UNIVERSITY OF YAOUNDE I

FACULTÉ DES SCIENCES
FACULTY OF SCIENCES

DÉPARTEMENT DE MATHÉMATIQUES
DEPARTMENT OF MATHEMATICS

POST GRADUATE SCHOOL OF SCIENCES, TECHNOLOGY AND GEOSCIENCES
Centre

Laboratoire d'Algèbre, Géométrie et Applications
Laboratory of Algebra, Geometry and Applications

Option: Algebra

THE CLOSEST VECTOR PROBLEM FOR SOME ROOT LATTICES AND ORTHOGONAL SIEVE ALGORITHM

"Ph.D THESIS"

IN THE FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY(Ph.D)

BY

FOBASSO TCHINDA Arnaud Girès
MASTER IN MATHEMATICS
REGISTRATION NUMBER: 12V0837

BIBLIOTHEQUE UNIVERSITAIRE UY

UNDER

THE DIRECTION OF: AND THE SUPERVISION OF:

Pr. FOUOTSA Emmanuel
ASSOCIATE PROFESSOR,
UNIVERSITY OF BAMENDA

Pr. NKUIMI JUGNIA Célestin
ASSOCIATE PROFESSOR,
UNIVERSITY OF YAOUNDE 1

Year: 2023

REPUBLIQUE DU CAMEROUN
Paix – Travail - Patrie
UNIVERSITÉ DE YAOUNDEI
FACULTÉ DES SCIENCES

REPUBLIC OF CAMEROON
Peace – Work - Fatherland
UNIVERSITY OFYAOUNDE
I FACULTYOF SCIENCE

DÉPARTEMENT DE MATHÉMATIQUES

DEPARTMENT OF MATHEMATICS

ATTESTATION DE CORRECTION DE LA THÈSE DE DOCTORAT/PhD

Nous soussignés, Pr. **AYISSI Raoul Domingo,** Pr. **FOUOTSA Emmanuel,** Pr. **KIANPI Maurice;** membres du jury de la thèse de Doctorat/PhD présenté par **Monsieur FOBASSO TCHINDA Arnaud Girès**, Matricule 12V0837, Thèse intitulé: **«The Closest Vector Problem for Some Root Lattices and Orthogonal Sieve Algorithm»** et soutenu en vue de l'obtention du diplôme de **DOCTORAT/PhD en Mathématiques**, attestons que toutes les corrections demandées par le jury de soutenance en vue de l'amélioration de ce travail, ont été effectuées.

En foi de quoi la présente attestation lui est délivrée pour servir et valoir ce que de droit.

Rapporteur	**Président**	**Examinateur**
Pr. FOUOTSA Emmanuel	**Pr. AYISSI Raoul Domingo**	**Pr. KIANPI Maurice**

UNIVERSITE DE YAOUNDE I
DEPARTEMENT DE
CHEF DE
DEPARTEMENT
MATHEMATIQUES
FACULTE DES SCIENCES

Pr Ayissi Raoul
Université de Yaoundé I

ANNÉE ACADEMIQUE 2021/2022
(Par Département et par Grade)
DATE D'ACTUALISATION 22 juin 2022

ADMINISTRATION

DOYEN : TCHOUANKEU Jean- Claude, *Maître de Conférences*
VICE-DOYEN / DPSAA: ATCHADE Alex de Théodore, *Maître de Conférences*
VICE-DOYEN / DSSE : NYEGUE Maximilienne Ascension, *Professeur*
VICE-DOYEN / DRC : ABOSSOLO ANGUE Monique, *Maître de Conférences*
Chef Division Administrative et Financière : NDOYE FOE Florentine Marie Chantal, *Maître de Conférences*
Chef Division des Affaires Académiques, de la Recherche et de la Scolarité DAARS : AJEAGAH Gideon AGHAINDUM, *Professeur*

1- DÉPARTEMENT DE BIOCHIMIE (BC) (39)

N°	NOMS ET PRÉNOMS	GRADE	OBSERVATIONS
1.	BIGOGA DAIGA Jude	Professeur	En poste
2.	BOUDJEKO Thaddée	Professeur	En poste
3.	FEKAM BOYOM Fabrice	Professeur	En poste
4.	FOKOU Elie	Professeur	En poste
5.	KANSCI Germain	Professeur	En poste
6.	MBACHAM FON Wilfred	Professeur	En poste
7.	MOUNDIPA FEWOU Paul	Professeur	*Chef de Département*
8.	OBEN Julius ENYONG	Professeur	En poste
9.	ACHU Merci BIH	Maître de Conférences	En poste
10.	ATOGHO Barbara MMA	Maître de Conférences	En poste
11.	AZANTSA KINGUE GABIN BORIS	Maître de Conférences	En poste
12.	BELINGA née NDOYE FOE F. M. C.	Maître de Conférences	Chef DAF / FS
13.	DJUIDJE NGOUNOUE Marceline	Maître de Conférences	En poste
14.	EFFA ONOMO Pierre	Maître de Conférences	En poste
15.	EWANE Cécile Annie	Maître de Conférences	En poste
16.	KOTUE TAPTUE Charles	Maître de Conférences	En poste
17.	MOFOR née TEUGWA Clotilde	Maître de Conférences	Doyen FS / UDs
18.	NANA Louise épouse WAKAM	Maître de Conférences	En poste
19.	NGONDI Judith Laure	Maître de Conférences	En poste
20.	NGUEFACK Julienne	Maître de Conférences	En poste
21.	NJAYOU Frédéric Nico	Maître de Conférences	En poste
22.	TCHANA KOUATCHOUA Angèle	Maître de Conférences	En poste
23.	AKINDEH MBUH NJI	Chargé de Cours	En poste
24.	BEBEE Fadimatou	Chargée de Cours	En poste
25.	BEBOY EDJENGUELE Sara Nathalie	Chargé de Cours	En poste
26.	DAKOLE DABOY Charles	Chargé de Cours	En poste

27.	DJUIKWO NKONGA Ruth Viviane	Chargée de Cours	En poste
28.	DONGMO LEKAGNE Joseph Blaise	Chargé de Cours	En poste
29.	FONKOUA Martin	Chargé de Cours	En poste
30.	KOUOH ELOMBO Ferdinand	Chargé de Cours	En poste
31.	LUNGA Paul KEILAH	Chargé de Cours	En poste
32.	MANANGA Marlyse Joséphine	Chargée de Cours	En poste
33.	MBONG ANGIE M. Mary Anne	Chargée de Cours	En poste
34.	OWONA AYISSI Vincent Brice	Chargé de Cours	En poste
35.	Palmer MASUMBE NETONGO	Chargé de Cours	En poste
36.	PECHANGOU NSANGOU Sylvain	Chargé de Cours	En poste
37.	WILFRED ANGIE Abia	Chargé de Cours	En poste

38.	FOUPOUAPOUOGNIGNI Yacouba	Assistant	En poste
39.	MBOUCHE FANMOE Marceline Joëlle	Chargée de Cours	En poste

2- DÉPARTEMENT DE BIOLOGIE ET PHYSIOLOGIE ANIMALES (BPA) (51)

1.	AJEAGAH Gideon AGHAINDUM	Professeur	*DAARS/FS*
2.	BILONG BILONG Charles-Félix	Professeur	Chef de Département
3.	DIMO Théophile	Professeur	En Poste
4.	DJIETO LORDON Champlain	Professeur	En Poste
5.	DZEUFIET DJOMENI Paul Désiré	Professeur	En Poste
6.	ESSOMBA née NTSAMA MBALA	Professeur	*Vice Doyen/FMSB/UYI*
7.	FOMENA Abraham	Professeur	En Poste
8.	KEKEUNOU Sévilor	Professeur	En poste
9.	NJAMEN Dieudonné	Professeur	En poste
10.	NJIOKOU Flobert	Professeur	En Poste
11.	NOLA Moïse	Professeur	En poste
12.	TAN Paul VERNYUY	Professeur	En poste
13.	TCHUEM TCHUENTE Louis Albert	Professeur	*Inspecteur de service* *Coord.Progr./MINSANTE*
14.	ZEBAZE TOGOUET Serge Hubert	Professeur	En poste

15.	ALENE Désirée Chantal	Maître de Conférences	*Chef Service/ MINESUP*
16.	BILANDA Danielle Claude	Maître de Conférences	En poste
17.	DJIOGUE Séfirin	Maître de Conférences	En poste
18.	JATSA BOUKENG Hermine épse MEGAPTCHE	Maître de Conférences	En Poste
19.	LEKEUFACK FOLEFACK Guy B.	Maître de Conférences	En poste
20.	MBENOUN MASSE Paul Serge	Maître de Conférences	En poste
21.	MEGNEKOU Rosette	Maître de Conférences	En poste
22.	MONY Ruth épse NTONE	Maître de Conférences	En Poste
23.	NGUEGUIM TSOFACK Florence	Maître de Conférences	En poste
24.	NGUEMBOCK	Maître de Conférences	En poste
25.	TOMBI Jeannette	Maître de Conférences	En poste

26.	ATSAMO Albert Donatien	Chargé de Cours	En poste
27.	BASSOCK BAYIHA Etienne Didier	Chargé de Cours	En poste
28.	DONFACK Mireille	Chargée de Cours	En poste

29.	ESSAMA MBIDA Désirée Sandrine	Chargée de Cours	En poste
30.	ETEME ENAMA Serge	Chargé de Cours	En poste
31.	FEUGANG YOUMSSI François	Chargé de Cours	En poste
32.	GONWOUO NONO Legrand	Chargé de Cours	En poste
33.	GOUNOUE KAMKUMO Raceline	Chargée de Cours	En poste
34.	KANDEDA KAVAYE Antoine	Chargé de Cours	En poste
35.	KOGA MANG DOBARA	Chargé de Cours	En poste
36.	LEME BANOCK Lucie	Chargé de Cours	En poste
37.	MAHOB Raymond Joseph	Chargé de Cours	En poste
38.	METCHI DONFACK MIREILLE FLAURE EPSE GHOUMO	Chargé de Cours	En poste
39.	MOUNGANG Luciane Marlyse	Chargée de Cours	En poste
40.	MVEYO NDANKEU Yves Patrick	Chargé de Cours	En poste
41.	NGOUATEU KENFACK Omer Bébé	Chargé de Cours	En poste
42.	NJUA Clarisse Yafi	Chargée de Cours	*Chef Div. Uté Bamenda*
43.	NOAH EWOTI Olive Vivien	Chargée de Cours	En poste
44.	TADU Zephyrin	Chargé de Cours	En poste
45.	TAMSA ARFAO Antoine	Chargé de Cours	En poste
46.	YEDE	Chargé de Cours	En poste
47.	YOUNOUSSA LAME	Chargé de Cours	En poste

48.	AMBADA NDZENGUE GEORGIA ELNA	Assistante	En poste
49.	FOKAM Alvine Christelle Epse KEGNE	Assistante	En poste
50.	MAPON NSANGOU Indou	Assistant	En poste
51.	NWANE Philippe Bienvenu	Assistant	En poste

3- DÉPARTEMENT DE BIOLOGIE ET PHYSIOLOGIE VÉGÉTALES (BPV) (33)

1.	AMBANG Zachée	Professeur	*Chef DAARS /UYII*
2.	DJOCGOUE Pierre François	Professeur	En poste
3.	MBOLO Marie	Professeur	En poste
4.	MOSSEBO Dominique Claude	Professeur	En poste
5.	YOUMBI Emmanuel	Professeur	*Chef de Département*
6.	ZAPFACK Louis	Professeur	En poste

7.	ANGONI Hyacinthe	Maître de Conférences	En poste
8.	BIYE Elvire Hortense	Maître de Conférences	En poste
9.	MALA Armand William	Maître de Conférences	En poste
10.	MBARGA BINDZI Marie Alain	Maître de Conférences	*DAAC /UDla*
11.	NDONGO BEKOLO	Maître de Conférences	*CE / MINRESI*
12.	NGODO MELINGUI Jean Baptiste	Maître de Conférences	En poste
13.	NGONKEU MAGAPTCHE Eddy L.	Maître de Conférences	*CT / MINRESI*
14.	TONFACK Libert Brice	Maître de Conférences	En poste
15.	TSOATA Esaïe	Maître de Conférences	En poste
16.	ONANA JEAN MICHEL	Maître de Conférences	En poste

17.	DJEUANI Astride Carole	Chargé de Cours	En poste
18.	GOMANDJE Christelle	Chargée de Cours	En poste
19.	GONMADGE CHRISTELLE	Chargée de Cours	En poste
20.	MAFFO MAFFO Nicole Liliane	Chargé de Cours	En poste

21.	MAHBOU SOMO TOUKAM. Gabriel	Chargé de Cours	En poste
22.	NGALLE Hermine BILLE	Chargée de Cours	En poste
23.	NNANGA MEBENGA Ruth Laure	Chargé de Cours	En poste
24.	NOUKEU KOUAKAM Armelle	Chargé de Cours	En poste
25.	NSOM ZAMBO EPSE PIAL ANNIE CLAUDE	Chargé de Cours	*En détachement/UNESCO MALI*
26.	GODSWILL NTSOMBOH NTSEFONG	Chargé de Cours	En poste
27.	KABELONG BANAHO Louis-Paul-Roger	Chargé de Cours	En poste
28.	KONO Léon Dieudonné	Chargé de Cours	En poste
29.	LIBALAH Moses BAKONCK	Chargé de Cours	En poste
30.	LIKENG-LI-NGUE Benoit C	Chargé de Cours	En poste
31.	TAEDOUNG Evariste Hermann	Chargé de Cours	En poste
32.	TEMEGNE NONO Carine	Chargé de Cours	En poste
33.	MANGA NDJAGA JUDE	Assistant	En poste

4- DÉPARTEMENT DE CHIMIE INORGANIQUE (CI) (31)

1.	AGWARA ONDOH Moïse	Professeur	*Chef de Département*
2.	Florence UFI CHINJE épouse MELO	Professeur	*Recteur Univ.Ngaoundere*
3.	GHOGOMU Paul MINGO	Professeur	*Ministre Chargé deMiss.PR*
4.	NANSEU Njiki Charles Péguy	Professeur	En poste
5.	NDIFON Peter TEKE	Professeur	*CT MINRESI*
6.	NDIKONTAR Maurice KOR	Professeur	*Vice-Doyen Univ. Bamenda*
7.	NENWA Justin	Professeur	En poste
8.	NGAMENI Emmanuel	Professeur	*DOYEN FS Univ.Ngaoundere*
9.	NGOMO Horace MANGA	Professeur	*Vice Chancelor/UB*

10.	ACAYANKA Elie	Maître de Conférences	En poste
11.	EMADACK Alphonse	Maître de Conférences	En poste
12.	KAMGANG YOUBI Georges	Maître de Conférences	En poste
13.	KEMMEGNE MBOUGUEM Jean C.	Maître de Conférences	En poste
14.	KENNE DEDZO GUSTAVE	Maître de Conférences	En poste
15.	KONG SAKEO	Maître de Conférences	En poste
16.	MBEY Jean Aime	Maître de Conférences	En poste
17.	NDI NSAMI Julius	Maître de Conférences	En poste
18.	NEBAH Née NDOSIRI Bridget NDOYE	Maître de Conférences	*CT/ MINPROFF*
19.	NJIOMOU C. épse DJANGANG	Maître de Conférences	En poste
20.	NJOYA Dayirou	Maître de Conférences	En poste
21.	NYAMEN Linda Dyorisse	Maître de Conférences	En poste
22.	PABOUDAM GBAMBIE AWAWOU	Maître de Conférences	En poste
23.	TCHAKOUTE KOUAMO Hervé	Maître de Conférences	En poste

24.	BELIBI BELIBI Placide Désiré	Chargé de Cours	*Chef Service/ ENS Bertoua*
25.	CHEUMANI YONA Arnaud M.	Chargé de Cours	En poste
26.	KOUOTOU DAOUDA	Chargé de Cours	En poste
27.	MAKON Thomas Beauregard	Chargé de Cours	En poste
28.	NCHIMI NONO KATIA	Chargé de Cours	En poste
29.	NJANKWA NJABONG N. Eric	Chargé de Cours	En poste

30.	PATOUOSSA ISSOFA	Chargé de Cours	En poste
31.	SIEWE Jean Mermoz	Chargé de Cours	En Poste

5- DÉPARTEMENT DE CHIMIE ORGANIQUE (CO) (38)			
1.	DONGO Etienne	Professeur	*Vice-Doyen/FSE/UYI*
2.	NGOUELA Silvère Augustin	Professeur	*Chef de Département UDS*
3.	NYASSE Barthélemy	Professeur	En poste
4.	PEGNYEMB Dieudonné Emmanuel	Professeur	*Directeur/ MINESUP/ Chef de Département*
5.	WANDJI Jean	Professeur	En poste
6.	MBAZOA née DJAMA Céline	Professeur	En poste

7.	Alex de Théodore ATCHADE	Maître de Conférences	*Vice-Doyen / DPSAA*
8.	AMBASSA Pantaléon	Maître de Conférences	En poste
9.	EYONG Kenneth OBEN	Maître de Conférences	En poste
10.	FOLEFOC Gabriel NGOSONG	Maître de Conférences	En poste
11.	FOTSO WABO Ghislain	Maître de Conférences	En poste
12.	KAMTO Eutrophe Le Doux	Maître de Conférences	En poste
13.	KENMOGNE Marguerite	Maître de Conférences	En poste
14.	KEUMEDJIO Félix	Maître de Conférences	En poste
15.	KOUAM Jacques	Maître de Conférences	En poste
16.	MKOUNGA Pierre	Maître de Conférences	En poste
17.	MVOT AKAK CARINE	Maître de Conférences	En poste
18.	NGO MBING Joséphine	Maître de Conférences	*Chef de Cellule MINRESI*
19.	NGONO BIKOBO Dominique Serge	Maître de Conférences	*C.E.A/ MINESUP*
20.	NOTE LOUGBOT Olivier Placide	Maître de Conférences	*DAAC/Uté Bertoua*
21.	NOUNGOUE TCHAMO Diderot	Maître de Conférences	En poste
22.	TABOPDA KUATE Turibio	Maître de Conférences	En poste
23.	TAGATSING FOTSING Maurice	Maître de Conférences	En poste
24.	TCHOUANKEU Jean-Claude	Maître de Conférences	*Doyen /FS/ UYI*
25.	YANKEP Emmanuel	Maître de Conférences	En poste
26.	ZONDEGOUMBA Ernestine	Maître de Conférences	En poste

27.	NGNINTEDO Dominique	Chargé de Cours	En poste
28.	NGOMO Orléans	Chargée de Cours	En poste
29.	OUAHOUO WACHE Blandine M.	Chargée de Cours	En poste
30.	SIELINOU TEDJON Valérie	Chargé de Cours	En poste
31.	MESSI Angélique Nicolas	Chargé de Cours	En poste
32.	TCHAMGOUE Joseph	Chargé de Cours	En poste
33.	TSAMO TONTSA Armelle	Chargé de Cours	En poste
34.	TSEMEUGNE Joseph	Chargé de Cours	En poste

35.	MUNVERA MFIFEN Aristide	Assistant	En poste
36.	NONO NONO Éric Carly	Assistant	En poste
37.	OUETE NANTCHOUANG Judith Laure	Assistante	En poste
38.	TSAFFACK Maurice	Assistant	En poste

6- DÉPARTEMENT D'INFORMATIQUE (IN) (22)

1.	ATSA ETOUNDI Roger	Professeur	*Chef Div.MINESUP*
2.	FOUDA NDJODO Marcel Laurent	Professeur	*Chef Dpt ENS/Chef IGA.MINESUP*

3.	NDOUNDAM Réné	Maître de Conférences	En poste
4.	TSOPZE Norbert	Maître de Conférences	En poste

5.	ABESSOLO ALO'O Gislain	Chargé de Cours	*Sous-Directeur/MINFOPRA*
6.	AMINOU Halidou	Chargé de Cours	*Chef de Département*
7.	DJAM Xaviera YOUH - KIMBI	Chargé de Cours	En Poste
8.	DOMGA KOMGUEM Rodrigue	Chargé de Cours	En poste
9.	EBELE Serge Alain	Chargé de Cours	En poste
10.	HAMZA Adamou	Chargé de Cours	En poste
11.	JIOMEKONG AZANZI Fidel	Chargé de Cours	En poste
12.	KOUOKAM KOUOKAM E. A.	Chargé de Cours	En poste
13.	MELATAGIA YONTA Paulin	Chargé de Cours	En poste
14.	MONTHE DJIADEU Valery M.	Chargé de Cours	En poste
15.	OLE OLE Daniel Claude Delort	Chargé de Cours	*Directeur adjoint ENSET. Ebolowa*
16.	TAPAMO Hyppolite	Chargé de Cours	En poste

