podobnym średnim czasem detekcji pojedynczego bitu jak przeszukiwanie wyczerpujące (na rysunkach oznaczone jako "brute-force"). Podobny trend można było zaobserwować gdy stosunek E_b/N_0 rósł do 5 dB (rys. 4.3c), przy czym demodulator hipersześcienne był w tym wypadku szybszy niż przeszukiwanie wyczerpujące prawie dla wszystkich rozmiarów konstelacji za wyjątkiem czteropunktowej konstelacji A_4 . Sytuacja zmieniła się diametralnie, gdy stosunek E_b/N_0 wzrósł do 10 dB (rys. 4.3d). Wtedy to demodulator hipersześcienne był znacznie szybszy niż dekodery sferyczne dla konstelacji o rozmiarach 16, 32, 64, 128 i 256 punktów oraz nieznacznie szybszy dla konstelacji o rozmiarze 512 punktów. Różnica pomiędzy sytuacją, gdy stosunek E_b/N_0 wynosił 15 dB (rys. 4.3e) a kanałem bezszumnym, tj. $E_b/N_0 \rightarrow \infty$, (rys. 4.3f) jest praktycznie niewielka. W obu wypadkach demodulator hipersześcienne okazuje się być ok. dziesięciokrotnie szybszy niż dekodery sferyczne, gdy tylko konstelacja jest złożona z przynajmniej 16 punktów. Ponadto ciekawy efekt można zaobserwować na rys. 4.3e oraz 4.3f. Da się zauważyć, iż średni czas demodulacji pojedynczego bitu spada w funkcji rozmiaru konstelacji. Wynika to głównie z tego, iż rozmiar konstelacji rośnie znacznie szybciej niż czas demodulacji pojedynczego symbolu, a ponieważ liczba bitów przypadająca na punkt konstelacji jest proporcjonalna do liczby punktów konstelacji, to maleje również średni czas potrzebny na detekcję pojedynczego bitu.



Rysunek 4.3: Średni czas demodulacji jednego bitu dla konstelacji bazującej na kracie A_4 dla różnych wartości E_b/N_0 .

Podsumowanie

W rozprawie poruszono trzy tematy związane z modulacjami wielowymiarowymi. Pierwszy odnosił się do sposobu budowania konstelacji wielowymiarowych. Drugi poruszył temat etykietowania konstelacji wielowymiarowych. Trzeci natomiast był związany z detekcją symboli wielowymiarowych.

W celu budowania konstelacji zaproponowano dwa* algorytmy. Pierwszy z nich, tj. algorytm wykorzystujący sklejanie (ang. *gluing algorithm*), umożliwia budowanie konstelacji wielowymiarowych wykorzystujących rodzinę krat A_n . Drugi algorytm jest pod tym względem bardziej generyczny i pozwala na konstrukcję konstelacji, jeśli tylko istnieje system pierwiastkowy danej kraty. Algorytm RD, bo o nim mowa, pozwala na uzyskanie bardzo dobrze *upakowanej* konstelacji również w dwóch wymiarach [15], natomiast w przypadku trzech wymiarów umożliwia on uzyskanie konstelacji o lepszych parametrach niż obecnie wykorzystywane [17].

Następnym obszarem związanym z konstelacjami wielowymiarowymi, któremu została poświęcona część rozprawy, była tematyka przyporządkowywania etykiet bitowych poszczególnym punktom konstelacji (tzw. etykietowanie) w sposób optymalny. Zaproponowany algorytm, nazwany algorytmem warstwowym (ang. *layer labelling algorithm*), umożliwia takie przyporządkowanie etykiet bitowych punktom konstelacji, iż przy stosunku sygnału do szumu niższym o ok. 0.5 dB bitowa stopa błędów utrzymuje się na tym samym poziomie.

Wreszcie w rozprawie poruszono sposób detekcji symboli wielowymiarowych. Algorytm zaproponowany w tym celu, tj. demodulator hipersześcienne, umożliwia detekcję symboli wielowymiarowych w czasie ok. dziesięciokrotnie krótszym niż w przypadku jednego ze stosowanych rozwiązań, tj. dekodera sferycznego. Rozważania teoretyczne, które doprowadziły do takiego wniosku, zostały potwierdzone odpowiednimi symulacjami komputerowymi. Co więcej wszystkie porównywane rozwiązania, tj. przeszukiwanie wyczerpujące, dekodery sferyczny oraz demodulator hipersześcienne, cechowały się identyczną jakością odbioru.

*Właściwie w rozprawie pokazane są trzy algorytmy służące do budowania konstelacji wielowymiarowych. Lecz jeden z nich (nie przedstawiony w niniejszym autoreferacie) był wykorzystany jedynie jako krok pośredni do przedstawienia algorytmu RD.