17.	BAYEM Jacques Narcisse	Assistant	En poste
18.	EKODECK Stéphane Gaël Raymond	Assistant	En poste
19.	MAKEMBE. S . Oswald	Assistant	En poste
20.	MESSI NGUELE Thomas	Assistant	En poste
21.	NKONDOCK. MI. BAHANACK.N.	Assistant	En poste
22.	NZEKON NZEKO'O ARMEL JACQUES	Assistant	En poste

7- DÉPARTEMENT DE MATHÉMATIQUES (MA) (31)

1.	AYISSI Raoult Domingo	Professeur	**Chef de Département**
2.	EMVUDU WONO Yves S.	Professeur	*Inspecteur MINESUP*

3.	KIANPI Maurice	Maître de Conférences	En poste
4.	MBANG Joseph	Maître de Conférences	En poste
5.	MBEHOU Mohamed	Maître de Conférences	En poste
6.	MBELE BIDIMA Martin Ledoux	Maître de Conférences	En poste
7.	NOUNDJEU Pierre	Maître de Conférences	*Chef Service des Programmes & Diplômes/FS/UYI*
8.	TAKAM SOH Patrice	Maître de Conférences	En poste
9.	TCHAPNDA NJABO Sophonie B.	Maître de Conférences	*Directeur/AIMS Rwanda*
10.	TCHOUNDJA Edgar Landry	Maître de Conférences	En poste

11.	AGHOUKENG JIOFACK Jean Gérard	Chargé de Cours	*Chef Cellule MINEPAT*
12.	BOGSO ANTOINE MARIE	Chargé de Cours	En poste
13.	CHENDJOU Gilbert	Chargé de Cours	En poste

14.	DJIADEU NGAHA Michel	Chargé de Cours	En poste
15.	DOUANLA YONTA Herman	Chargé de Cours	En poste
16.	KIKI Maxime Armand	Chargé de Cours	En poste
17.	MBAKOP Guy Merlin	Chargé de Cours	En poste
18.	MENGUE MENGUE David Joe	Chargé de Cours	*Chef Dpt /ENS Uté Maroua*
19.	NGUEFACK Bernard	Chargé de Cours	En poste
20.	NIMPA PEFOUKEU Romain	Chargée de Cours	En poste
21.	OGADOA AMASSAYOGA	Chargée de Cours	En poste
22.	POLA DOUNDOU Emmanuel	Chargé de Cours	*En stage*
23.	TCHEUTIA Daniel Duviol	Chargé de Cours	En poste
24.	TETSADJIO TCHILEPECK M. E.	Chargé de Cours	En poste

25.	BITYE MVONDO Esther Claudine	Assistante	En poste
26.	FOKAM Jean Marcel	Assistant	En poste
27.	LOUMNGAM KAMGA Victor	Assistant	En poste
28.	MBATAKOU Salomon Joseph	Assistant	En poste
29.	MBIAKOP Hilaire George	Assistant	En poste
30.	MEFENZA NOUNTU Thiery	Assistant	En poste
31.	TENKEU JEUFACK Yannick Léa	Assistant	En poste

8- DÉPARTEMENT DE MICROBIOLOGIE (MIB) (22)

1.	ESSIA NGANG Jean Justin	Professeur	*Chef de Département*
2.	NYEGUE Maximilienne Ascension	Professeur	*VICE-DOYEN / DSSE/FS/UYI*
3.	NWAGA Dieudonné M.	Professeur	En poste

4.	ASSAM ASSAM Jean Paul	Maître de Conférences	En poste
5.	BOUGNOM Blaise Pascal	Maître de Conférences	En poste
6.	BOYOMO ONANA	Maître de Conférences	En poste
7.	KOUITCHEU MABEKU Epse KOUAM Laure Brigitte	Maître de Conférences	En poste
8.	RIWOM Sara Honorine	Maître de Conférences	En poste
9.	SADO KAMDEM Sylvain Leroy	Maître de Conférences	En poste

10.	BODA Maurice	Chargé de Cours	En position d'absence irrégulière
11.	ESSONO OBOUGOU Germain G.	Chargé de Cours	En poste
12.	NJIKI BIKOÏ Jacky	Chargée de Cours	En poste
13.	TCHIKOUA Roger	Chargé de Cours	En poste
14.	ESSONO Damien Marie	Chargé de Cours	En poste
15.	LAMYE Glory MOH	Chargé de Cours	En poste
16.	MEYIN A EBONG Solange	Chargée de Cours	En poste
17.	NKOUDOU ZE Nardis	Chargé de Cours	En poste
18.	TAMATCHO KWEYANG Blandine Pulchérie	Chargée de Cours	En poste
19.	TOBOLBAÏ Richard	Chargé de Cours	En poste

20.	MONI NDEDI Esther Del Florence	Assistante	En poste
21.	NKOUE TONG ABRAHAM	Assistant	En poste
22.	SAKE NGANE Carole Stéphanie	Assistante	En poste

9. DEPARTEMENT DE PYSIQUE(PHY) (43)

1.	BEN- BOLIE Germain Hubert	Professeur	En poste
2.	DJUIDJE KENMOE épouse ALOYEM	Professeur	En poste
3.	EKOBENA FOUDA Henri Paul	Professeur	*Vice-Recteur. Uté Ngaoundéré*
4.	ESSIMBI ZOBO Bernard	Professeur	En poste
5.	NANA ENGO Serge Guy	Professeur	En poste
6.	NANA NBENDJO Blaise	Professeur	En poste
7.	NDJAKA Jean Marie Bienvenu	Professeur	*Chef de Département*
8.	NJANDJOCK NOUCK Philippe	Professeur	En poste
9.	NOUAYOU Robert	Professeur	En poste
10.	PEMHA Elkana	Professeur	En poste
11.	SAIDOU	Professeur	*Chef de centre/IRGM/MINRESI*
12.	TABOD Charles TABOD	Professeur	*Doyen FSUniv/Bda*
13.	TCHAWOUA Clément	Professeur	En poste
14.	WOAFO Paul	Professeur	En poste
15.	ZEKENG Serge Sylvain	Professeur	En poste

16.	BIYA MOTTO Frédéric	Maître de Conférences	*DG/HYDRO Mekin*
17.	BODO Bertrand	Maître de Conférences	En poste
18.	ENYEGUE A NYAM épse BELINGA	Maître de Conférences	En poste
19.	EYEBE FOUDA Jean sire	Maître de Conférences	En poste
20.	FEWO Serge Ibraïd	Maître de Conférences	En poste
21.	HONA Jacques	Maître de Conférences	En poste
22.	MBINACK Clément	Maître de Conférences	En poste
23.	MBONO SAMBA Yves Christian U.	Maître de Conférences	En poste
24.	NDOP Joseph	Maître de Conférences	En poste
25.	SIEWE SIEWE Martin	Maître de Conférences	En poste
26.	SIMO Elie	Maître de Conférences	En poste
27.	VONDOU DerbetiniAppolinaire	Maître de Conférences	En poste
28.	WAKATA née BEYA Annie	Maître de Conférences	*Directeur/ENS/UYI*

29.	ABDOURAHIMI	Chargé de Cours	En poste
30.	CHAMANI Roméo	Chargé de Cours	En poste
31.	EDONGUE HERVAIS	Chargé de Cours	En poste
32.	FOUEDJIO David	Chargé de Cours	*Chef Cell. MINADER*
33.	MELI'I Joelle Larissa	Chargée de Cours	En poste
34.	MVOGO ALAIN	Chargé de Cours	En poste
35.	WOULACHE Rosalie Laure	Chargée de Cours	*Absente depuis Janvier 2022*
36.	AYISSI EYEBE Guy François Valérie	Chargé de Cours	En poste
37.	DJIOTANG TCHOTCHOU Lucie Angennes	Chargée de Cours	En poste
38.	OTTOU ABE Martin Thierry	Chargé de Cours	En poste
39.	TEYOU NGOUPOU Ariel	Chargé de Cours	En poste

40.	KAMENI NEMATCHOUA Modeste	Assistant	En poste
41.	LAMARA Maurice	Assistant	En poste
42.	NGA ONGODO Dieudonné	Assistant	En poste
43.	WANDJI NYAMSI William	Assistant	En poste

10- DÉPARTEMENT DE SCIENCES DE LA TERRE (ST) (42)

1.	BITOM Dieudonné-Lucien	Professeur	*Doyen / FASA / UDs*
2.	FOUATEU Rose épse YONGUE	Professeur	En poste
3.	NDAM NGOUPAYOU Jules-Remy	Professeur	En poste
4.	NDJIGUI Paul Désiré	Professeur	*Chef de Département*
5.	NGOS III Simon	Professeur	En poste
6.	NKOUMBOU Charles	Professeur	En poste
7.	NZENTI Jean-Paul	Professeur	En poste

8.	ABOSSOLO née ANGUE Monique	Maître de Conférences	*Vice-Doyen / DRC*
9.	BISSO Dieudonné	Maître de Conférences	*Directeur/Projet Barrage Memve'ele*
10.	EKOMANE Emile	Maître de Conférences	En poste
11.	FUH Calistus Gentry	Maître de Conférences	*Sec. D'Etat/MINMIDT*
12.	GANNO Sylvestre	Maître de Conférences	En poste
13.	GHOGOMU Richard TANWI	Maître de Conférences	*Chef de Département /Uté Maroua*
14.	MOUNDI Amidou	Maître de Conférences	CT/ MINIMDT
15.	NGO BIDJECK Louise Marie	Maître de Conférences	En poste
16.	NGUEUTCHOUA Gabriel	Maître de Conférences	CEA/MINRESI
17.	NJILAH Isaac KONFOR	Maître de Conférences	En poste
18.	NYECK Bruno	Maître de Conférences	En poste
19.	ONANA Vincent Laurent	Maître de Conférences	*Chef service Maintenance & du Matériel/UYII*
20.	TCHAKOUNTE J. épse NUMBEM	Maître de Conférences	*Chef.cell / MINRESI*
21.	TCHOUANKOUE Jean-Pierre	Maître de Conférences	En poste
22.	TEMGA Jean Pierre	Maître de Conférences	En poste
23.	YENE ATANGANA Joseph Q.	Maître de Conférences	*Chef Div. /MINTP*
24.	ZO'O ZAME Philémon	Maître de Conférences	*DG/ART*

25.	ANABA ONANA Achille Basile	Chargé de Cours	En poste
26.	BEKOA Etienne	Chargé de Cours	En poste
27.	ELISE SABABA	Chargé de Cours	En poste
28.	ESSONO Jean	Chargé de Cours	En poste
29.	EYONG JOHN TAKEM	Chargé de Cours	En poste
30.	MAMDEM TAMTO LIONELLE ESTELLE	Chargé de Cours	En poste
31.	MBESSE CECILE OLIVE	Chargée de Cours	En poste
32.	MBIDA YEM	Chargé de Cours	En poste
33.	METANG Victor	Chargé de Cours	En poste
34.	MINYEM Dieudonné	Chargé de Cours	*CD/ Uté Maroua*
35.	NGO BELNOUN Rose Noël	Chargée de Cours	En poste
36.	NOMO NEGUE Emmanuel	Chargé de Cours	En poste
37.	NTSAMA ATANGANA Jacqueline	Chargé de Cours	En poste
38.	TCHAPTCHET TCHATO De P.	Chargé de Cours	En poste
39.	TEHNA Nathanaël	Chargé de Cours	En poste
40.	FEUMBA Roger	Chargé de Cours	En poste
41.	MBANGA NYOBE Jules	Chargé de Cours	En poste

42.	NGO'O ZE ARNAUD	Assistant	En poste

Répartition chiffrée des Enseignants de la Faculté des Sciences de l'Université de Yaoundé I

NOMBRE D'ENSEIGNANTS					
DÉPARTEMENT	**Professeurs**	**Maîtres de Conférences**	**Chargés de Cours**	**Assistants**	**Total**
BCH	8 (00)	14 (10)	15 (05)	02 (01)	**39 (16)**
BPA	14 (01)	11 (07)	22 (07)	04 (02)	**51 (17)**
BPV	06 (01)	10(01)	16 (09)	01 (00)	**33 (11)**
CI	09(01)	14(04)	08 (01)	00 (00)	**31 (06)**
CO	06 (01)	20 (04)	08 (03)	04 (01)	**38(09)**
IN	02 (00)	02 (00)	12 (01)	06 (00)	**22 (01)**
MAT	02 (00)	08 (00)	14 (01)	07 (01)	**31 (02)**
MIB	03 (01)	06 (02)	10 (03)	03 (02)	**22 (08)**
PHY	15 (01)	13 (02)	11 (03)	04 (00)	**43 (06)**
ST	07 (01)	16 (03)	18 (04)	01 (00)	**42(08)**
Total	**72 (07)**	**114 (33)**	**134 (37)**	**32 (07)**	**352 (84)**

Soit un total de **352 (84)** dont :

- Professeurs **72 (07)**
- Maîtres de Conférences **114 (33)**
- Chargés de Cours **134 (37)**
- Assistants **32 (07)**

( ) = Nombre de Femmes **84**

DEDICATION

I dedicate this work to my lovely wife, Elodie
and
my children Owen and Helena.

Acknowledgements

I am very grateful to **God** for his support in my life.

I am also very grateful to my supervisors, Pr. **Emmanuel Fouotsa** and Pr. **Célestin Nkuimi Jugnia**, for their outstanding and invaluable support towards the success of this research.

I would like to thank all the members of the Research Team in Cryptography (Mathematics and Cryptology Online Seminars) and the University of Yaounde I Laboratory of Algebra and Geometry (LAGA) of the Faculty of science, for their various recommendations during research seminars. Many thanks in particular to the coordinator Pr. **Etienne Temgoua**, and also to, Pr. **Daniel Tieudjo**, Pr. **Célestin Lele**, Pr. **Selestin Ndjeya**, Pr. **Marcel Tonga**, Pr. **Maurice Kianpi**, Dr. **Romain Nimpa**, Dr. **Ogadoa**, Dr. **Michel Djiadeu**, Dr. **Salomon Mbatakou**, Dr. **Surdive Atamewoue**, Dr. **Hilaire Mbiakop**, Dr. **Alexandre Fotue**, Dr. **Rostand kuitche**, Dr. **Hervé Talé** and Dr. **Yannick Tenkeu**. This research would not have been possible without their support and inspiration.

I also thank all members of the department of mathematics of the Faculty of science in the University of Yaoundé I for their help in various ways throughout my education.

I am greatly indebted to my father Mr. **Timothée Tchinda** and my mothers Mrs. **Hélène Tchinda** and **Madelène Tchinda** to have given me an adequate moral and material support. I also thank Mr. **Raoul Jepang**, Mrs. **Marie Laning**, Mr. **Serge Meni**, Mrs. **Élise Tchoffo**, and Mrs. **Camille Jepang** for their encouragement and various supports.

My many thanks to my dear wife **Élodie Nandjo Tappa**, my brothers and my sisters(**Céline**, **Francine**, **Carine**, **Guy**, **Gladice**, **Herman**, **Brice**, **Inès**, **Cristèle**, **Thierry**, **Vanessa** and **Lynda**) for their various support during my studies.

I do not forget my old friends, **Saturin**, **Carlos**, **Nadia**, **Ronald**, **Françis**, **Patrick**, **Gildas**, **Érik**, **Brunhilda**, **Terence**, **Laurian**, **Joseph**, **Giresse**, **Junior**, **Olivia**, **Ruth**, **Augustin**, **Rose**, **Aline** and **Serge** just to mention a few. Be sure of my friendship.

Contents

Abstract

Euclidean lattice-based cryptography originated in the $1990's$ with Miklos Ajtai where he demonstrated that Euclidean lattices can serve as a basis for cryptography. The security of lattice-based cryptosystems is based on the presumed hardness of lattice problems such as closest and shortest vector problems. Lattice-based cryptography is growing rapidly today: its potential effectiveness, its apparent resistance to quantum attacks, and above all its proofs of security under very precise hypotheses of algorithmic difficulties of fairly well understood problems.

Although the Shortest Vector and Closest Vector Problems are difficult for Euclidean lattices, there are some families of lattices for which these problems are efficiently solvable. We have for example integer lattice $\mathbb{Z}^n$, root lattices A_n $(n \geq 1)$, D_n $(n \geq 2)$, E_6, E_7, E_8, their duals, and the $A_n \otimes A_m$ $(n, m \geq 1)$.

In this thesis we propose a polynomial algorithm for solving the closest vector problem in the root lattice $D_n \otimes D_m$ $(n, m \geq 2)$.

We also consider the root lattice $A_{n1} \otimes ... \otimes A_{nk}$ $(n1, ..., nk \geq 1)$ for which we propose a polynomial algorithm for solving the Closest Vector Problem. This was successful using the associativity of lattices and non commutativity of tensor product.

Furthermore, Sieving algorithms have been very efficient in solving some extended insistences of Shortest Vector Problem. In this thesis, we use the famous LLL-reduction algorithm and the symmetries of lattices to give a new Sieve algorithm for orthogonal integer lattice $\Lambda \subset \mathbb{Z}^n$. Lattice-based cryptography going rapidly today thanks to its potential effectively. All over this work, we have successfully implemented all the algorithms in the Maple computer software 18.0.

Key Words : orthogonal integer lattice, closest vector problem, shortest vector problem, Sieve algorithm, LLL algorithm, .

Résumé

La Cryptographie basée sur les réseaux euclidiens est née dans les années 1990 avec Miklos Ajtai où il démontre que les réseaux euclidiens peuvent servir de base solide à la cryptographie. La sécurité des cryptosystèmes basés sur les réseaux est basée sur la difficulté des problèmes du réseau tels que, les problèmes du vecteur le plus proche, et du vecteur le plus court. La cryptographie basé sur les réseaux connait aujourd'hui un essor rapide: son apparente résistance aux attaques quantiques, et surtout ses preuves de sécurité sous des hypothèses très précises de difficultés algorithmique de problèmes assez bien compris.

Bien que les problèmes du vecteur le plus court et du vecteur le plus proche cités plus haut soient difficiles pour les réseaux, il existe certaines familles de réseaux pour lesquelles ces problèmes sont solubles en utilisant un algorithme polynomial. Nous avons par exemple les réseaux entiers $\mathbb{Z}^n$, les réseaux de racine A_n $(n \geq 1)$, D_n $(n \geq 2)$, E_6, E_7, E_8, leurs duaux, et $A_n \otimes A_m$, $(n, m \geq 1)$. Dans cette thèse nous proposons un algorithme polynomial de résolution du problème du vecteur le plus proche dans le réseaux $D_n \otimes D_m$ $(n, m \geq 2)$.

Nous considérons également le réseau $A_{n1} \otimes ... \otimes A_{nk}$ $(n1, ..., nk \geq 1)$ pour lequel nous proposons un algorithme polynomial de résolution du problème du vecteur le plus proche. Cela a été fait en utilisant l'associativité des réseaux et la non commutativité du produit tensoriel.

De plus, les algorithmes de crible ont été très efficaces pour résoudre certaines instances étendues du Problème du Vecteur le plus Court. Dans cette thèse, nous utilisons le fameux algorithme de réduction LLL et le symétrie des réseaux pour proposer un nouvel algorithme de crible pour les réseaux entier orthogonaux $\Lambda \subset \mathbb{Z}^n$. La cryptographie basée sur les réseaux Euclidiens progresse rapidement aujourd'hui grâce à son efficacité. Tout au long de ce travail, nous avons réussi à implémenter tous les algorithmes avec le logiciel informatique Maple 18.0.

Mots clés : réseaux orthogonaux, problème du vecteur le plus court, problème du vecteur le plus proche, algorithme de crible, algorithme LLL.

List of Figures

List of Algorithms

Chapter One

INTRODUCTION

1.1 Context and Motivation

1.1.1 Context

The central purpose of cryptography is to allow two peoples, traditionally called Alice and Bob, to communicate through a secure channel such that a passive opponent Oscar cannot alter or manipulate the information. Cryptanalysis is the art for an unauthorized person to decrypt, decode, decipher a message. It is therefore the set of methods for attacking a cryptographic system. Cryptology is the combination of cryptography and cryptanalysis. Cryptography focuses on four different objectives:

• **Confidentiality:** Confidentiality ensures that only the intended recipient can decrypt the message and reads its contents.

• **Integrity:** Integrity focuses on the ability to be certain that the information contained within the message cannot be modified while in storage or transit.

• **Non-repudiation:** Non-repudiation means the sender of the message cannot backtrack in the future and deny their reasons for sending or creating the message.

• **Authenticity:** Authenticity ensures the sender and recipient can verify each other's identities and the destination of the message.