Bibliografia

- [1] B. Chen and M. Jiang. Design of Three-Dimensional Constellations for Wireless Communication Systems. In *IEEE International Conference on Communications*, pages 2876–2881, 2015.
- [2] Z. Chen, J. S. Bae, S. Chung, J. Koh, and S. G. Kang. Multi-Envelope 3-Dimensional Constellations for Polarization Shift Keying Modulation. In *2010 International Conference on Information and Communication Technology Convergence, ICTC 2010*, pages 173–174, 2010.
- [3] A. Dochhan, H. Griebner, B. Teipen, and M. Eiselt. A 3-Dimensional Simplex Modulation Format with Improved OSNR Performance Compared to DP-BPSK. In *2013 ITG Symposium Proceedings - Photonic Networks*, pages 1–5, 2013.
- [4] J. K. Fischer, S. Alreesh, R. Elschner, F. Frey, M. Nölle, C. Schmidt-Langhorst, and C. Schubert. Bandwidth-Variable Transceivers based on Four-Dimensional Modulation Formats. *Journal of Lightwave Technology*, 32(16):2886–2895, 2014.
- [5] J. S. Han and M. J. Kim. Offset Quadrature-Quadrature Phase Shift Keying with Half-Sine Pulse Shaping. In *International Conference on ICT Convergence*, pages 931–935, 2013.
- [6] B. Hassibi and H. Vikalo. On the Expected Complexity of Sphere Decoding. In *Conference Record of Thirty-Fifth Asilomar Conference on Signals, Systems and Computers (Cat.No.01CH37256)*, volume 2, pages 1–32, 2001.
- [7] B. Hassibi and H. Vikalo. On the Sphere-Decoding Algorithm I. Expected Complexity. *IEEE Transactions on Signal Processing*, 53(8):2806–2818, aug 2005.
- [8] S. G. Kang. An OFDM with 3-D Signal Mapper and 2-D IDFT Modulator. *IEEE Communications Letters*, 12(12):871–873, 2008.
- [9] M. Karlsson and E. Agrell. Spectrally Efficient Four-Dimensional Modulation. In *Optical Fiber Communication Conference*, pages 1–3, Washington, D.C., 2012. Osa.
- [10] M. Karlsson and E. Agrell. Multidimensional Modulation and Coding in Optical Transport. *Journal of Lightwave Technology*, 35(4):876–884, 2017.
- [11] J. Karout, X. Liu, S. Chandrasekhar, E. Agrell, M. Karlsson, and R. J. Essiambre. Experimental Demonstration of an Optimized 16-ary Four-Dimensional Modulation Format Using Optical OFDM. In *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference*, pages 1–3, Washington, D.C., 2013. Osa.
- [12] T. G. Markiewicz. An Energy Efficient QAM Modulation with Multidimensional Signal Constellation. *International Journal of Electronics and Telecommunications*, 62(2):159–165, 2016.

- [13] T. G. Markiewicz. Faster than Sphere (Decoder). A Demodulation Algorithm for Multidimensional Constellations. In *International Symposium on Wireless Communication Systems*, pages 136–142, 2016.
- [14] T. G. Markiewicz. O algorytmach demodulacji konstelacji wielowymiarowych. *Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne*, 84(6):201–204, 2016.
- [15] T. G. Markiewicz. Construction and Labelling of Triangular QAM. *IEEE Communications Letters*, pages 1–4 (to be published), 2017.
- [16] T. G. Markiewicz. Labelling of Multidimensional Constellations. *IEEE Transactions on Communications [submitted to]*, pages 1–30, 2017.
- [17] T. G. Markiewicz. Projektowanie trójwymiarowej konstelacji dla techniki OFDM wykorzystującej dwuwymiarową odwrotną transformację Fouriera. *Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne*, 86(6):315–318, 2017.
- [18] H. Nikopour and H. Baligh. Sparse Code Multiple Access. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, pages 332–336, 2013.
- [19] H. Nikopour, E. Yi, A. Bayesteh, K. Au, M. Hawryluck, H. Baligh, and J. Ma. SCMA for Downlink Multiple Access of 5G Wireless Networks. In *2014 IEEE Global Communications Conference*, pages 3940–3945, 2014.
- [20] Frédérique Oggier and Emanuele Viterbo. Algebraic Number Theory and Code Design for Rayleigh Fading Channels. *Foundations and Trends in Communications and Information Theory*, 1(3):333 – 415, 2004.
- [21] S. J. Park. Triangular Quadrature Amplitude Modulation. *IEEE Communications Letters*, 11(4):292–294, 2007.
- [22] S. J. Park and M. K. Byeon. Error Performances of 64-ary Triangular Quadrature Amplitude Modulation in AWGN Channel. In *Vehicular Technology Conference*, pages 1752–1755, 2007.
- [23] D. Saha. Trellis Coded Quadrature-Quadrature Phase Shift Keying. In *Military Communications Conference - Crisis Communications: The Promise and Reality, 1987. MIL-COM 1987. IEEE*, pages 929–933, 1987.
- [24] D. Saha and T. G. Birdsall. Quadrature-Quadrature Phase-Shift Keying. *IEEE Transactions on Communications*, 37(5):437–448, 1989.
- [25] Y. Tao, L. Liu, S. Liu, and Z. Zhang. A Survey: Several Technologies of Non-Orthogonal Transmission for 5G. *China Communications*, 12(10):1–15, 2015.
- [26] M. S. Viazovska. The Sphere Packing Problem in Dimension 8. *Annals of Mathematics*, 185(3):991–1015, 2017.
- [27] E. Viterbo and J. Boutros. A Universal Lattice Code Decoder for Fading Channels. *IEEE Transactions on Information Theory*, 45(5):1639–1642, 1999.