These objectives help ensure a secure and authentic transfer of information.
Based on the number of keys that are employed for encryption and decryption, there are two types of cryptography: secret key (symmetric) cryptography and public key (asymmetric) cryptography. With secret key cryptography, the same key is used for both encryption and decryption. A sender and a recipient must already have a shared key. Key distribution is then a tricky problem as was the motivation for developing public key cryptography.

With public key cryptography, two different keys are used for encryption and decryption. Every user in an asymmetric key cryptosystsem has both a public key and private key. The private key is kept secret at all times, but the public key may be freely distributed and it won't affect security (unlike sharing the key in a symmetric cryptosystem).

A revolution in cryptography came along with the discovery of public-key encryption, where only the receiver of messages needs to be in possession of the secret key, while a sender just needs to know the public key of the receiver. The discovery of public-key cryptography is usually attributed to Diffie and Hellman[17], with Rivest, Shamir and Adleman providing the first implementation[47].

The security of public key cryptographic algorithms is based on mathematical problems that are hard to solve:

Discrete Logarithm Problem (DLP): Let $G =< g >$ be a cyclic group of order n with generator g and h an element of G, find $x \in \{1, ..., n\}$ such that $h = g^x$. Note that the integer x is uniquely determined modulo the group order. Just as for the continuous logarithm function, one also writes $x = \log_g h$ and refers to x as the discrete logarithm of h to the base g. Discrete logarithm problem is hard on group embedded in a finite extension field and on a group of points of ordinary elliptic curves[26, 27]. Some protocols based on discrete logarithm are Diffie-Hellman key exchange protocol [17], ElGamal cryptosystem [43]

Factorization Problem: Let N be the product of two large prime numbers p and q of roughly the same size, e and d two integers such that:
$ed \equiv 1(\mod \varphi(N))$. Given the public key (N, e) and the cipher text y, it is difficult to find d to obtain the plain text x such that $y \equiv x^e \mod N$. The most known method to solve the factorization (RSA) problem is factoring the modulus N. This task is impractical if N is sufficiently large [47, 8].

Classical lattice problems: The first fundamental hard problem in lattice is the Shortest Vector Problem (SVP). Given a basis for a lattice, the problem is to find a non-zero vector in the lattice whose length is minimal over all non-zero lattice vectors. This problem is NP-hard for randomized reductions. Note that the shortest vector problem in a lattice is not unique. [1, 3, 4, 18, 24].

The Closest Vector Problem (CVP) is the generalization of the Shortest Vector Problem. In this problem one is given a lattice defined by some basis as well as a target vector in the ambient vector space in which the lattice lies, the task is to determine a vector in the lattice which is close to the target vector [1, 3, 4, 18, 24, 49, 50]. Encryption an decryption of the GGH, NTRU cryptosystems are based on the Closest Vector Problem.

1.1.2 Motivation

Today, cryptography is used in a large number of products. It is thus found in electronic votes, payment by bank cards, electronic mail, databases, smart cards, digital decoders, electronic purchases. Unfortunately, quantum computers can make their security vulnerable. The reason quantum cryptography can do this is that, with a powerful enough computer, algorithms that would usually take 10 years to crack could now take only weeks or days with quantum computer. Indeed, in [48], Peter Shor proposes a polynomial time algorithm running on a quantum computer which solves both of factoring and discrete logarithm problems. Now the physicists have actually not been able to build a large quantum computer yet, and the complete breakdown of most cryptography used today is probably not right around the corner.

The United States is preparing new encryption standards that even the National Security Agency (NAS) will not be able to crack, specifies the Director cyber security of the National Aeronautics and Space Administration (NASA). These new standards are intended to resist quantum computer, which could potentially compromise public-key cryptographic algorithms. In December 2016, the National Institute of Standards and Technology (NIST) announced an international competition, selected 7 finalist from the 69 initial submissions. After careful consideration during $3rd$ Round of the NIST post quantum standardization process, NIST has identified 4 candidate algorithms for standardization, as well as those that will continue to be evaluated in a fourth round of analysis. The public-key encryption and key-establishment algorithm that will be standardized are CRYSTALS-Dilithium, FALCON, and SPHINCS+. While there are multiple signature algorithms selected, NIST recommends CRYSTAL-Dilithium as the primary algorithm to be implemented. In 2018, Léo Ducas and al. presented a new Digital Signature Scheme DILITHIUM whose security is based on the hardness of finding short vectors in lattices [20]. The most compact lattice-based signature schemes [19, 21] crucially require the generation of secret randomness from the discrete Gaussian distribution. Generating such samples in a way that is secure against side-channel attacks is highly non trivial and can easily lead to insecure implementations, as demonstrated in [9, 44]. DILITHIUM uses uniform Sampling, as was originally proposed in [35, 28]. In addition, four of the alternative key-establishment candidate algorithms will advance to a fourth round evaluation: BIKE (Bit Flipping Key Encapsulation), classic McEliece, HQC (Hamming Quasi-Cyclic), and SIKE (Super singular Isogeny Key Encapsulation). These candidates are still being considered for future standardization

($https://doi.org/10.6028/NIST.IR.8413$). The goal of this competition is to make the algorithms available in 2024 so that government and industries can adopt them.
To avoid an economics war, it is imperative to set up new cryptosystems that will be resistant to these quantum computers. It is therefore judicious to seek among the mathematical tools, those which present hard problems, which can be used for cryptography. We list error correcting codes, isogenies and Euclidean lattices. In this thesis, we will focus our attention only on Euclidean lattices. Indeed, lattice based cryptographic constructions hold a great promise for post-quantum cryptography, as they enjoy very strong security proofs based on worst-case hardness, relatively efficient implementations, as well as great simplicity. In addition, lattice based cryptography is believed to be secure against quantum computers.

Euclidean lattices are the regular arrangements of points in space, or more precisely, the discrete subgroups of $\mathbb{R}^n$ for some positive integer n. In 1982, Arjen Lenstra, Hendrick Lenstra and Lásló Lovàsz developed a polynomial algorithm for lattice reduction [11, 16, 46, 50, 49]. This algorithm known under the name LLL, coming from its authors names, constituted a real revolution of lattice theory. First of all, its complexity is without comparison with the algorithms described until then to study Euclidean lattices, and, above all, it has opened the way to an impressive number of applications. The three historical applications are the factorization of polynomials with integer or rational cœfficients, simultaneous rational approximations [33] and integer programming in fixed dimension[30]. This algorithm also received immediate success in the field of cryptanalysis. In particular, it was used by Lagarias and Odlyzko[32, 34] to break the knapsack's cryptosystem proposed by Merkle and Hellman. The algorithm due to Lenstra, Lenstra and Lovàsz is still a very popular tool in the cryptanalysis of public key cryptosystems, such as some fast variants of RSA [6, 8, 38], and some fast variants of the DSA signature scheme[29, 42]. The LLL algorithm has also shown that finding the private exponent of a RSA key is computationally equivalent to factoring the modulus[39]. Indeed, it makes it possible to construct a deterministic polynomial reduction. Another important field of application of the LLL algorithm is the algorithmic theory of numbers: it has made it possible to invalidate the conjecture of Mertens, and is also used to calculate minimal polynomials of algebraic numbers for example, or to work in the field of numbers[46].

1.2 Problematic

A central problem in the theory of lattices is the Closest Vector Problem (CVP). It is often seen as one of the hardest computational lattice problems as many lattices problems polynomially reduce to it. We can point as example, the Shortest Vector Problem (SVP)[25], and more generally, finding all successive minima of lattice[45]. Furthermore it was already proven in 1981 that for general lattices, CVP was NP-hard under deterministic reductions. In 1998, SVP was proven to be NP-hard under randomized reductions[4]. A deterministic reduction that SVP is NP-hard has not been discovered yet. Although the CVP is an NP-hard problem for general lattices, it is interesting to design lattices for which CVP can be solved efficiently while at the same time optimizing other lattices properties like the packing density. Special lattices are for example the root lattices A_n $(n \geq 1)$, D_n $(n \geq 2)$, E_n $(n = 6, 7, 8)$, their duals and the Leech lattice [13, 22]. These lattices can be used as the basis for efficient block quantizers for uniformly distributed inputs and to construct code for a band-limited channel with Gaussian noise [23, 13]. Indeed, recent attempts to create lattice-based cryptographic schemes are promising and are mostly based on removing some error to a lattice vector using a CVP algorithm [36, 37]. Léo Ducas and Wessel van Woerden proposed a polynomial algorithm for solving CVP for the case of the lattice $A_n \otimes A_m$ $(n, m \geq 1)$ in order to give a generalization of resolution of CVP on some case of cyclotomic integer lattices $\mathbb{Z}[\zeta_\alpha]$ (with $\alpha = p.q$, where p and q are prime) and their duals [22]. SVP has been extensively studied as purely mathematical problem, being central in the study of the geometry of numbers and as algorithm problems, having many applications in communication theory and computer science. There are two main algorithmic techniques for solving exact SVP: enumeration and sieving. Enumeration algorithms were initiated by Pohst [45] in 1981 and one of the best enumeration algorithm was given by Kannan in 1983 [31]. This method runs in $n^{o(n)}$ time but is polynomial in space. The main idea of sieve algorithms is to randomly select lattice vectors, then compare them in order to end up getting the shortest lattice vectors, running the algorithm for many steps. This method was introduced by Ajtai, Kumar and Sivakumar in 2001 [5], lowering the time complexity of the SVP to $2^{o(n)}$, but required $2^{o(n)}$ space and randomness. In 2010, Micciancio et al. presented GaussSieve [41], the first sieving heuristic that outperformed enumeration routines. In 2011, Panagiotis proposed a new heuristic sieving algorithm [50] that performed quite well in the practice with estimated running time $2^{0,52n}$ and space complexity $2^{0,2n}$. In 2017, Leo Ducas [18] exploits the fact that *sieving* returns many short vectors, rather than only one to propose a new practical improvement for sieve algorithms. The questions below are problems that have interested us throughout our thesis.

Question 1: Find a polynomial time algorithm to solve the Closest Vector Problem in tensor product of three root lattices of type A ($A_n \otimes A_m \otimes A_p$; $n, m, p \geq 1$), and in the general case of tensor product of a finite number k of root lattices of type A ($A_{n_1} \otimes ... \otimes A_{n_k}$; $n_1, ..., n_k \geq 1$).

Question 2: Find a polynomial time algorithm to solve the Closest Vector Problem in two root lattices of type D ($D_n \otimes D_m$; $n, m \geq 2$).

Question 3: Give sieve algorithm for the case of orthogonal integer lattice of dimension n.

1.3 Contributions

1.3.1 Research Objectives

The objectives of this thesis consist to respond to questions 1, 2 and 3. The answers of questions 1 and 2 will help to solve the Closest Vector Problem in the general case of cyclotomic integer rings. The answer of question 3 will help to solve the Shortest Independent Vector Problem in some orthogonal integer lattice. These results will allow to extend the families of lattices that should not be used for post quantum signature schemes based on lattices.

1.3.2 Results obtained and Methodology

The main contributions of this thesis are presented as follows:

1. We use the associativity of lattices and non commutativity of tensor product to give a polynomial algorithm allowing to solve the Closest Vector Problem in the tensor product of three root lattices of type A ($A_n \otimes A_m \otimes A_p$; $n, m, p \geq 1$), and give a polynomial algorithm for the case of tensor product of a finite number k of root lattices of type A ($A_{n1} \otimes ... \otimes A_{nk}$; $n1, ..., nk \geq 1$). This efficient algorithm performs with $O(d.(((n+1)(m+1)-1)p)^2 min\{(n+1)(m+1)-1), p\})$ arithmetic operations.

2. We established that the root lattice D_{nm} is a full rank sub-lattice of the tensor product $D_n \otimes D_m$ ($n, m \geq 2$) of the root lattices D_n and D_m. This allows to provide efficient algorithm for solving the Closest Vector Problem in $D_n \otimes D_m$ ($n, m \geq 2$) by using the same method for the case

of root lattice D_n. The proposed algorithm performs at most $O(n+m)$ arithmetic operations.

3. We use the famous LLL-reduction algorithm and the symmetries of lattices to give a new sieve algorithm that we called OrthogonalSieve algorithm. This algorithm gives at least n and at most 2^n short vectors in general case of orthogonal integer lattice $\Lambda \subset \mathbb{Z}^n$. This algorithm runs in $O(n2^n)$ time and can be polynomial in space and the list of short vectors obtained enables to solve the Shortest Independent Vector Problem (SIVP) [7] for some orthogonal integer lattices. We also give an algorithm for the particular case of integer lattice $\mathbb{Z}^n$. Indeed, for the particular lattices $\Lambda \subset \mathbb{Z}^n$, A_n and D_n, we respectively have $2n$, $n(n+1)$ and $2n(n-1)$ short vectors.

The above results consist of the following publications:

1. Arnaud Girès Fobasso Tchinda, Emmanuel Fouotsa, Celestin Nkuimi Jugnia, Sieve Algorithms for Some Orthogonal Integer Lattices, Discrete Mathematics, Algorithms and Applications, (2022) https://doi.org/10.1142/S179383022501518.

2. Arnaud Girès Fobasso Tchinda, Emmanuel Fouotsa and Celestin Nkuimi Jugnia, A Polynomial Algorithm for Solving the Closest Vector Problem in Tensored Root Lattices of Type D, *SN Computer Science, Springer* (2022) https://doi.org/10.1007s42979-022-01440-2.

3. Arnaud Girès Fobasso Tchinda, Emmanuel Fouotsa, Celestin Nkuimi Jugnia, Generalization of Closest Vector Problem in Tensored Root Lattices of Type A. Under review at *Indian Journal of Pure and Applied Mathematics, Springer.*

1.4 Organization of the Thesis

Besides this introduction, the thesis contains three chapters. The last two chapters start with an introduction followed by the main results of the chapter. Then, these chapters conclude with remarks that summarize the results of the chapter and address further works. We end the thesis by giving some general conclusions which summarize the results of the thesis and also address the most interesting further work.

Chapter 2 is a survey of the lattice background and some basic definitions and results on lattice reduction. Closest Vector Problem, Shortest Vector Problem, Sieve algorithm and some lattice reductions were discussed.

In **Chapter 3**, we present the polynomial algorithms for solving the Closest Vector Problem for the case of tensor product of a finite root lattices of type A_n $(n \geq 1)$, and for tensor product of two root lattices of type D_n $(n \geq 2)$.

In **Chapter 4**, we give a list of all short vectors of the particular case of orthogonal integer lattices $\mathbb{Z}^n$. We also propose an enumeration algorithm which will allow us to obtain the list of shortest vectors in all orthogonal integer lattices $\Lambda \subseteq \mathbb{Z}^n$.

For correctness, a Maple computer software implementation of the algorithm has been done.

Conclusion: It contains a summary of the main results from the research conducted. There is also a discussion of future work to be carried out on algorithms for solving Closest Vector Problems for tensor product of a finite root lattices of type D_n $(n \geq 2)$; and giving an algorithm which will give a list of short vector in general case of any orthogonal lattice.

Chapter Two

General preliminaries on lattices

In this chapter, we will give an introduction to lattices and the different concepts used in lattice-based cryptography. It should serve as a starting point for reading the following chapters, as well as giving a general introduction to some of the concepts used in the area. We start with an introduction of lattices in Section 2.1. In Section 2.2, we give some invariants of Euclidean lattices and the algorithmic problems related to them. Some of these invariants are easy to evaluate, and the notion of reduction makes it possible to obtain information on invariants that are difficult to calculate from those that are easy to evaluate. This is studied in Section 2.3 as well as the Gaussian heuristic. Before concluding this chapter, we will talk about lattice problems in Section 2.4. The result announced in this chapter come mainly from [2, 4, 11, 13, 18, 24, 41, 45, 49].
Throughout this work, for any positive integer n, we use the Euclidean inner product on $\mathbb{R}^n$ that is defined by: $\langle \mathbf{x} \,,\, \mathbf{y} \rangle := x_1y_1 + x_2y_2 + \cdots + x_ny_n$ for $\mathbf{x} := (x_1, x_2, \cdots, x_n)$ and $\mathbf{y} := (y_1, y_2, \cdots, y_n)$ in $\mathbb{R}^n$. The Euclidean norm on $\mathbb{R}^n$ is defined as follows: $||\mathbf{x}|| := \sqrt{\langle \mathbf{x} \,,\, \mathbf{x} \rangle}$.
We denote by $\mathcal{B}(x, R)$ the closed Euclidean $n-$dimensional ball of radius R centered at x, that is : $\mathcal{B}(x, R) = \{y \in \mathbb{R} : \ \|x - y\| < R\}$. If no center is specified, then the center is zero; $\mathcal{B}(R) := \mathcal{B}(O, R)$.

2.1 Lattices

A lattice in $\mathbb{R}^n$ is a set of points with a periodic structure. More formally, it can be viewed as a discrete additive subgroup of $\mathbb{R}^n$. We give the following definition, for which an example can be seen in Figure 2.1, Figure 2.2, Figure 2.3 and Figure 2.4.

Definition 2.1.1. *[24] Given a set $B = \{b_1, ..., b_d\}$ of d linearly independent vectors in $\mathbb{R}^n$, we can define the lattice $\Lambda(B)$ as the set of all integer linear* *combinations of these vectors. That is*

$$\Lambda(B) = \left\{ \sum_{i=1}^{d} z_i \boldsymbol{b}_i \ : \ (z_1, z_2, \cdots, z_d) \in \mathbb{Z}^d \right\}. \tag{2.1}$$

We say that B forms a basis for $\Lambda(B)$, and the integers n and d the dimension and the rank of the lattice, respectively. Indeed, The rank of a lattice Λ is defined as the number of linearly independent vector in any basis for that lattice. A lattice Λ that is full-rank is defined as a lattice where the number of linearly independent vectors in any basis for this lattice is equal to the dimension of the lattice. This means that if $d = n$, then the lattice is called full-rank lattice.

In this definition, it is an implicit requirement that $n \geq d$. This will always be the assumption, unless something else is explicitly specified. A more compact and convenient way of writing the definition of $\Lambda(B)$, is to consider B as a matrix in $\mathbb{R}^{n\times d}$ with $b_1, ..., b_d$ as columns. Using this matrix, we can also write $\Lambda(B)$ as:

$$\Lambda(B) = \left\{ Bx \ : \ x \in \mathbb{Z}^d \right\}. \tag{2.2}$$

The basis of a lattice is not necessarily unique, in fact most lattices will have an infinite number of different bases. Given a basis B of a lattice Λ, one can obtain another basis $B' = U \times B$ by multiplication with a unimodular matrix U such that $\Lambda(B) = \Lambda(B')$. Indeed, a modular transformation matrix is defined as an integer matrix, whose inverse is also integral. This implies the following properties:

1$-$ U must be integral;

2$-$ U must be square;

3$-$ $|det(U)|$ must be exactly 1.

The following figure is an example of a lattice of dimension 2 and three equivalent basis.

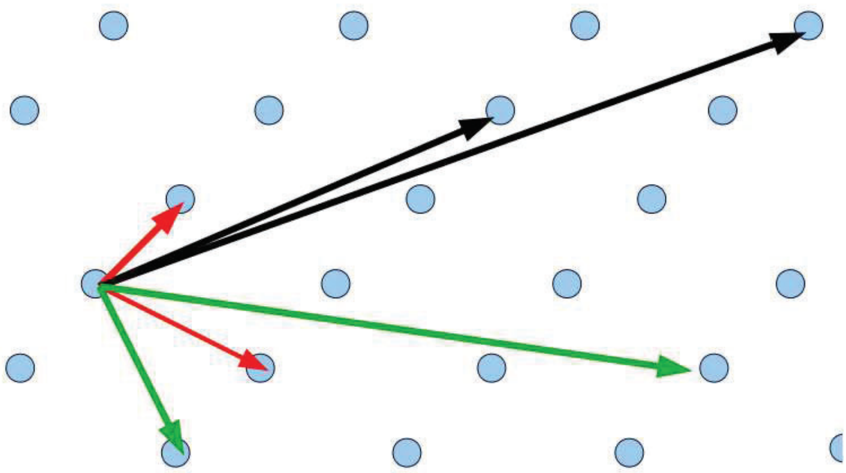


Figure 2.1: A lattice of dimension 2 and three equivalent basis.

The following lemma formalized the notion of equivalent bases.

Lemma 2.1.2. *[24] Two bases B_1 and B_2 in $\mathbb{R}^{n\times d}$ are equivalent if and only if $B_2 = B_1U$ (or $B_1 = B_2U$) for some unimodular matrix $U \in \mathbb{Z}^{d\times d}$.*

Proof. Let $B_1, B_2 \in \mathbb{R}^{n\times d}$ two bases; assume that $B_2 = B_1U$ for some unimodular matrix $U \in \mathbb{Z}^{d\times d}$;
given $y \in \Lambda(B_2)$, we have that $y = B_1Ux$ for some $x \in \mathbb{Z}^d$; let $x' = Ux$, since U is an integer matrix, x' is an integer vector and $y \in \Lambda(B_1)$.Thus $\Lambda(B_2) \subseteq \Lambda(B_1)$; equivalently for $z \in \Lambda(B_1)$, we have $z \in \Lambda(B_2)$.
Therefore, $\Lambda(B_2) = \Lambda(B_1)$.
Now assume that $\Lambda(B_2) = \Lambda(B_1)$. Each column b_i of B_2 lies in $\Lambda(B_2)$ and by assumption also in $\Lambda(B_1)$. Therefore there must exist $x_i \in \mathbb{Z}^d$, such that $b_i = B_1x_i$. Let $U \in \mathbb{Z}^{d\times d}$ be the matrix with $x_1, ..., x_d$ as columns, we see that $B_2 = B_1U$. Similarly there exists $V \in \mathbb{Z}^{d\times d}$ such that $B_1 = B_2V$. Combining the two, we get that $B_2 = B_1U = B_2VU$ and that $B_2^TB_2 = (VU)^TB_2^TB_2VU$. By taking the determinants on both sides, we see that $det(B_2^TB_2) = det(VU)^2det(B_2^TB_2)$ which, unless $det(B_2^TB_2) = 0$, implies that $det(VU) = \pm 1$. Now, since both U and V are integers matrices, it must then be the case that $det(U) = \pm 1$, and we can conclude that U is unimodular. □

Another important notion is that of the dual lattice.

Definition 2.1.3. *Given a lattice $\Lambda \subset \mathbb{R}^n$, the dual lattice $\Lambda^* \subseteq \mathbb{R}^n$ of Λ is defined as*

$$\Lambda^* = \{x \in \mathbb{R}^n \ : \ \forall y \in \Lambda, \ \langle x, y\rangle \in \mathbb{Z}\} \tag{2.3}$$

Notice how the dual lattice is defined in a rather non-constructive way. In a lattice given by Definition 2.1.1, it is clear how one would

find a lattice point: simply choose any vector $x \in \mathbb{Z}^n$ and multiply it by the basis B to obtain a lattice point. Even more simple is to give a basis of the lattice, since this is the starting point of the definition. Nonetheless, the dual lattice is also a lattice in the sense of Definition 2.1.1. This can be seen from the following result which gives us a basis for the dual lattice.

Lemma 2.1.4. *Given a lattice Λ with basis B, define the dual basis $D = B(B^T B)^{-1}$. Then D is a basis for the dual lattice Λ^* for Λ. Thus $\Lambda(B)^* = \Lambda(D)$.*

Proof. Let $y = B(B^T B)^{-1}x$ for some $x \in \mathbb{Z}^n$; let $t = Bx'$ be any lattice point in $\Lambda(B)$ where $x' \in \mathbb{Z}^n$, we have: $\langle y, t\rangle = y^T t = (B(B^T B)^{-1}x)^T Bx' = x^T(B(B^T B)^{-1})^T Bx' = x^T(B^T B)^{-1}(B^T B)x' = x^T x' \in \mathbb{Z}$. Thus $y \in \Lambda(B)^*$.
Now, let $z \in \Lambda(B)^*$. Since $span(B) = span(D)$, we can write $z = Dx$ for $x \in \mathbb{R}^n$. Consider $B^T z$, this is a vector having the inner product of z and all columns of B as entries. But since $B^T z = B^T Dx = B^T B(B^T B)^{-1}x = x$. Therefore we can conclude that $x \in \mathbb{Z}^n$, implying that $z \in \Lambda(D)$. □

We move on and give a few small useful results about a lattice and its dual.

Lemma 2.1.5. *For any lattice Λ it is the case that $(\Lambda^*)^* = \Lambda$.*

Proof. Let B be a basis for Λ. Using Lemma 2.1.4 the basis of $(\Lambda^*)^* = \Lambda$ is

$$(B(B^T B)^{-1})((B(B^T B)^{-1})^T(B(B^T B)^{-1})^{-1} = B$$

□

We will continue with the description of Gram-Schmidt Orthogonalization.

Gram-Schmidt Orthogonalization

The Gram-Schmidt Orthogonalization algorithm is an iterative approach to orthogonalizing vectors of a basis. The first vector b_1 of a given basis B is taken as a reference and the second vector b_2 is projected on to an $(n-1)-$ hyper plane perpendicular to b_1. The third vector b_3 is projected onto a $(n-2)-$ hyper plane perpendicular to the plane described by b_1 and b_2. This process continues in an iterative fashion until all degrees of freedom are exhausted. The new orthogonal vector is denoted by b_i^* and it basis as B^*.

$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^* \quad for \;\; all \;\; 1 \le j < i \le n \tag{2.4}$$

An example of a base of given basis $(b1, b2)$ and its Gram-Schmidt Orthogonalization $(b1*, b2*)$ of a lattice is given by the following figure.

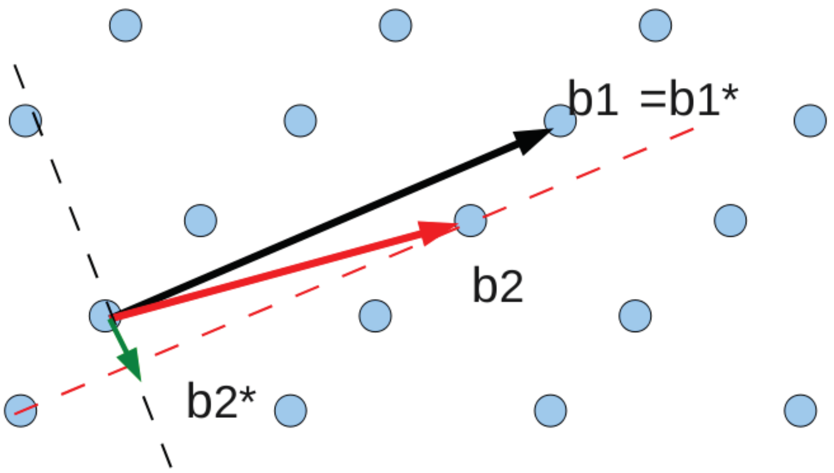


Figure 2.2: Two vectors b_1, b_2 and their Gram-Schmidt Orthogonalization b_1^*, b_2^*.

where $\mu_{ij} = \dfrac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}$.

Remark 2.1.6. *Let Λ be a lattice of dimension n and $B = \{b_1, ..., b_n\}$ a basis thereof. Let B^* be the Gram-Schmidt Orthogonalization of the basis B.*
If $r = \min\limits_{1 \leq i \leq n} \|b_i^*\|$, then any non-zero vector of Λ has a norm greater than r.

2.2 Some invariants of a lattice

Given a lattice Λ, we call a quantity related to Λ an invariant if it does not depend on the choice of the basis of Λ that we could make. Indeed, it is an intrinsic quantity to the lattice, which does not depend on the representation. We have already defined two simple invariants (dimension and rank). We define in this section the fundamental parallelepiped, the minima, the radius, and the volume, also called determinant.
We start with the notion of a fundamental parallelepiped which is tied to a specific lattice basis.

Definition 2.2.1. *[24] For any lattice basis B we define the fundamental parallelepiped of B as*

$$\mathcal{P}(B) = \{Bx | \; x \in \mathbb{R}^n \; , \; \forall i \; : \; 0 \leq x_i < 1\} \, . \tag{2.5}$$

where x_i is the i'th entry in x.

The following figure is an example of a lattice in $\mathbb{R}^n$ with two different bases $B = \{b_1; b_2\}$, $B' = \{b'_1; b'_2\}$, and their corresponding fundamental parallelepipeds $\mathcal{P}(B)$, $\mathcal{P}(B')$.

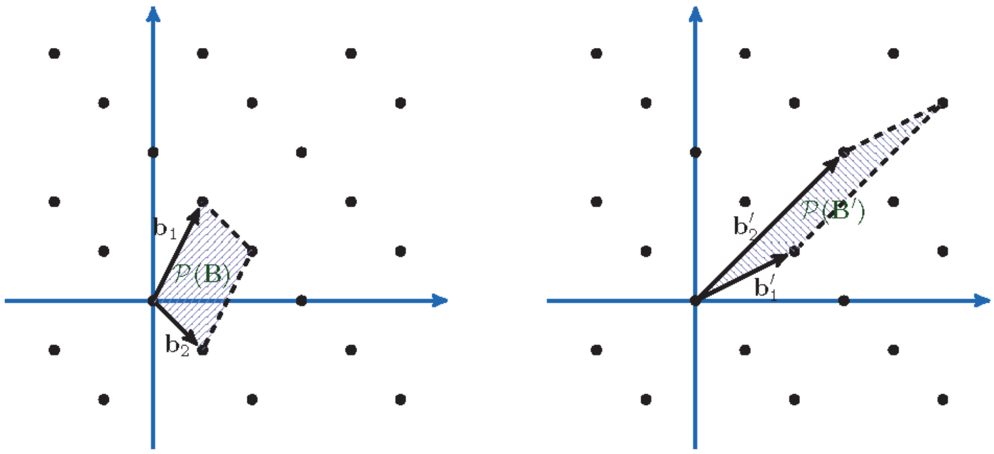


Figure 2.3: A lattice in $\mathbb{R}^n$ shown with two different bases $B = \{b_1; b_2\}$, $B' = \{b'_1; b'_2\}$, and corresponding to fundamental parallelepipeds $\mathcal{P}(B)$, $\mathcal{P}(B')$.

Lemma 2.2.2. *Let Λ be a lattice of rank d, and let $b_1, b_2, ..., b_d \in \Lambda$ be d linearly independent lattices vectors. Then, $b_1, b_2, ..., b_d$ form a basis of Λ if and only if $\mathcal{P}(b_1, b_2, ..., b_d) \cap \Lambda = \{0\}$.*

Proof. Assume first that $b_1, b_2, ..., b_d \in \Lambda$. Then, by Definition 2.1.1, Λ is the set of all their integer combinations. Since $\mathcal{P}(b_1, b_2, ..., b_d)$ is defined as the set of linear combinations of $b_1, b_2, ..., b_d$ with coefficients in $[0; 1[$, the intersection of the two sets is $\{0\}$.
For the second direction, assume that $\mathcal{P}(b_1, b_2, ..., b_d) \cap \Lambda = \{0\}$. Since Λ is a rank d and $b_1, b_2, ..., b_d$ are linearly independent, we can write any lattice vector $x \in \Lambda$ as $\sum y_i b_i$ for some $y_i \in \mathbb{R}$. Since by definition a lattice is closed under addition, the vector $x' = \sum(y_i - \lfloor y_i \rfloor) b_i$ is also in Λ. By our assumption, $x' = 0$. This implies that all y_i are integers and hence x is an integer combination of $b_1, b_2, ..., b_d$. $\square$

In the next definition we will give about basic lattices relating to the fundamental parallelepipeds of different bases for the same lattice.

Definition 2.2.3. *Let $\Lambda(B)$ be a lattice of rank d and dimension n, where $B \in \mathbb{R}^{n \times d}$ is any basis. We define the determinant of a lattice, denoted by $det(\Lambda)$, as the n-dimensional volume of the fundamental parallelepiped $\mathcal{P}(B)$, as below:*

$$det(\Lambda) = \sqrt{det(B^T B)}. \tag{2.6}$$

In the above definition the choice of bases does not matter and so the determinant is well-defined. This is because the n volumes of any two fundamental parallepipeds of a given lattice are equal. This can be seen easily using Lemma 2.1.2. Given two bases B_1 and B_2 of Λ, we know from Lemma 2.1.2 that $B_2 = B_1U$ for some unimodular matrix $U \in \mathbb{Z}^{n\times n}$. This gives us:

$$\sqrt{det(B_2^T B_2)} = \sqrt{det(U^T B_1^T B_1 U)} = \sqrt{det(B_1^T B_1)}. \tag{2.7}$$

If the lattice Λ is of full rank, then B is a square matrix and consequently, we have:

$$det(\Lambda) =\mid det(B) \mid . \tag{2.8}$$

Proposition 2.2.4. *The determinant of a lattice is independent of the choice of the basis B.*

Proof. Let B_1, B_2 be equivalent bases. Then by Lemma 2.1.2, there is a unimodular matrix U such that $B_2 = B_1U$. Thus, $det(\Lambda(B_2))=\sqrt{det(B_2^T B_2)}=$ $\sqrt{det(U^T B_1^T B_1 U)} = \sqrt{det(U)^2.det(B_1^T B_1)} = \sqrt{det(B_1^T B_1)} = det(\Lambda(B_1))$. □

Lemma 2.2.5. *For any lattice $\Lambda = \Lambda(B)$ it is the case that $det(\Lambda^*) = \dfrac{1}{det(\Lambda)}$.*

Proof. We have $det(\Lambda^*) = \sqrt{det((B(B^TB)^{-1})^T(B(B^TB)^{-1})} = \sqrt{det(B^TB)^{-1}} =$ $\dfrac{1}{\sqrt{det(B^TB)}} = \dfrac{1}{det(\Lambda)}$. □

The determinant is a very useful quantity when describing a lattice. One important feature is that the density of the lattice points is inverse proportional to the determinant of the lattice. Finally, we define the minimum distance in a lattice and more generally the $i'th$ successive minimum as follows.

Definition 2.2.6. *Let $\Lambda(B)$ be a lattice of dimension n. Let $i \leq n$, the $i'th$ minimum of the lattice, denoted $\lambda_i(\Lambda)$, is defined by:*

$$\lambda_i(\Lambda) = \min \{r \ , \ dim((\Lambda \cap \mathcal{B}(r))) = i\} . \tag{2.9}$$

The successive minima of a given lattice are all reached. There exist vectors of the lattice of norms equal to the successive minima, and this can be so in particular for linearly independent vectors.

Definition 2.2.7. *For any lattice Λ with a basis B, the minimum distance of Λ is the smallest distance between any two lattices points given as below:*

$$\lambda(\Lambda) = \inf \{\|x - y\| \quad : \quad x, y \in \Lambda, \quad x \neq y\} \tag{2.10}$$

Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice. We say that Λ' is a sublattice of Λ if $\Lambda' \subseteq \Lambda$ is a lattice as well. If Λ' is a sublattice of Λ, then $\lambda_i(\Lambda) \leq \lambda_i(\Lambda')$ for $i \leq \mathtt{dim}(\Lambda')$.

The following figure is an example of a lattice of dimension 2 and a geometrical interpretation of the determinant.

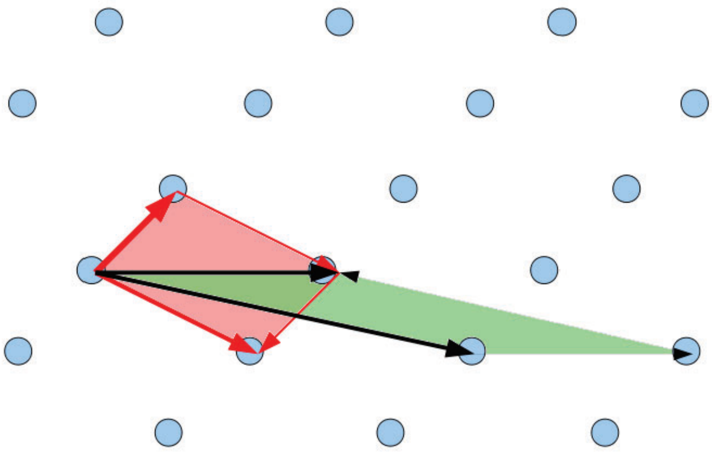


Figure 2.4: A lattice of dimension 2 and geometrical interpretation of the determinant.

We observe that the minimum distance can be equivalently defined as the length of the shortest nonzero lattice vector as bellow:

$$\lambda(\Lambda) = \inf \{\|v\| \quad : \quad v \in \Lambda \setminus \{0\}\}$$

In the above definition the distance between two lattice points is the Euclidean distance. One could have generalized the definition to any norm, but for simplicity we will not.

Remark 2.2.8. *Let Λ be a lattice of dimension n and $B = \{b_1, ..., b_n\}$ any basis. Let B^* be the Gram-Schmidt Orthogonalization of the basis B.*
1- The lattice Λ always admits a vector v of minimal norm ($\|v\| = \lambda_1(\Lambda)$).
2- Given a basis $B = (b_1, ..., b_n)$ of a lattice $\Lambda \subseteq \mathbb{R}^n$, and the associated Gram-Schmidt orthogonalization $B^* = (b_1^*, ..., b_n^*)$, we have $\mathtt{det}(\Lambda) = \prod_{i=1}^{n} \|b_i^*\|$ and $vol(\mathcal{P}(b_1, ..., b_n)) = \prod_{i=1}^{n} \|b_i^\star\|$.

2.3 Minkowski's Theorem and Lattice Reductions

For lattice reduction problems and finding the shortest vectors, a bound is used to check if a given basis can be improved or if it is already very small. The two Minkowski's theorems presented in this section make it possible to simply bound the successive minima of a lattice.

Theorems of Minkowski

Theorem 2.3.1. *(First Theorem of Minkowski)*
For any full-rank lattice $\Lambda \subseteq \mathbb{R}^n$, *we have:*

$$\lambda_1(\Lambda) \leq \sqrt{n}(det(\Lambda))^{1/n} \tag{2.11}$$

where $\lambda_1(\Lambda)$ *denote the minimum Euclidean norm of vectors in* $\Lambda \setminus \{0\}$*.*
$\sqrt{n}(det(\Lambda))^{1/n}$ *is called the Minkowski bound.*

For the proof of this theorem, we will need the following proposition and theorem.

Theorem 2.3.2. *(Minkowski-convex body)*
Let $\Lambda \subseteq \mathbb{R}^n$ *be a* full-rank *lattice. Then for any symmetric central set* S*, if* $vol(S) > 2^n det(\Lambda)$*, then* S *contains a non-zero point of the lattice.*

Proposition 2.3.3. *The volume of a ball of dimension* n *and radius* r *is* $vol(B(O,r)) \geq \left(\frac{2r}{\sqrt{n}}\right)^n$.

Proof. (First Theorem of Minkowski)
Since $\lambda_1(\Lambda)$ is the shortest non-zero vector of the lattice Λ, then $B(O, \lambda_1(\Lambda))$ does not contain any non-zero vector of the lattice. Thus by Theorem 2.3.2, $vol(B(O, \lambda_1(\Lambda))) \leq 2^n.det(\Lambda)$.
Subsequently, from Proposition 2.3.3 we have $vol(B(O, \lambda_1(\Lambda))) \geq \left(\frac{2\lambda_1(\Lambda)}{\sqrt{n}}\right)^n$;
we get then $\left(\frac{2\lambda_1(\Lambda)}{\sqrt{n}}\right)^n \leq vol(B(O, \lambda_1(\Lambda))) \leq 2^n.det(\Lambda)$;
so $\left(\frac{2\lambda_1(\Lambda)}{\sqrt{n}}\right)^n \leq 2^n.det(\Lambda)$;
thus $\frac{2\lambda_1(\Lambda)}{\sqrt{n}} \leq 2\,(det(\Lambda))^{1/n}$;
Therefore, $\lambda_1(\Lambda) \leq \sqrt{n}(det(\Lambda))^{1/n}$. □

Definition 2.3.4. *(Hermite's invariant)*
Hermite's invariant of a given lattice of dimension n *is defined as below:*

$$\gamma(\Lambda) = \left(\frac{\lambda_1(\Lambda)}{det(\Lambda)^{1/d}}\right)^2 \tag{2.12}$$

Theorem 2.3.5. *(Second theorem of Minkowski)*
For all lattice Λ *of dimension* n*, we have:*

$$\left(\prod_{i=1}^{n} \lambda_i(\Lambda)\right)^{1/n} \leq \sqrt{\gamma_n}.det(\Lambda)^{1/n} \tag{2.13}$$

Proof. To do this, it is necessary to use instead of the Euclidean ball of diameter λ_1, disjoint ellipsoids of diameter $\lambda_1, \lambda_2, ..., \lambda_n$ centered on the points of the lattices. Indeed, let $x_1, ..., x_n \in \Lambda$ be linearly vectors achieving the successive minima (i.e $\|x_i\| = \lambda_i(\Lambda)$);
let $x_1^\star, ..., x_n^\star$ be their Gram Schmidt orthogonalization; consider the open ellipsoid T with axes $x_1^\star, ..., x_n^\star$ and lengths $\lambda_1(\Lambda), ..., \lambda_n(\Lambda)$

$$T = \left\{ y \in \mathbb{R}^n \ : \ \sum_{i=1}^{n} \left(\frac{\langle y, x_i^\star \rangle}{\|x_i^\star\| \lambda_i(\Lambda)} \right)^2 < 1 \right\}$$

let $y \in \Lambda$ and let $k = \max\{k \in 1, ..., n \ : \ \|y\| \geq \lambda_k(\Lambda)\}$;
then $y \in span(x_1^\star, ..., x_k^\star) = span(x_1, ..., x_k)$, else $x_1, ..., x_k, y$ would be $k+1$ linearly independent vectors of length less than $\lambda_{k+1}(\Lambda)$;
thus $\sum\limits_{i=1}^{n} \left(\frac{\langle y, x_i^\star \rangle}{\|x_i^\star\| \lambda_i(\Lambda)} \right)^2 = \sum\limits_{i=1}^{k} \left(\frac{\langle y, x_i^\star \rangle}{\|x_i^\star\| \lambda_i(\Lambda)} \right)^2$;
since $\sum\limits_{i=1}^{k} \left(\frac{\langle y, x_i^\star \rangle}{\|x_i^\star\| \lambda_i(\Lambda)} \right)^2 \geq \dfrac{1}{(\lambda_k(\Lambda))^2} \sum\limits_{i=1}^{k} \left(\frac{\langle y, x_i^\star \rangle}{\|x_i^\star\|} \right)^2 = \dfrac{\|y\|^2}{(\lambda_k(\Lambda))^2} \geq 1$, $y \notin T$;
by theorem 2.3.2, $vol(T) \geq 2^n det(\Lambda)$;
on the other hand, by the volume formula for ellipsoids
$vol(T) = \left(\prod\limits_{i=1}^{n} \lambda_i(\Lambda) \right) vol(\mathcal{B}(1)) \geq \left(\prod\limits_{i=1}^{n} \lambda_i(\Lambda) \right) \left(\dfrac{2}{\sqrt{n}} \right)^n$;
combining both bounds yields, $\left(\prod\limits_{i=1}^{n} \lambda_i(\Lambda) \right)^{1/n} \leq \sqrt{n}\, (det(\Lambda))^{1/n}$. □

Minkowski's second theorem generalizes the first and shows that the geometric mean of all the minima of a lattice of dimension n is bounded by a function γ_n and the determinant of the lattice. Indeed, Minkowski's second theorem shows that a basis whose product of vector's norms is of the order of the lattice's volume is a "good basis".
Now, we will recall some lattice reductions allowing either to determine a short vector, or a list of short vectors. In practice, the algorithms often look for a basis whose inner product of the vectors is within a multiplicative constant of the volume of the lattice. For example, the LLL algorithm calculates a "good basis" with an exponential factor in n.
We recall the two fundamental lattice problem below.

2.4 Lattice Problems

In this section, we present some standard lattice problems as well as shortest vector problem (SVP), shortest independent vector problem (SIVP) and closest vector problem (CVP).

2.4.1 Closest vector problem (CVP)

A central problem in the theory of lattice is the closest vector problem (CVP). One need to give a lattice and a target point in the $\mathbb{R}$-linear *span* of that lattice, and then find a closest lattice point to the target. It is often seen as one of the hardest computational lattice problems as many lattice problems polynomially reduce to it. Let $\Lambda \subset \mathbb{R}^d$ be a lattice. Given an arbitrary point $t \in span(\Lambda)$. The vector x in Λ that minimizes the distance $||t - x||$ is called a closest vector to t. Although the Closest Vector Problem is classified as NP-hard [24], there are some lattices where this problem can be solved efficiently. It is the case of integer lattice $\mathbb{Z}^n$, the root lattices A_n $(n \geq 1)$, D_n $(n \geq 2)$, E_n $(n = 6, 7, 8)$, the Leech lattice, and some cases of cyclotomic integer lattices $\mathbb{Z}[\zeta_\alpha]$ (with $\alpha = p.q$, where p and q are prime). We propose a polynomial algorithm to solve the closest vector problem in the tensor product of some root lattices in Chapter 3.

2.4.2 Shortest vector problem (SVP)

The most important computational problem in lattices is the shortest vector problem. The shortest vector problem asks to find a non zero lattice vector of small norm for a given lattice basis as input. This norm is called the first minimum $\lambda_1(\Lambda)$ or the minimum distance and is in general unique up to the sign. This means that: given a basis of a lattice Λ, find a lattice vector whose norm is exactly $\lambda_1(\Lambda)$. This Problem is classified as NP-hard [24]. Minkowski's theorem gives a simple way to bound the length of the shortest lattice vector. Another variant of this problem is *shortest independent vector problem* (SIVP). The shortest independent vector problem asks to find a linearly independent set $\{v_1, ..., v_n\}$ such that all vectors have length at most $\gamma.\lambda_1(\Lambda(B))$ for a given lattice basis B as input (where $\gamma \geq 1$). We construct an enumeration algorithm for integer lattice $\mathbb{Z}^n$ to provide a full list of its shortest vectors. We also construct an algorithm which gives at least n and at most 2^n short vectors of a general orthogonal lattice $\Lambda \subseteq \mathbb{Z}^n$ in Chapter 4.
The main method for tackling these problems is lattice reduction.

2.5 Some Lattice Reductions

A lattice has an infinity of bases, which are all equivalent from an algebraic point of view, this is not true techniqually , and some of these bases have interesting Euclidean properties. The objective of the reduction is to find in a reasonable time a basis of fairly good Euclidean properties, made up of fairly

orthogonal vectors, and short enough to give approximations for successive minima. But in dimension 5, the successive minima do not necessarily form a basis of the lattice. It is therefore difficult to find an absolute criterion which defines what is a good basis. Several notions of reductions exist and each corresponds to a notion of quality of the reduced base. The main reductions are: reduction in the sense of Korkine and Zolotarev, reduction in the sense of Lenstra, Lenstra and Lovàsz, Minkowski's reduction and Schnorr block reduction. It should be noted that the notion of reduction operates a compromise between the quality of the reduction and the complexity to obtain it. For example, the reduction in the sense of Korkine and Zolotarev produces a base whose quality is much higher than that which is produced by the reduction in the sense of Lentra, Lenstra ans Lovàsz, but the computation time to obtain it is greater. In the following, we are only going to be interested in reduction, in the sense of Lenstra, Lentra and Lovàsz and Gauss reduction.
We recall that, the goal of lattice basis reduction is to find a basis with short vectors and orthogonal to each other. We also know that Gram-Schmidt process does not preserve the structure of integer lattice. It would be interesting to focus on the LLL-reduction which uses Gram-Schmidt process and returns integer vectors. The most usual notions of reduction is probably LLL-reduction.

2.5.1 LLL and Gauss Reductions

The LLL- reduction is one of the most commonly used. Let $\frac{1}{4} < \delta < 1$, let $B = (b_1, ..., b_n) \in \mathbb{Z}^{n\times n}$ be a basis of a lattice. We say that B is size-reduced if all Gram-Schmidt coefficients satisfy $\mid \mu_{ij} \mid \leq \frac{1}{2}$.
We say that B satisfies the Lovàsz conditions if for all $i \in \{1, ..., n\}$ we have $\delta \|b_i^*\|^2 \leq \|b_{i+1}^*\|^2 + \mu_{i+1,i} \|b_i^*\|^2$.
Therefore, if a basis B is size-reduced and satisfies the Lovàsz conditions, then we say that B is LLL- reduced. The LLL algorithm is given in [49] and it is showed that the number of LLL swaps is $\mathcal{O}(n^2 \lg\|B\|)$.
The LLL-reduction implies that the norms of Gram-Schmidt-Orthogonalization vector never drop too fast. Indeed the vectors are not far from being orthogonal.
LLL-reduction does not solve the problem for all lattices. Indeed, for random lattice, we use the Gaussian heuristic and Gauss reduction to obtain the list of short vectors of the lattice. This method is called Sieve. We will define Gauss reduced as below.

Definition 2.5.1. *(Gauss reduction)*

For vectors $u, v \in \Lambda$*, if* $\max(\|u\|, \|v\|) \leq \min(\|u - v\|, \|u + v\|)$*, then* u*,* v *are called Gauss-reduced.*

2.5.2 Hermite-Korkine-Zoltarev (HKZ)-reduction

A basis $B = (b_1, ..., b_n)$ is said to be HKZ(Hermite-Korkine-Zolotarev)-reduced if its first vector reaches the *minimum* of Λ and if *orthogonally* to b_1 the other b_i's are themselves $HKZ-$reduced. This implies that for any i, we have: $\|b_i^*\| \leq \sqrt{n-i+1}\left(\|b_j^*\|\right)^{\frac{1}{n-i+1}}$.

Remark 2.5.2. *Each of the two reductions has its own particularity. Indeed,* $HKZ-$*reduction is very* strong*, but* expensive *to compute. On the other hand,* $LLL-$*reduction is* fairly cheap*, but an* $LLL-$*reduced basis is of much lower quality.*

2.5.3 Minkowski's reduction

A basis $B = (b_1, ..., b_n)$ of a lattice Λ is reduced in the sense of Minkowski if the following conditions hold:
- The vector b_1 is the short vector in lattice Λ;
- The vector b_{i+1} is the shortest among all independent vectors of vectors $(b_1, ..., b_i)$, so that $(b_1, ..., b_{i+1})$ can be extended to a basis of Λ.

In an equivalent way, a basis is reduced in the sense of Minkowski if the following inequalities are satisfied:

$$\forall\ i \leq n,\ \|x_1 b_1 + ... + x_n b_n\| \geq \|b_i\| \tag{2.14}$$

for all n-tuples of integers $(x_1, ..., x_n)$ formed by the integers $x_i, ..., x_n$ relatively prime.

2.6 Some root lattices

Root lattices emerge from so called root systems of vectors. There are three families of root lattices (A, D and E), and they have been the object of very detailed studies [12, 13, 14, 40]. In the following, we recall the definitions of the root lattices of type A_n $(n \geq 1)$, D_n $(n \geq 2)$, and give their generator matrix.

2.6.1 Definition and Basis of A_n $(n \geq 1)$

Definition 2.6.1. *Let n be a positive integer. The subset A_n $(n \geq 1)$ of $\mathbb{R}^{n+1}$ defined by:*

$$A_n := \{\boldsymbol{x} \in \mathbb{Z}^{n+1} \ : \ \langle \boldsymbol{x}, \overline{1} \rangle = 0\}, \tag{2.15}$$

where $\overline{1} := (1, 1, \cdots, 1)$, is a lattice of rank n in $\mathbb{R}^n$.

The shortest vectors in the lattice A_n $(n \geq 1)$ are all the permutations of $(1, -1, 0, 0, ..., 0)$. The basis of the root lattice A_n is given in the following Lemma 2.6.2.

Lemma 2.6.2. *(Basis of A_n $(n \geq 1)$) A generator matrix of the lattice A_n is the $n \times (n+1)$-matrix B given by:*

$$\mathrm{B} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & & & \cdots & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{pmatrix}, \tag{2.16}$$

A generator matrix of its dual A_n^* is the $n \times (n+1)$-matrix B^* given by:

$$\mathrm{B}^* = \frac{1}{n+1} \begin{pmatrix} n & -1 & -1 & \cdots & -1 \\ -1 & n & -1 & \cdots & -1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -1 & \cdots & n & -1 & -1 \\ -1 & \cdots & -1 & n & -1 \end{pmatrix}, \tag{2.17}$$

with $\frac{n}{n+1}$ on the main diagonal and $\frac{-1}{n+1}$ everywhere else.

2.6.2 Definition and Basis of D_n $(n \geq 2)$

In the following, we recall the definition of the root lattice of type D_n $(n \geq 2)$, and give its generator matrix.

Definition 2.6.3. *Let n be a positive integer. The subset D_n $(n \geq 2)$ of $\mathbb{R}^n$ defined by:*

$$D_n := \{\boldsymbol{x} \in \mathbb{Z}^n \ : \ \langle \boldsymbol{x}, \overline{1} \rangle \text{ is even}\}, \tag{2.18}$$

where $\overline{1} := (1, 1, \cdots, 1)$, is a lattice of rank n in $\mathbb{R}^n$.

The shortest vectors in the lattice D_n are all the permutations of $(\mp 1, \mp 1, 0, 0, ..., 0)$.

Lemma 2.6.4. *(Root lattice D_n^*) Let $n \geq 3$, the lattice D_n^* dual to D_n is*

$$D_n^* = \bigcup_{i=0}^{3} ([i] + D_n) \tag{2.19}$$

where, $[0] = (0^n)$, $[1] = (\frac{1}{2})^n$, $[2] = (0^{n-1}, 1)$ *and* $[3] = (\frac{1}{2}^{n-1}, -\frac{1}{2})$.

In the following sections it will be useful to know a basis for D_n and D_n^*.

Lemma 2.6.5. *(Basis of D_n and D_n^*)[13] A generator matrix of the lattice D_n is the $n \times n$-matrix* B *given by:*

$$\mathrm{B}^* = \begin{pmatrix} -1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & & & \ldots & & & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{pmatrix} \tag{2.20}$$

A generator matrix of the lattice D_n^* dual to D_n is the $n \times n$-matrix B^* *given by:*

$$\mathrm{B}^* = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \ldots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \tag{2.21}$$

2.7 Concluding remarks

Some important definitions and properties of lattices have been given. The invariant of lattices, some reductions of lattices and the two principal problems of lattice have been respectively presented. Some important definitions and properties that will be useful to bring out the results in the next chapters were also discussed.

Chapter Three

Closest Vector Problem in tensored root lattices of some lattices of type A and type D

In this chapter, we propose a polynomial algorithm to solve the Closest Vector Problem (CVP) in the tensor product of three root lattices of type A_n ($n \geq 1$), and two root lattices of type D_n ($n \geq 2$). In 2018, Léo Ducas and Wessel van Woerden proposed a polynomial algorithm allowing to solve this problem in the tensor product of two root lattices of type A_n ($n \geq 1$) [22]. In our present case, we use the associativity of the lattice of type A and the same techniques to solve this problem in the tensor product of three lattices of type A. And we show that the root lattice D_{nm} is a full rank sub-lattice of the tensor product $D_n \otimes D_m$ ($n, m \geq 2$) of the root lattices D_n and D_m, enabling us to derive a polynomial algorithm for solving the Closest Vector Problem in D_n ($n \geq 2$). The proposed algorithm performs at most $O(n+m)$ arithmetic operations. A motivation could be to use the full characterization of the Voronoi relevant vector in this case in terms of simple cycle in the complete directed tripartite graph $K_{n+1,m+1,p+1}$. So we need to establish the relationship between the Voronoi relevant vectors in the tensor product $A_n \otimes A_m \otimes A_p$ and the complete directed tripartite graphs $K_{n+1,m+1,p+1}$. Subsequently, we will modify some parameters of the polynomial algorithm in [13] to solve this problem in $A_n \otimes A_m \otimes A_p$, and even in the tensor product of a finite number of lattices $A_{n_1} \otimes \dots \otimes A_{n_k}$ ($n_1, \dots, n_k \geq 1$) of type A. So we determined a polynomial algorithm to solve CVP in $A_n \otimes A_m \otimes A_p$ in $O\left(d.(((n+1)(m+1)-1)p)^2 \min\left\{(n+1)(m+1)-1, p\right\}\right)$ ($d \geq 1$) arithmetic operations, and an algorithm to solve this problem in $k \geq 4$ root lattices $A_{n_1} \otimes \dots \otimes A_{n_k}$.

This chapter is organized as follows: In Section 3.1, we review the definitions of graphs, tensor product and basic properties of the root lattices of type

A, D and simple graph to understand the results of further sections. In section 3.4.2 and Section 3.2, we present the characterization of the voronoi relevant vector in the tensor product of three root lattices of type A, give a polynomial algorithm to solve the problem of the nearest vector in $A_n \otimes A_m \otimes A_p$, and We will also determine a polynomial algorithm to solve the closest vector problem in the general case of the tensor product of k ($k \geq 4$) root lattices of type A ($A_{n_1} \otimes ... \otimes A_{n_k}$, $n_1, ..., n_k \geq 1$). This algorithm runs in $O\left(d.(((n_1+1)...(n_{k-1}+1)-1)n_k)^2 \min\left\{(n_1+1)...(n_{k-1}+1)-1, n_k\right\}\right)$ (where $d \geq 1$) arithmetic operations. In Section 3.4.3, we propose a polynomial algorithm to solve CVP in the tensor product $D_n \otimes D_m$ ($n, m \geq 2$), where D_n and D_m are two root lattices of type D.

3.1 Preliminaries

Although the closest vector problem is classified as NP-hard [24], there are some lattices where this problem can be solved efficiently. It is the case of integer lattice $\mathbb{Z}^n$, the root lattices A_n ($n \geq 1$), D_n ($n \geq 2$), E_n ($n = 6, 7, 8$), the Leech lattice, and some cases of cyclotomic integer lattices $\mathbb{Z}[\zeta_\alpha]$ (with $\alpha = p.q$, where p and q are prime).
We recall here the definitions and properties that will be used throughout this chapter.
All definitions in this section are taken from [22, 49]. We start with the definitions of tensor product of two and three lattices.

Definition 3.1.1. *Let $\Lambda_1 \subseteq \mathbb{R}^{n_1}$ and $\Lambda_2 \subseteq \mathbb{R}^{n_2}$ be lattices of respectively ranks n_1 and n_2. Let $a_1, ..., a_{n_1} \in \mathbb{R}^{n_1}$ and $b_1, ..., b_{n_2} \in \mathbb{R}^{n_2}$ be respective bases. The tensor product $\Lambda_1 \otimes \Lambda_2 \subset \mathbb{R}^{n_1 n_2}$ is defined as the lattice with basis $\left\{a_i \otimes b_j : i \in \{1, ..., n_1\}, j \in \{1, ..., n_2\}\right\}$.*
Here $x \otimes y = (x_1, ..., x_{n_1}) \otimes (y_1, ..., y_{n_2})$ with $x \in \mathbb{R}^{n_1}$ and $y \in \mathbb{R}^{n_2}$ is defined as the natural embedding in $\mathbb{R}^{n_1 n_2}$ as follows :
$(x_1y_1, x_1y_2, ..., x_1y_{n_1}, x_2y_1, ..., x_{n_1}y_{n_2}) \in \mathbb{R}^{n_1 n_2}$.
For three lattices, the tensor product $\Lambda_1 \otimes \Lambda_2 \otimes \Lambda_3 \subset \mathbb{R}^{n_1 n_2 n_3}$ (with $\Lambda_3 \subset \mathbb{R}^{n_3}$ and its basis $c_1, ..., c_{n_3} \in \mathbb{R}^{n_3}$) is defined as the lattice with basis:
$\left\{a_i \otimes b_j \otimes c_k : i \in \{1, ..., n_1\}, j \in \{1, ..., n_2\}, k \in \{1, ..., n_3\}\right\}$.
Here $x \otimes y \otimes z = (x \otimes y) \otimes z = ((x_1y_1, x_1y_2, ..., x_1y_{n_1}, x_2y_1, ..., x_{n_1}y_{n_2}) \otimes z)$ thus, $x \otimes y \otimes z = (x_1y_1z_1, x_1y_1z_2, ..., x_1y_1z_{n_3}, x_1y_2z_1, ..., x_{n_1}y_{n_2}z_{n_3}) \in \mathbb{R}^{n_1 n_2 n_3}$.

Definition 3.1.2. *Let $\Lambda_1 \subseteq \mathbb{R}^{n_1}$, $\Lambda_2 \subseteq \mathbb{R}^{n_2}$, ..., $\Lambda_k \subseteq \mathbb{R}^{n_k}$ be lattices of respectively ranks $n_1, ..., n_k$; let $a_1^{(1)}, ..., a_{n_1}^{(1)} \in \mathbb{R}^{n_1}$; $a_1^{(2)}, ..., a_{n_2}^{(2)} \in \mathbb{R}^{n_2}$;..., $a_1^{(k)}, ..., a_{n_k}^{(k)} \in$*

$\mathbb{R}^{n_k}$ *be respective bases. The tensor -product* $\Lambda_1 \otimes \Lambda_2 \otimes ... \otimes \Lambda_k \subset \mathbb{R}^{n_1 n_2 ... n_k}$ *is defined as a lattice with basis:*
$\left\{a^{(1)}_{i^{(1)}} \otimes a^{(2)}_{i^{(2)}} \otimes ... \otimes a^{(k)}_{i^{(k)}} : i^{(1)} \in \{1, ..., n_1\}, i^{(2)} \in \{1, ..., n_2\}, ... i^{(k)} \in \{1, ..., n_k\}\right\}$.
Here, we use the associativity to compute:
$x^{(1)} \otimes x^{(2)} \otimes ... \otimes x^{(k)} = \left(x^{(1)}_1 x^{(2)}_1 ... x^{(k)}_1, x^{(1)}_1 x^{(2)}_1 ... x^{(k)}_2, ..., x^{(1)}_{n_1} x^{(2)}_{n_2} ... x^{(k)}_{n_k}\right) \in \mathbb{R}^{n_1 n_2 ... n_k}$.

We will continue with the notion of Voronoi region. In the following, we give its definition and some properties.

Definition 3.1.3. *Let* Λ *be a lattice of dimension* n*. The Voronoi region of* Λ *is defined as below:*

$$V(\Lambda) = \{x \in span(\Lambda) \ : \|x\| \leq \|x - v\| \ for \ all \ v \in \Lambda\} \tag{3.1}$$

So the Voronoi region consists of all points of $span(\Lambda)$ *that are at least as close to* $0 \in \Lambda$ *as to any other point of* Λ*.*

The Voronoi region is the intersection of half spaces $H_v := \{x \in span(\Lambda) : 2\langle x, v\rangle \leq \langle v, v\rangle\}$ for all $v \in \Lambda \setminus \{0\}$. Note that the only half spaces H_v in this intersection that matter are those corresponding to a facet ($rank(\Lambda) - 1$ dimensional face of $V(\Lambda)$) $\{x \in span(\Lambda) : \ \|x\| = \|x - v\|\} \cap V(\Lambda)$ of the Voronoi region. Such $v \in \Lambda$ are called Voronoi relevant vector.

Definition 3.1.4. *Let* Λ *be a lattice of dimension* n*. The Voronoi relevant vectors are the minimal set* $RV(\Lambda) \subset \Lambda$ *of vectors such that:*

$$V(\Lambda) = \bigcap_{v \in RV(\Lambda)} H_v \tag{3.2}$$

Voronoi showed that for $v \in \Lambda \setminus \{0\}$ we have that v is a Voronoi relevant vector if and only if 0 and v are the only closest vectors to $\frac{1}{2}v$ in Λ. It was proved by Minkowski in 1897 that a lattice of rank n can only have at most $2(2^n - 1)$ Voronoi relevant vectors [22].

Lemma 3.1.5. *Let* Λ *be a lattice.* $v \in \Lambda \setminus \{0\}$ *is a Voronoi relevant vector if and only if :*

$$\langle v, x\rangle < \langle x, x\rangle \ for \ all \ x \in \Lambda \setminus \{0, v\} \tag{3.3}$$

Proof. Let Λ be a lattice and let $v \in \Lambda \setminus \{0\}$ a Voronoi relevant vector of Λ; we have $\|\frac{1}{2}v - x\|^2 - \|\frac{1}{2}v\|^2 = \langle x, x\rangle - \langle v, x\rangle$ and thus for a $v \in \Lambda \setminus \{0\}$ and all $x \in \Lambda \setminus \{0, v\}$; note that both 0 and v have exactly distance $\|\frac{1}{2}v\|$ to $\frac{1}{2}v$ and therefore the first statement is that of the definition, while the later statement is that of the lemma. □

Lemma 3.1.6. *Let $t \in span(\Lambda)$ and $x \in \Lambda$. There exists a vector $y \in \Lambda$ such that $\|(x+y)-t\| < \|x-t\|$ if and only if there exists a Voronoi relevant vector $v \in RV(\Lambda)$ such that $\|(x+v)-t\| < \|x-t\|$.*

Proof. Let $t \in span(\Lambda)$ and $x \in \Lambda$. Assume that there exists a vector $v \in RV(\Lambda)$ such that $\|(x+v)-t\| < \|x-t\|$. Since $RV(\Lambda) \subset \Lambda$, then for $y = v$, we have $\|(x+y)-t\| < \|x-t\|$.
Now suppose there exists a vector $v \in \Lambda$ such that $\|(x+y)-t\| < \|x-t\|$; then $\|y-(t-x)\| < \|t-x\|$; thus $(t-x) \notin H_v$; therefore $(t-x) \notin V(\Lambda)$. So there exists a vector $v \in RV(\Lambda)$ such that $\|t-x\| > \|(t-x)-v\|$; therefore there exists $v \in RV(\Lambda)$ such that $\|(x+v)-t\| < \|x-t\|$. □

3.2 The closest vector problem in some root lattices of type A_n

We start this with the case of root lattice and type A_n ($n \geq 1$) as below.

3.2.1 The closest vector problem in root lattice A_n

We will start this section by characterizing the vectors of A_n. We recall that the lattice A_n consists of all vectors $x = (x_1, \cdots, x_{n+1}) \in \mathbb{Z}^{n+1}$ such that

$$\sum_{i=1}^{n+1} x_i = 0, \tag{3.4}$$

The algorithm below is a polynomial CVP algorithm for the lattice A_n [22]. This algorithm takes as input a vector $t \in \texttt{span}(A_n)$, calculates the round $x' = \lceil t \rfloor$ of this vector (the vector x' is a close vector to t). We calculate the sum of the components x' and, if this sum is equal to zero, then it is the closest vector to t, else we add or remove successively 1 to some components as shown by the algorithm below.

Algorithm 1 A polynomial CVP algorithm for the lattice A_n

Require: Given a target $\mathbf{t} = \left(t_1, \cdots, t_{(n+1)}\right) \in \mathtt{span}(A_n)$.
Ensure: A closest vector $\mathbf{x}$ to $\mathbf{t}$ in A_n.

1: Let $\mathbf{x}' = \left(\lceil t_1 \rfloor, \cdots, \lceil t_{(n+1)} \rfloor\right) \in \mathtt{span}(A_n)$ is a close vector to $\mathbf{t}$

2: Compute $\delta(t_i) := t_i - \lceil t_i \rfloor$. Let $\Delta := \sum_{i=1}^{(n+1)} x'_i$ the deficit of $\mathbf{x}'$

$\triangleright$: Note that $\mathbf{x}' \in A_n$ if and only if $\Delta = 0$.

3: Put $\delta(t_1), \cdots, \delta(t_{n+1})$ in ascending order as below:

$$-\frac{1}{2} \leq \delta(t_{i_1}) \leq \cdots \leq \delta(t_{i_{n+1}}) \leq \frac{1}{2}, \qquad \text{(we rank in ascending order.)}$$

4: a) if $\Delta = 0$, then $\mathbf{x} = \mathbf{x}'$ is a closest vector to t;

b) if $\Delta > 0$, then a closest vector $\mathbf{x}$ to t is obtained from x' by subtracting 1 from $x'_{i_1}, \cdots, x'_{i_\Delta}$.

c) if $\Delta < 0$, then a closest vector $\mathbf{x}$ to t is obtained from adding 1 to $x'_{i_{(n+1)}}, \cdots, x'_{i_{(n+1)+\Delta+1}}$.

Example 3.2.1. *Finding some closest vector in A_8.*
*Consider the vector $t = (1.3, -0.7, -0.6, 2, -3, 1, 0, 2.7, -2.7) \in$ **span**(A_8); (we have $t \in$ **span**(A_8) because $\sum_{i=1}^{9} t_i = 0$);*
we will determine a nearest vector $x \in A_8$ of t.

(1) we will have: $x' = (1, -1, -1, 2, -3, 1, 0, 3, -3)$;
thus $\Delta = \sum_{i=1}^{9} x'_i = 1 - 1 - 1 + 2 - 3 + 1 + 0 + 3 - 3 = -1$;
so $\Delta = -1$;

(2) we will start by calculating $\delta(t_i)$ for $i = 1, \cdots, 9$ as below:
$\delta(t_1) = 1.3 - 1 = 0.3$, $\delta(t_2) = -0.7 + 1 = 0.3$, $\delta(t_3) = -0.6 + 1 = 0.4$, $\delta(t_4) = 2-2 = 0$, $\delta(t_5) = -3+3 = 0$, $\delta(t_6) = 1-1 = 0$, $\delta(t_7) = 0-0 = 0$, $\delta(t_8) = 2.7 - 3 = -0.3$ *and* $\delta(t_9) = -2.7 + 3 = 0.3$;
in the following, we will arrange these $\delta(t_i)$ in ascending order (and this as in the algorithm of previous section):

- *we have: $\delta(t_{i_1}) = 0.3 \leq \delta(t_{i_2}) = 0.3 \leq \delta(t_{i_3}) = 0.4$;*
- $\delta(t_{i_4}) = \delta(t_{i_5}) = \delta(t_{i_6}) = 0$;
- $\delta(t_{i_7}) = -0.3 \leq \delta(t_{i_8}) = 0 \leq \delta(t_{i_9}) = 0.3$;

(3) we have, $\Delta = -1$,
given that $\Delta = -1 < 0$, *we will only add* 1 *to* x'_{i_9} *(since* $x'_{i_{9-1+1}} = x'_{i_9}$*);*
so: $x = (1, -1, -1, 2, -3, 1, 0, 3, -3 + 1) = (1, -1, -1, 2, -3, 1, 0, 3, -2$
Therefore, the nearest vector of t *in* A_8 *is:*
$x = (1, -1, -1, 2, -3, 1, 0, 3, -2)$

3.2.2 The closest vector problem in root lattice $A_n \otimes A_m$

We start this section by the characterization of the vectors of the root lattice $A_n \otimes A_m$ $(n, m \geq 1)$ as below. We first recall the definition of the tensor product:

Definition 3.2.2. *Let* $\Lambda_1 \subseteq \mathbb{R}^{n_1}$ *and* $\Lambda_2 \subseteq \mathbb{R}^{n_2}$ *be lattices of respectively ranks* n_1 *and* n_2,
let $a_1, ..., a_{n_1} \in \mathbb{R}^{n_1}$ *and* $b_1, ..., b_{n_2} \in \mathbb{R}^{n_2}$ *be their respective bases. The tensor product* $\Lambda_1 \otimes \Lambda_2 \subset \mathbb{R}^{n_1 n_2}$ *is defined as the lattice with basis* $\{a_i \otimes b_j : i \in \{1, ..., n_1\},\ \ j \in \{1, ..., n_2\}\}$.
Here $x \otimes y = (x_1, ..., x_{n_1}) \otimes (y_1, ..., y_{n_2})$ *with* $x \in \mathbb{R}^{n_1}$ *and* $y \in \mathbb{R}^{n_2}$ *can be seen as an element of* $\mathbb{R}^{n_1 n_2}$ *as follows :* $(x_1 y_1, x_1 y_2, ..., x_1 y_{n_1}, x_2 y_1, ..., x_{n_1} y_{n_2}) \in \mathbb{R}^{n_1 n_2}$.

Characterisation of the vectors of the root lattice $A_n \otimes A_m$

The root lattice $A_n \otimes A_m \subseteq \mathbb{Z}^{(n+1)(m+1)}$ $(n, m \geq 1)$ consists of all elements $x = (x_{11}, ..., x_{1(m+1)}, x_{21}, ..., x_{2(m+1)}, ..., x_{(n+1)1}, ..., x_{(n+1)(m+1)}) \in \mathbb{Z}^{(n+1)(m+1)}$ satisfying the following conditions:

(1) $\sum\limits_{i=1}^{n+1} x_{ij} = 0 \ \ for \ \ all \ \ j = 1, ..., m + 1$

(2) $\sum\limits_{j=1}^{m+1} x_{ij} = 0 \ \ for \ \ all \ \ i = 1, ..., n + 1.$

The notation $x = (x_{11}, ..., x_{1(m+1)}, x_{21}, ..., x_{2(m+1)}, ..., x_{(n+1)1}, ..., x_{(n+1)(m+1)})$ above, means that there exist two vectors $u = (u_1, ..., u_{n+1}) \in A_n$ and $v = (v_1, ..., v_{m+1}) \in A_m$ such that: $x_{ij} = u_i v_j$ for $i = 1, ..., n + 1$ and $j = 1, ..., m + 1$.

Basis of root lattice $A_n \otimes A_m$

A basis of the root lattice $A_n \otimes A_m$ has some nice properties. First let $b^{ij} \in A_n \otimes A_m$ be given by:

- $b^{ij}_{i,j} = b^{ij}_{i+1,j+1} = 1$;
- $b^{ij}_{i+1,j} = b^{ij}_{i,j+1} = -1$;

and 0 otherwise for all $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$.
Therefore, we can note $B := \{b^{ij} : \ i \in \{1, ..., n\} \ and \ j \in \{1, ..., m\}\}$ as a basis of $A_n \otimes A_m$. Because the basis B is so sparse, we can efficiently encode and decode elements in this basis.

Example 3.2.3. *A good basis of the root lattice $A_2 \otimes A_3$ is given by:*
$B = \{b^{11}, b^{12}, b^{13}, b^{21}, b^{22}, b^{23}\}$ *where:*

- $b^{11} = (1, -1, 0, 0, -1, 1, 0, 0, 0, 0, 0, 0)$;
- $b^{12} = (0, 1, -1, 0, 0, -1, 1, 0, 0, 0, 0, 0)$;
- $b^{13} = (0, 0, 1, -1, 0, 0, -1, 1, 0, 0, 0, 0)$;
- $b^{21} = (0, 0, 0, 0, 1, -1, 0, 0, -1, 1, 0, 0)$;
- $b^{22} = (0, 0, 0, 0, 0, 1, -1, 0, 0, -1, 1, 0)$;
- $b^{23} = (0, 0, 0, 0, 0, 0, 1, -1, 0, 0, -1, 1)$.

Solving the closest vector problem in root lattice $A_n \otimes A_m$

The results of this section are taken from [22].
We will characterize the Voronoi relevant vector in the root lattice $A_n \otimes A_m$. First, we will limit the search space by the following lemma.

Lemma 3.2.4. *For all Voronoi relevant vectors $v \in A_n \otimes A_m$, we have $|v_{ij}| < 2$ for all $i \in \{1, ..., n+1\}$ and $j \in \{1, ..., m+1\}$.*

Proof. Let $u \in A_n \otimes A_m$ be a Voronoi relevant vector. We suppose that there exist i, j such that $|u_{ij}| \geq 2$; because of symmetry we can assume without loss of generality that $|u_{11}| \geq 2$. And because u is a Voronoi relevant vector if and only if $-u$ is also a Voronoi relevant vector, we can also assume that $u_{ij} \geq 2$.
Let $x^{ij} \in A_n \otimes A_m$ for all $i = 2, ..., n+1$ and $j = 2, ..., m+1$ be given by $x_{11} = x_{ij} = 1$; $x_{i1} = x_{1j} = -1$ and 0 otherwise.
Note that $\langle x^{ij}, x^{ij} \rangle = 4$ for all i, j. Then by definition 3.1.3 we get: $u_{11} + u_1 + u_{ij} - u_{i1} - u_{j1} = \langle u, x^{ij} \rangle < \langle x^{ij}, x^{ij} \rangle = 4$ for all $i = 1, ..., n+1$; and $j = 1, ..., m+1$.
also note that because these are all integers, we even have that:
$u_{11} + u_1 + u_{ij} - u_{i1} - u_{j1} \leq 3$. Summing multiple of these relations for a fixed $j = 2, ..., n+1$ gives:
$mu_{11} - mu_{i1} + \sum\limits_{j=2}^{m+1} (u_{1j} + u_{ij}) \leq 3(m+1-1) = 3m$;
furthermore $-u_{11} = \sum\limits_{j=2}^{m+1} u_{1j}$ and $u_{i1} = \sum\limits_{j=2}^{m+1} u_{ij}$;
so the inequation becomes: $(m+1)u_{11} - (m+1)u_{i1} \leq 3m$;
as a result of $u_{11} \geq 2$, we now get that: $u_{i1} \geq \dfrac{3}{m+1} - 1$;
and thus $u_{i1} > -1$; for all $i = 2, ..., n+1$ and $j = 1, ..., m+1$; then thus $u_{i1} \geq 0$;

Figure 3.1: Example graph G_t corresponding to
$t = (0, 0, 0, 0, 0, 1, 0, -1, 0, 0, -1, 0, 1, -1, 1, 0, 0, 0, 1, -1) \in A_3 \otimes A_4$.

for all $i = 2, ..., n+1$ and $j = 1, ..., m+1$
but in that case: $0 = \sum_{i=1}^{n+1} u_{i1} \geq 2 + 0 + ... + 0 = 2$ which gives a contradiction.
so $|u_{11}| < 2$ □

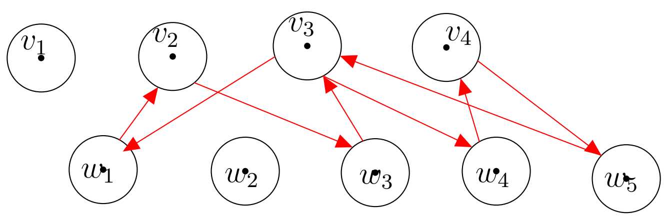
As a result all Voronoi relevant vectors of $A_n \otimes A_m$ must lie in $X := \{-1, 0, 1\}^{(n+1)(m+1)} \bigcap (A_n \otimes A_m)$. In the following, we will give the correspondence between the elements of X and certain subgraphs of the complete directed bipartite labelled graph $K_{n+1,m+1} = (V, E)$. We label the $n+1$ nodes $V_1 := \{v_1, ..., v_{n+1}\}$ and $V_2 := \{w_1, ..., w_{m+1}\}$. Let $V = V_1 \cap V_2$; we let a coefficient $t_{ij} \in X$ corresponds to the pair (v_i, w_j) of nodes of $K_{n+1,m+1}$.

Definition 3.2.5. *Let $t \in \{-1, 0, 1\}^{(n+1)(m+1)}$ be given. We will define the subgraph $G_t = (V_t, E_t) \subset K_{n+1,m+1} = (V, E)$ corresponding to t. Let E_t consist of the following directed edges:*

- *The edge (v_i, w_j) for each t_{ij} that has value -1;*
- *The edge (w_j, v_i) for each t_{ij} that has value 1;*

and let V_t consist of all nodes with non zero in-or outdegree.

Example 3.2.6.



Proposition 3.2.7. *The Voronoi relevant vectors of root lattice $A_n \otimes A_m$ are precisely all $v \in X \setminus \{0\}$ such that G_v is connected and the indegree and outdegree of every node is exactly 1.*

Proof. Let $u \in X \setminus \{0\}$ be given. Note that we already have:
$\langle u, x \rangle = \sum_{i,j} u_{ij} x_{ij} \leq \sum_{i,j} |x_{ij}| \leq \sum_{i,j} |x_{ij}|^2 = \langle x, x \rangle$. for $x \in A_n \otimes A_m$. because $u \in X \setminus \{0\}$. We remark that if $x \in X$, we have $\sum_{i,j} |x_{ij}| = \sum_{i,j} |x_{ij}|^2$ and $\langle x, x \rangle = \sum_{i,j} |x_{ij}|$ if and only if $u_{ij} x_{ij} = |x_{ij}|$ for all $i = 1, ..., n+1$ and

$j = 1, ..., m+1$; so $x_{ij} = 0$ or $x_{ij} = u_{ij} \in X \setminus \{0\}$.
This makes it clear that the only candidates such that $\langle u, x\rangle = \langle x, x\rangle$ are those $x \in X$ such that $G_x \subset G_u$. By Lemma 22 [22], we get that $u \in RV(A_n \otimes A_m)$ if and only if G_o and G_u are the only subgraphs of that form of G_u. In fact note that each G_x with $x \in X$ consists of a union of disconnected Eulerian graphs and thus a union of disconnected cycles. Furthermore note that every cycle in G_x corresponds to a subgraph $H \subset G_x$ for which there exists an $x' \in X$ such that $H = G_{x'}$. But that means that G_u is a Voronoi relevant vector if and only if G_u contains only the trivial cycle G_o and G_u and no other cycles. We will show that this is only the case when G_u is a simple cycle.
Because G_u is a union of disconnected cycles, we must have that G_u is connected as otherwise taking one of those disconnected cycles would give a non trivial subgraph. $G_x \subsetneq G_u$. So G_o must be connected and thus consist of a single cycle. In the case G_u contains a non trivial cycle, the one when starting in w and returning to w for the first time. So G_u must be connected and the indegree and outdegree of every node must be 1. But in that case G_u is a simple cycle and it is clear that G_u only has the trivial cycles corresponding to G_o and G_u. So u is a voronoi relevant vector in that case. □

Lemma 3.2.8. *Let $x \in A_n \otimes A_m$ and let $t \in span(A_n \otimes A_m)$ be our target. If there exists a Voronoi relevant vector $v \in RV(A_n \otimes A_m)$ such that $\|(x+v) - t\| < \|x - t\|$, we can find such a Voronoi relevant vector in $O((n+m)nm)$ arithmetic operations on reals. If it does not exist this will also be detected by the algorithm.*

Proof. Let $u := x - t$ be the difference vector of t and x. We construct weighted directed complete bipartite graph $K_{n+1,m+1}(u)$ with weight function W defined as follows: for $i \in \{1, ..., n+1\}$ and $j \in \{1, ..., m+1\}$

$W(v_i, w_j) = (u_{ij} - 1)^2 - u_{ij}^2 = 1 - 2u_{ij}$

$W(w_j, v_i) = (u_{ij} + 1)^2 - u_{ij}^2 = 1 + 2u_{ij}$

Now consider some $G_v \subset K_{n+1,m+1}(u)$ with the same weights for an arbitrary $v \in RV(A_n \otimes A_m)$. Then by construction, we have:
$W(G_v) = \sum_{i,j:v_{ij}\neq 0} 1 + 2v_{ij}.u_{ij} = \langle v, v\rangle + 2\langle v, u\rangle = \|u+v\|^2 - \|u\|^2.$
So $\|(x+v)-t\| < \|x-t\|$ for a $v \in RV(A_n \otimes A_m)$ if and only if $G_v \subset K_{n+1,m+1}(u)$ has negative weight. By Proposition 3.2.7, the Voronoi relevant vectors of $A_n \otimes A_m$ are precisely all $v \in X \setminus \{0\}$ such that G_v consists of a single simple cycle. Thus every simple cycle of length at least 4 in $K_{n+1,m+1}$ corresponds to a Voronoi relevant vector. So the problem of finding a $v \in RV(A_n \otimes A_m)$ such that $\|(x+v) - t\| < \|x - t\|$ is equivalent to finding a simple cycle of length at least 4 with negative weight in $K_{n+1,m+1}$. Note that because

$W(v_i, w_j) + W(w_j, v_i) = 2 \geq 0$ for all $i \in \{1, ..., n+1\}$ and $j \in \{1, ..., m+1\}$, there exists no simple cycles of length 2. Therefore, we just need to find a simple cycle of negative weight. this can be done by Bellman-Ford algorithm in $O(C.|E|) = O(\min\{n+m\}nm)$ operations, where $C = 2\min\{n+1, m+1\}$ bounds the length of the cycles considered. The construction of the graph itself can easily be done in $O(n+m+nm)$ operations and thus adds nothing to the complexity. The Bellman-Ford algorithm also detects if simple negative weight cycles exist or not [15]. □

Lemma 3.2.9. *For any $t \in span(A_n \otimes A_m)$, we can find an $x \in A_n \otimes A_m$ such that $\|x - t\| \leq 2\sqrt{(n+1)(m+1)}$ in $O(nm)$ operations.*

A polynomial CVP algorithm for the lattice $A_n \otimes A_m$ is given as below:

Algorithm 2 A polynomial CVP algorithm for the lattice $A_n \otimes A_m$.

Require: $n, m, d \geq 1$ and $t = \sum_{i,j} a_{ij} b^{ij} \in \mathtt{span}(A_n \otimes A_m)$ with $a_{ij} \in 2^{-d}\mathbb{Z}$

Ensure: a closest vector $\mathbf{x}$ to $\mathbf{t}$ in $A_n \otimes A_m$

```
Find (a_qr)_{q,r} such that t = Σ_{qr} a_qr b^qr;
a := Σ_{q,r} ⌊a_qr⌉ b^qr, b := a;
for i = 1, ..., d (outer loop) do
        t_i := Σ_{q,r} 2^{-i} ⌊2^i a_qr⌉ b^qr;
            construct weighted K_{n+1,m+1} (with u := a - t_i);
            while K_{n+1,m+1}(a - t_i) has a negative cycle G_u do (inner loop)
             a := a + u;
        else
             break;
        x_i := a;
end for
    x_d is a closest vector to t;
```

Theorem 3.2.10. *Given a target $t = \sum_{i,j} a_{ij} b^{ij} \in span(A_n \otimes A_m)$ with all $a_{ij} \in 2^{-d}\mathbb{Z}$ and with $d \geq 1$ we can find a closest vector to t in $A_n \otimes A_m$ in $O(d.(nm)^2(n+m))$ operations.*

Proof. Let $a_{kl} \in 2^{-d}\mathbb{Z}$ such that $t = \sum_{k,l} a_{kl} b^{kl} \in 2^{-d}\mathbb{Z}^{(n+1)(m+1)}$. These a_{kl} can be done in time $O(nm)$. Let $t_i = \sum_{k,l} 2^{-i} \lfloor 2^i a_{kl} \rceil b^{kl}$ for $i = 0, ..., l$; so $t_l = t$. These can be also be computed in time $O(nm)$ each as each b^{kl} has only 4

nonzero cœfficient there are at most 4 basis elements that are non zero there. Note that if our current target is t_i and our current best approximation is $a \in A_n \otimes A_m$, we will improve in every iteration with at least 2^{-i+1} between squared distances if we improve at all as for a relevant vector $v \in RV(A_n \otimes A_m)$ we have $\|a+v-t_i\|^2 - \|a-t_i\|^2 = 2\langle a-t_i, v\rangle + \langle v, v\rangle \in 2^{-i+1}\mathbb{Z}^{(n+1)(m+1)}$; because a and v are integer vectors, and $t_i \in 2^{-i}\mathbb{Z}^{(n+1)(m+1)}$, when searching a closest vector to t_i we start with the approximation x_{i-1}. To bound the number of iterations of the inner loop to get x_i, we need the following bound for $i \geq 1$:
$\|t_{i-1} - x_{i-1}\|^2 - \|t_i - x_i\|^2 = (\|t_i - x_{i-1}\| - \|t_i - x_i\|)(\|t_i - x_{i-1}\| + \|t_i - x_i\|)$;
$\leq (\underline{\|t_{i-1} - x_{i-1}\| + \|e_i\| + \|t_i - x_i\|})(\|t_{i-1} - x_{i-1}\| + \|e_i\| - \|t_i - x_i\|)$; since we have $\|t_i - x_i\| \leq 2\sqrt{(n+1)(m+1)}$ for all $i \geq 0$ by Lemma 3.2.9;
we get, $\|t_i - x_{i-1}\|^2 - \|t_i - x_i\|^2 \leq (dist(t_{i-1}, A_n \otimes A_m) + 2^{-i+2}\sqrt{(n+1)(m+1)} + 2\sqrt{(n+1)(m+1)})(dist(t_{i-1}, A_n \otimes A_m) + 2^{-i+2}\sqrt{(n+1)(m+1)} - dist(t_i, A_n \otimes A_m)$;
so $\|t_i - x_{i-1}\|^2 - \|t_i - x_i\|^2 \leq (4\sqrt{(n+1)(m+1)} + 2^{-i+2}\sqrt{(n+1)(m+1)})$
$(dist(t_{i-1}, A_n \otimes A_m) - dist(t_i, A_n \otimes A_m) + 2^{-i+2}\sqrt{(n+1)(m+1)})$;
then $\|t_i - x_{i-1}\|^2 - \|t_i - x_i\|^2 \leq (4 + 2^{-i+2})\sqrt{(n+1)(m+1)}(dist(t_{i-1}, A_n \otimes A_m) - dist(t_i, A_n \otimes A_m) + 2^{-i+2}\sqrt{(n+1)(m+1)})$;
thus $\|t_i - x_{i-1}\|^2 - \|t_i - x_i\|^2 \leq (4 + 2^{-i+2})\sqrt{(n+1)(m+1)}(\|t_{i-1} - t_i\| + 2^{-i+2}\sqrt{(n+1)(m+1)})$;
i.e $\|t_i - x_{i-1}\|^2 - \|t_i - x_i\|^2 \leq (4 + 2^{-i+2})\sqrt{(n+1)(m+1)}(2^{-i+2}\sqrt{(n+1)(m+1)} + 2^{-i+2}\sqrt{(n+1)(m+1)})$;
i.e $\|t_i - x_{i-1}\|^2 - \|t_i - x_i\|^2 \leq (4 + 2^{-i+2})\sqrt{(n+1)(m+1)}(2^{-i+3}\sqrt{(n+1)(m+1)})$;
thus $\|t_i - x_{i-1}\|^2 - \|t_i - x_i\|^2 \leq 16.2^{-i+1} + 8(2^{-i+1})(2^{-i+1})(n+1)(m+1)$;
therefore, $\|t_i - x_{i-1}\|^2 - \|t_i - x_i\|^2 \leq 16.2^{-i+1}(1 + 2^{-i})(n+1)(m+1)$;
so for fixed i the inner loop starts with $a = x_{i-1}$ and improves this approximation until $\|t_i - a_s\| = \|t_i - x_i\|$. So we get the following:
$\|t_i - x_{i-1}\|^2 = \|t_i - a\|^2 < \|t_i - a_1\|^2 < ... < \|t_i - a_s\|^2 = \|t_i - x_i\|^2$
and because $\|t_i - x_{i-1}\|^2 - \|t_i - x_i\|^2 \leq 16.2^{-i+1}(1 + 2^{-i})(n+1)(m+1)$ and in every iteration this decreases with at least 2^{-i+1} there can be at most $16.(1 + 2^{-i}) + 1$ iterations (+1 for the final check) for every $i \geq 1$. So giving a closest vector x_{i-1} to t_{i-1} we can find a closest vector x_i to t_i in $O(nm)$ iterations. By Lemma 3.2.8, each iteration takes $O(nm \min n, m)$ operations. By Lemma 3.2.9, we can find a $a \in A_n \otimes A_m$ such that $\|t_0 - a\|^2 \leq 4(n+1)(m+1)$ and thus, $\|t_0 - a\|^2 - \|t_0 - x_0\|^2 \leq 4(n+1)(m+1)$; and as difference decreases with at least $2^{-0+1} = 2$ every iteration, the number of iterations to obtain x_0 from the first approximation is also in $O(nm)$ and thus the total number of

operations to find x_0 is in $O((nm)^2 \min n, m)$. This changes nothing to the total complexity and thus we can find a closest vector to $t_d = t$ in $A_n \otimes A_m$ in $A_n \otimes A_m$ in $O(d.(nm)^2 \min n, m)$ operations. $\square$

3.2.3 Solving the closest vector problem in $A_n \otimes A_m \otimes A_p$ $(n, m, p \geq 1)$

In this section, we will characterize the Voronoi relevant vector in $A_n \otimes A_m \otimes A_p$ $(n, m, p \geq 1)$ in order to determine a polynomial algorithm to solve the closest vector problem in this lattice.
We will use the same techniques as for the case of the tensor product of two root lattices of type A. But in this case of the tensor product of three root lattices of type A, we will use the complete directed tripartite graph.

Definition 3.2.11. *Let $n, m, p \geq 1$, be three positives integers that are not all zero. We call root lattices $A_n \otimes A_m \otimes A_p \subseteq \mathbb{Z}^{(n+1)(m+1)(p+1)}$ of rank nmp all of the elements*
$x = (x_{111}, ..., x_{11(p+1)}, x_{121}, ..., x_{12(p+1)}, ..., x_{(n+1)(m+1)(p+1)}) \in \mathbb{Z}^{(n+1)(m+1)(p+1)}$ *satisfying the following conditions:*

$$\begin{array}{rcl} \sum_{i=1}^{n+1} x_{ijk} = & 0 & for \quad all \quad j = 1, ..., m+1 \quad and \quad k = 1, ..., p+1 \\ \sum_{j=1}^{m+1} x_{ijk} = & 0 & for \quad all \quad i = 1, ..., n+1 \quad and \quad k = 1, ..., p+1 \\ \sum_{k=1}^{p+1} x_{ijk} = & 0 & for \quad all \quad i = 1, ..., n+1 \quad and \quad j = 1, ..., m+1. \end{array}$$

We will use the indices i, j and k throughout this section.

3.2.4 Characterizing the Voronoi relevant vectors

As announced, we construct a polynomial algorithm to solve the closest vector problem for the lattice $A_n \otimes A_m \otimes A_p$. For this we characterize the Voronoi relevant vector of $A_n \otimes A_m \otimes A_p$. First we will limit our search space. Many of the results presented here are due by Léo Ducas and Wessel van Woerden [22].

Proposition 3.2.12. *For all voronoi relevant vectors $u \in A_n \otimes A_m \otimes A_p$ we have $|u_{ijk}| < 6$ for all $i = 1, ..., n+1$; $j = 1, ..., m+1$ and $k = 1, ..., p+1$.*

Proof. Let $u \in A_n \otimes A_m \otimes A_p$ be a Voronoi relevant vector. We suppose that there exists i, j, k such that $|u_{ijk}| \geq 6$; because of symmetry of the Voronoi region we can assume without loss of generality that $|u_{111}| \geq 6$. And because u

is a Voronoi relevant vector if and only if $-u$ is also a Voronoi relevant vector, we can also assume that $u_{ijk} \geq 6$.
Let $x^{ijk} \in A_n \otimes A_m \otimes A_p$ for all $i = 2, ..., n+1$; $j = 2, ..., m+1$ and $k = 2, ..., p+1$ be given by $x_{111} = x_{1jk} = x_{ij1} = x_{i1j} = 1$; $x_{11k} = x_{1j1} = x_{i11} = x_{ijk} = -1$ and 0 otherwise.
Note that $\langle x^{ijk}, x^{ijk} \rangle = 8$ for all i, j, k. Then by Definition 2, we get: $u_{111} + u_{1jk} + u_{ij1} + u_{i1j} - u_{11k} - u_{1j1} - u_{i11} - u_{ijk} = \langle u, x^{ijk} \rangle < 8$ for all $i = 1, ..., n+1$; $j = 1, ..., m+1$ and $k = 1, ..., p+1$.
also note that because these are all integers, we even have that:
$u_{111} + u_{1jk} + u_{ij1} + u_{i1k} - u_{11k} - u_{1j1} - u_{i11} - u_{ijk} \leq 7$. Summing multiple of these relations for a fixed $j = 2, ..., m+1$ gives:
$mu_{111} - mu_{11k} + mu_{i1k} - mu_{i11} + \sum\limits_{j=2}^{m+1} (u_{1jk} + u_{ij1} - u_{1j1} - u_{ijk}) \leq 7(m+1-1)$;
summing multiple of these relations for a fixed $k = 2, ..., p+1$ gives:
$mpu_{111} - mpu_{i11} + \sum\limits_{k=2}^{p+1} (mu_{i1k} - mu_{11k}) + \sum\limits_{k=2}^{p+1} (\sum\limits_{j=2}^{m+1} (u_{1jk} + u_{ij1} - u_{1j1} - u_{ijk})) \leq 7(m+1-1)(p+1-1)$;
furthermore $-mu_{i11} = \sum\limits_{k=2}^{p+1} mu_{i1k}$ and $-mu_{111} = \sum\limits_{k=2}^{p+1} mu_{11k}$;
as becomes : $mpu_{111} - mpu_{i11} - mu_{i11} + mu_{111} + \sum\limits_{k=2}^{p+1} (\sum\limits_{j=2}^{m+1} (u_{1jk} + u_{ij1} - u_{1j1} - u_{ijk})) \leq 7(m+1-1)(p+1-1)$;
furthermore, $\sum\limits_{k=2}^{p+1} \sum\limits_{j=2}^{m+1} (u_{1jk} + u_{ij1} - u_{1j1} - u_{ijk}) = \sum\limits_{k=2}^{p+1} (-u_{11k} - u_{i11} + u_{111} + u_{i1k}) = u_{111} - pu_{i11} + pu_{111} - u_{i11}$;
so the inequation becomes: $mpu_{111} - mpu_{i11} - mu_{i11} + mu_{111} + u_{111} - pu_{i11} + pu_{111} - u_{i11} \leq 7(m+1-1)(p+1-1)$;
thus $(m+1)(p+1)(u_{111} - u_{i11}) \leq 7(m+1-1)(p+1-1)$; so $u_{111} - u_{i11} \leq \dfrac{7(m+1-1)(p+1-1)}{(m+1)(p+1)}$;
by hypothesis we have $u_{111} \geq 6$, then we now get:
$u_{i11} \geq \dfrac{-7(m+1-1)(p+1-1)}{(m+1)(p+1)} + 6$; and thus $u_{i11} \geq -1 + \dfrac{7(m+1+p+1-1)}{(m+1)(p+1)}$;
we also have $7((m+1)(p+1)-1) > (m+1)(p+1)$ for all $n+1, m+1 \geq 3$
so $u_{i11} \geq 0$ for all $i = 2, ..., n+1$ and $u_{111} \geq 6$;
but in that case: $0 = \sum\limits_{i=1}^{n+1} u_{i11} \geq 6 + 0 + ... + 0 = 6$ which gives a contradiction.
so $|u_{111}| < 6$ $\square$

Remark 3.2.13. *From the Proposition 3.2.12 we can deduce that all Voronoi relevant vectors of $A_n \otimes A_m \otimes A_p$ must lie in*
$X := \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}^{(n+1)(m+1)(p+1)} \cap (A_n \otimes A_m \otimes A_p)$.
As for the case of two root lattices we have determined the set of coordinates of the Voronoi relevant vector in $A_n \otimes A_m$ but the characterization of its ele-

ments according to a certain subgraphs of the complete directed tripartite graph $K_{n+1,m+1,p+1} = (V, E)$ is very difficult. This is the reason why, we will use the associativity of the lattice of type A and the results obtained by Léo Ducas and wessel van Woerden in [22] to solve CVP in the tensor product of more than two lattices of type A.

Since $A_n \otimes A_m$ is a sub lattice of root lattice $A_{(n+1)(m+1)-1}$, and that the lattices are non commutative, we can give the correspondence between the elements of Y and certain subgraphs of the complete tripartite labelle graph $K_{n+1,m+1,p+1} = (V, E)$. We label the $n+1$ nodes $V_1 := \{u_1, ..., u_{n+1}\}$, $V_2 := \{v_1, ..., v_{m+1}\}$ and $V_3 := \{w_1, ..., w_{p+1}\}$; and we let the coefficient $t_{ijk} \in Y$ correspond to the triplet $(u_i, v_j, w_k) := (u_i, v_j) \wedge (v_j, w_k)$ of nodes of $K_{n+1,m+1,p+1}$.

Definition 3.2.14. *Let $t \in \{-1, 0, 1\}^{nmp}$ be given. Let $K_{n+1,m+1,p+1}$ be the complete directed tripartite graph with $n+1$ nodes $u_1, ..., u_{n+1}$; $m+1$ nodes $v_1, ..., v_{m+1}$ and $p+1$ nodes $w_1, ..., w_{p+1}$. We define the subgraph $G_t = (V_t, E_t) \subset K_{n+1,m+1,p+1}$ corresponding to t where E_t consists of the following directed edges.*

- *The edge $(u_i, v_j, w_k) = (u_i, v_j) \wedge (v_j, w_k)$ for each $t()_{ijk}$ that has value 1;*
- *The edge $(u_i, v_j, w_k) = (u_i, v_j) \wedge (w_k, v_j)$ for each t_{ijk} that has value 1;*
- *The edge $(u_i, v_j, w_k) = (v_j, u_i) \wedge (v_j, w_k)$ for each t_{ijk} that has value -1;*
- *The edge $(u_i, v_j, w_k) = (v_j, u_i) \wedge (w_k, v_j)$ for each t_{ijk} that has value -1;*

and V_t as all nodes with non zero in-or outdegree. Note that the condition for $\{-1, 0, 1\}^{nmp}$ to be part of $A_n \otimes A_m \otimes A_p$ corresponds to the fact for every node of G_t the difference between the indegree and the outdegree must be even.

From Definition 3.2.14, we can give the following lemma.

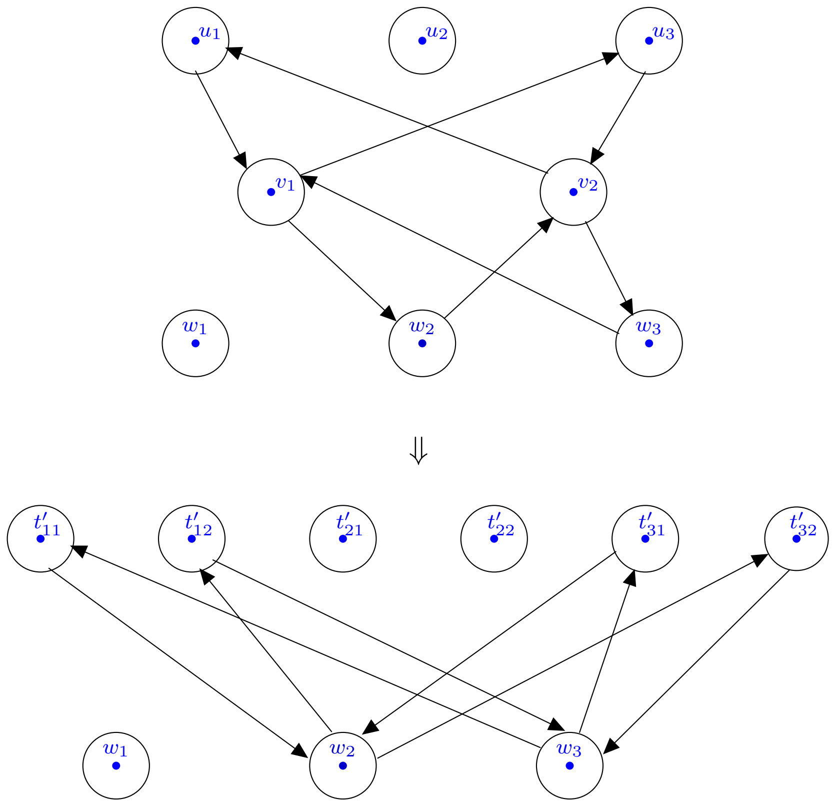
Lemma 3.2.15. *For any complete directed tripartite graph $K_{n+1,m+1,p+1}$, we can define an equivalent sub graph $G_{t'} = (V_{t'}, E_{t'}) \subset K_{(n+1)(m+1)-1,p+1}$ corresponding to t' where $E_{t'}$ consists of the following directed edges.*

- *The edge (t'_{ij}, w_k) for each t'_{ijk} that has value 1 if $t'_{ij} = (u_i, v_j)$;*
- *The edge (t'_{ij}, w_k) for each t'_{ijk} that has value -1 if $t'_{ij} = (v_j, u_i)$.*

In this case, the difference between the indegree and the outdegree of every node of $G_{t'}$ must be even.

Figure 3.2: Example graph G'_t corresponding to
$t' = (0, 1, -1, 0, -1, 1, 0, 0, 0, 0, 0, 0, 0, 1, -1, 0, -1, 1) \in A_2 \otimes A_1 \otimes A_2$

Example 3.2.16.



Therefore, we will use the same techniques as for the case of the tensor product of two root lattices of type A to solve the problem in three root lattices of type A.

Proposition 3.2.17. *Now consider $Y := \{-1, 0, 1\}$. The Voronoi relevant vectors of $A_n \otimes A_m \otimes A_p$ are precisely all $s \in Y \setminus \{0\}$ such that G_s consists of a simple cycle.*

Proof. Just use (Theorem 2, [22]) and associativity of tensor product in root lattices of type A. □

From Theorem 2, [22] we can deduce that the number of Voronoi relevant vectors of
$A_n \otimes A_m \otimes A_p$ is equal to:

$$\sum_{i=2}^{min\{(n+1)(m+1),(p+1)\}} \binom{(n+1)(m+1)}{i}\binom{p+1}{i}.i!.(i-1)!$$

3.2.5 Finding the closest vector in $A_n \otimes A_m \otimes A_p$

The Voronoi relevant vectors of $A_n \otimes A_m \otimes A_p$ being characterized, we will in the following present a polynomial algorithm allowing to solve CVP in this type of lattice.

Lemma 3.2.18. *Let $x \in A_n \otimes A_m \otimes A_p$, and let $t \in \mathtt{span}(A_n \otimes A_m \otimes A_p)$ be our target. If there exists a Voronoi relevant vector $u \in RV(A_n \otimes A_m \otimes A_p)$ such that $\|(x+u)-t\| < \|x-t\|$ we can find such a Voronoi relevant vector in $O\left(((n+1)(m+1)-1+p)((n+1)(m+1)-1)p\right)$ operations. If it doesn't exist this will be detected by the algorithm.*

Proof. Just use (Lemmas 3 and 8 [22]) and the associativity of tensor product in root lattices of type A. □

Lemma 3.2.19. *Let $b^{ijk} \in A_n \otimes A_m \otimes A_p$. be given by:*
- $b^{ijk}_{i,j,k} = b^{ijk}_{i,j+1,k+1} = b^{ijk}_{i+1,j,k+1} = b^{ijk}_{i+1,j+1,k} = 1$;
- $b^{ijk}_{i,j,k+1} = b^{ijk}_{i,j+1,k} = b^{ijk}_{i+1,j,k} = b^{ijk}_{i+1,j+1,k+1} = -1$;
- *and* 0 *otherwise for all* $i = 1, ..., n+1$; $j = 1, ..., m+1$ *and* $k = 1, ..., p+1$.

Note that
$B := \left\{b^{ijk} : i = \{1, ..., n+1\};\ j = \{1, ..., m+1\} \text{ and } k = \{1, ..., p+1\}\right\}$
is a basis of $A_n \otimes A_m \otimes A_p$. Because the basis B is so sparse we can efficiently encode elements in this basis.

Lemma 3.2.20. *For any $t \in \mathtt{span}(A_n \otimes A_m \otimes A_p)$ we can find an $x \in A_n \otimes A_m \otimes A_p$ such that*
$\|x-t\| \leq 2\sqrt{(n+1)(m+1)(p+1)}$ *in $O(((n+1)(m+1)-1)p))$ operations.*

Proof. Just use (Lemma 7, [22]) and the associativity of tensor product in root lattices of type A. □

In Lemma 5 [22], if $\sum\limits_{i,j,k} a_{ij} b^{ijk} \in \mathtt{span}(A_n \otimes A_m \otimes A_p) \cap (2^{-d}\mathbb{Z}^{(n+1)(m+1)(p+1)})$ from the transformation, it is clear that $a_{ij} \in 2^{-d}\mathbb{Z}$. Since $A_n \otimes A_m \otimes A_p$ has only integer vectors, we can say that if
$t \in 2^{-d}\mathbb{Z}^{(n+1)(m+1)(p+1)}$ then the squared distance to the target will in each iteration improve with at least 2^{-i+1} which is exactly what we need to bound the number of iterations. □
A polynomial CVP algorithm for the lattice $A_n \otimes A_m \otimes A_p$ is given as below:

Algorithm 3 A polynomial CVP algorithm for the lattice $A_n \otimes A_m \otimes A_p$.

Require: $n, m, p, d \geq 1$ and $t = \sum_{i,j,k} a_{ijk} b^{ijk} \in \texttt{span}(A_n \otimes A_m \otimes A_p.)$ with $a_{ijk} \in 2^{-d}\mathbb{Z}$

Ensure: a closest vector $\mathbf{x}$ to $\mathbf{t}$ in $A_n \otimes A_m \otimes A_p$.

```
Find (a_pqr)_{p,q,r}, such that t = Σ_{pqr} a_pqr b^pqr;
a := Σ_{p,q,r} ⌊a_pqr⌉ b^pqr, b := a;
for i = 1, ··· , d(outer loop) do
        t_i := Σ_{p,q,r} 2^{-i} ⌊2^i a_pqr⌉ b^pqr;
            construct weighted K_{(n+1)(m+1),(p+1)} (with s := a − t_i);
            a := a + s;
        else
        break;
        x_i := a;
end for
    x_d is a closest vector to t;
```

Proposition 3.2.21. *Given a target $t = \sum_{i,j} a_{ij} b^{ij} \in span(A_n \otimes A_m \otimes A_p)$ with all $a_{ij} \in 2^{-d}\mathbb{Z}$ and with $d \geq 1$ we can find a closest vector to t in $A_n \otimes A_m \otimes A_p$ in*
$O(d.(((n+1)(m+1)-1)p)^2 min\{(n+1)(m+1)-1), p\})$ arithmetic operations with the previous algorithm.

Proof. Just use (Theorem 3, [22]) and the associativity of tensor product in root lattice of type A. □

3.3 Closest Vector Problem in $A_{n_1} \otimes A_{n_2} \otimes ... \otimes A_{n_k}$

According to the previous remark, we can generalize the resolution of CVP in the tensor product of k root lattices of type A.
Let k lattices $A_{n_1}, ..., A_{n_k}$ of type A.

Definition 3.3.1. *Let $n_1, ..., n_k \geq 1$, be k positive integers that are not all zero. We call root lattice $A_{n_1} \otimes A_{n_2} \otimes ... \otimes A_{n_k} \subset \mathbb{Z}^{(n_1+1)...(n_k+1)}$ of rank $n_1 n_2 ... n_k$ all of the elements*
$x = (x_{11...1}, x_{11...1(n_k+1)}, x_{121...1}, ..., x_{(n_1+1)...(n_k+1)}) \in \mathbb{Z}^{(n_1+1)...(n_k+1)}$ *satisfying*

conditions:

$$\begin{array}{rcl} \sum_{i^{(1)}=1}^{n_1+1} x_{i^{(1)}i^{(2)}...i^{(k)}} &=& 0 \quad for \quad all \quad i^{(2)} \in \{1,...,n_2+1\} \quad ... \quad i^{(k)} \in \{1,...,n_k+1\} \\ \sum_{i^{(2)}=1}^{n_2+1} x_{i^{(1)}i^{(2)}...i^{(k)}} &=& 0 \quad for \quad all \quad i^{(1)} \in \{1,...,n_1+1\} \quad ... \quad i^{(k)} \in \{1,...,n_k+1\} \\ ... && .. \quad ... \quad ... \quad\quad ... \\ \sum_{i^{(k)}=1}^{n_k+1} x_{i^{(1)}i^{(2)}...i^{(k)}} &=& 0 \quad for \quad all \quad i^{(1)} \in \{1,...,n_1+1\} \quad ... \quad i^{(k-1)} \in \{1,...,n_{k-1}+1\}. \end{array}$$

We will use the indices $i^{(1)},...,i^{(k)}$ throughout this section.

We note that by gradually regrouping these lattices, and two by two, and by using the associativity of the tensor product, solving closest vector problem in $A_{n_1} \otimes A_{n_2} \otimes ... \otimes A_{n_k}$ amounts to solving the same problem in $(A_{n_1} \otimes A_{n_2}) \otimes A_{n_3} \otimes ... \otimes A_{n_k}$.
Step by step, solving this problem in $A_{n_1} \otimes A_{n_2} \otimes ... \otimes A_{n_k}$ could be reduced to solving it in $A_{n_1(n_2+1)...(n_{k-1})-1} \otimes A_{n_k}$.
The previous Section illustrates well the case for $k = 3$. For the general case, we just have to use the same technique, and we will obtained a CVP algorithm for this general case. This algorithm runs in
$O\left(d.(((n_1+1)...(n_{k-1}+1)-1)n_k)^2 \min\{(n_1+1)...(n_{k-1}+1)-1, n_k\}\right)$ (where $d \geq 1$) arithmetic operations.

3.4 Closest vector problem for some root Lattice of type D

Before going on the characterization of the vectors of the root lattice $D_n \otimes D_m$, we will present a polynomial algorithm which solves the CVP in the root lattice D_n.

3.4.1 The closest vector problem in D_n

Given $x \in \mathbb{R}^n$, the closest point to x in D_n is whichever of $f(x)$ and $g(x)$ having an even sum of coordinates (one will have an even sum, the other an odd sum), where the functions f and g are defined as follows: For an arbitrary $x_i \in \mathbb{R}$, we define the functions $f(x_i)$ and $w(x_i)$ for all $i = 1,...,n$ as follows:

- if $x_i = 0$ then $f(x_i) = 0$ and $w(x_i) = 1$
- if $0 < m + \frac{1}{2} < x_i < m + 1$ then $f(x_i) = m$ and $w(x_i) = m + 1$

- if $-m - \frac{1}{2} \leq x_i \leq -m$ then $f(x_i) = -m$ and $w(x_i) = -m - 1$
- if $0 < m + \frac{1}{2} < x_i < m + 1$ then $f(x_i) = m + 1$ and $w(x_i) = m$
- if $-m - 1 < x_i < -m - \frac{1}{2}$ then $f(x_i) = -m - 1$ and $w(x_i) = -m$

We also write $x_i = f(x_i) + \delta(x_i)$, so that $|\delta(x_i)| \leq \frac{1}{2}$ is the distance from x_i to the nearest integer.
Given that $x = (x_1, ..., x_n) \in \mathbb{R}^n$, let $k(1 \leq k \leq n)$ such that $|\delta(x_k)| \leq |\delta(x_i)|$ for all $1 \leq i \leq n$ and $|\delta(x_k)| = |\delta(x_i)|$ implies $k \leq i$. Then $f(x) = (f(x_1), f(x_2), ..., f(x_k), ..., f(x_n))$ and $g(x)$ is defined by:
$g(x) = (f(x_1), f(x_2), ..., w(x_k), ..., f(x_n))$.

3.4.2 Characterisation of the vectors of the root lattice $D_n \otimes D_m$

We will start this section by the characterization of the vectors of the root lattice $D_n \otimes D_m$ $(n, m \geq 2)$ as below. We first recalls the definition of the tensor product:

Definition 3.4.1. *Let $\Lambda_1 \subseteq \mathbb{R}^{n_1}$ and $\Lambda_2 \subseteq \mathbb{R}^{n_2}$ be lattices of respectively ranks n_1 and n_2,*
let $a_1, ..., a_{n_1} \in \mathbb{R}^{n_1}$ and $b_1, ..., b_{n_2} \in \mathbb{R}^{n_2}$ be their respective bases. The tensor product $\Lambda_1 \otimes \Lambda_2 \subset \mathbb{R}^{n_1 n_2}$ is defined as the lattice with basis $\{a_i \otimes b_j : i \in \{1, ..., n_1\},\ \ j \in \{1, ..., n_2\}\}$.
Here $x \otimes y = (x_1, ..., x_{n_1}) \otimes (y_1, ..., y_{n_2})$ with $x \in \mathbb{R}^{n_1}$ and $y \in \mathbb{R}^{n_2}$ can be seen as an element of $\mathbb{R}^{n_1 n_2}$ as follows : $(x_1 y_1, x_1 y_2, ..., x_1 y_{n_1}, x_2 y_1, ..., x_{n_1} y_{n_2}) \in \mathbb{R}^{n_1 n_2}$.

The root lattice $D_n \otimes D_m \subseteq \mathbb{Z}^{nm}$ $(n, m \geq 2)$ consists of all elements $x = (x_{11}, ..., x_{1m}, x_{21}, ..., x_{2m}, ..., x_{n1}, ..., x_{nm}) \in \mathbb{Z}^{nm}$ satisfying the following conditions:

(1) $\sum_{i=1}^{n} x_{ij}$ *even for all* $j = 1, ..., m$

(2) $\sum_{j=1}^{m} x_{ij}$ *even for all* $i = 1, ..., n.$

(The notation $x = (x_{11}, ..., x_{1m}, x_{21}, ..., x_{2m}, ..., x_{n1}, ..., x_{nm})$ above, means that there exist two vectors $u = (u_1, ..., u_n) \in D_n$ and $v = (v_1, ..., v_m) \in D_m$ such that: $x_{ij} = u_i v_j$ for $i = 1, ..., n$ and $j = 1, ..., m.$)

Indeed, we have $(x_{11}, ..., x_{1m}, ..., x_{2m}, ..., x_{n1}, ..., x_{nm}) = (u_1 v_1, ..., u_1 v_m, u_2 v_1, ..., u_2 v_m, ..., u_n v_1, ..., u_n v_m) \in D_n \otimes D_m$. Since the sums

$\sum\limits_{i=1}^{n} u_i$ and $\sum\limits_{j=1}^{m} v_j$ are even, then $\sum\limits_{i=1}^{n} u_i v_j$ is even for all $j = 1, ..., m$ and $\sum\limits_{j=1}^{m} u_i v_j$ is even for all $i = 1, ..., n$.

Remark 3.4.2. *Let D_n and D_m $(n, m \geq 2)$be two root lattices. Then D_{nm} is a full rank sub lattice of the lattice $D_n \otimes D_m$.*
Indeed, the vector $x = (0, 0, 2, 1, 1, 0, -1, 1)$ is the vector of the root lattice D_8 because $0+0+2+1+1+0-1+1 = 4$, which is even. But this vector is not in the root lattice $D_2 \otimes D_4$ because $\sum\limits_{j=1}^{4} x_{1j} = x_{11}+x_{12}+x_{13}+x_{14} = 0+0+2+1 = 3$, which is odd.

Lemma 3.4.3. *(Basis of $D_n \otimes D_m$) Let D_n and D_m $(n, m \geq 2)$ be two root lattices,*
the basis $B_{n\otimes m} := \{b^{ij} : \; i = 1, ..., n \;$ and $\; j = 1, ..., m\}$ of the root lattice $D_n \otimes D_m$ is given by:

- $b^{11}_{1,1} = b^{11}_{1,2} = b^{11}_{2,1} = b^{11}_{2,2} = 1$
- $b^{i1}_{i-1,1} = b^{i1}_{i-1,2} = 1$; $b^{i1}_{i,2} = b^{i1}_{i;1} = -1$ *for all* $i = 2, ..., n$
- $b^{1j}_{1,j-1} = b^{1j}_{2,j-1} = 1$; $b^{1j}_{1,j} = b^{1j}_{2,j} = -1$ *for all* $j = 2, ..., m$
- $b^{ij}_{i-1,j-1} = b^{ij}_{i,j} = 1$; $b^{ij}_{i-1,j} = b^{ij}_{i,j-1} = -1$ *for all* $i = 2, ..., n$ *and* $j = 2, ..., m$
- 0 *otherwise*

3.4.3 A polynomial algorithm for solving the CVP in $D_n \otimes D_m$

We first present a general description of our CVP efficient algorithm in $D_n \otimes D_m$ $(n, m \geq 2)$ as below:

Description of the algorithm

This algorithm takes as input a vector of a linear space spanned $span(D_n \otimes D_m)$ (where D_n and D_m are two root lattices of type D with $n, m \geq 2$) and returns a closest vector to this vector in $D_n \otimes D_m$ as follows: Given a vector $t = (t_{11}, ..., t_{1m}, t_{21}, ..., t_{2m}, ..., t_{n1}, ..., t_{nm})$ of $span(D_n \otimes D_m) \subseteq \mathbb{R}^{nm}$.
We will start by determining the closest vector to t in the root lattice D_{nm}.
To do this, we will calculate the functions:
$f(t) = (f(t_{11}), ..., f(t_{1m}), f(t_{21}), ..., f(t_{2m}), ..., f(t_{n1}), ..., f(t_{nm}))$ and
$g(t) = (f(t_{11}), ..., f(t_{k(l-1)}), w(t_{kl}), f(t_{k(l+1)}), ..., f(t_{nm}))$ (where $f(t_{ij}) = \lfloor t_{ij} \rceil$

for all $i = 1, ..., n$ and $j = 1, ..., m$; and the function g is obtained by proceeding as in the case of a single root lattice of type D [13]). Given that the two functions f and g differ by only one component, and by the value 1, then either the sum of the function's coordinates f or g will be even .
Then, if the sum of all the coordinates of $f(t)$ is even then $h := f$, else $h := g$. Thus, $h \in D_{nm}$. After determining the closest vector $h \in D_{nm}$ of t, the closest vector to h in $D_n \otimes D_m$ is obtained as follows:
We carry out the sums $\sum_{i=1}^{n} h(t_{ij})$ for all $j = 1, ..., m$ and $\sum_{j=1}^{m} h(t_{ij})$ for $i = 1, ..., n$. If all these sums are even, then $h \in D_n \otimes D_m$. Therefore, $x := h$. Else we proceed as follow:
Then we initialize the counters c, d, α and β as follows: $c := 0$, $d := 0$, $\alpha := 1$ and $\beta := 1$. We calculate for each $i = 1, ..., n$ the sums $\sum_{j=1}^{m} h(t_{ij})$. Thus, for $i = 1, ..., n$ if $\sum_{j=1}^{m} h(t_{ij})$ odd, then $c := c+1$; $u_\alpha := \sum_{j=1}^{m} h(t_{ij})$ and $\alpha = \alpha+1$. We calculate also for each $j = 1, ..., m$ the sums $\sum_{i=1}^{n} h(t_{ij})$. As above, for $j = 1, ..., m$ if $\sum_{i=1}^{n} h(t_{ij})$ odd, then $d := d + 1$; $v_\beta := \sum_{j=1}^{m} h(t_{ij})$ and $\beta = \beta + 1$.
After calculating all the sums above, if $c = 0$ and $d = 0$ then $x := h$. Else, for each $r = 1, ..., c$ we denote by $f(h_{u_\alpha})$ and $g(h_{u_\alpha})$ the corresponding functions to the vector h as defined in Section 3.4.1. Similarly, for each $s = 1, ..., d$ we denote by $f(h_{v_\beta})$ and $g(h_{v_\beta})$ the corresponding functions to the vector h. Here, the functions $f(h_{u_\alpha})$ and $g(h_{u_\alpha})$ are associated with the vector h whose sum of the coordinates is equal to u_α. In the same way, the functions $f(h_{v_\beta})$ and $g(h_{v_\beta})$ are associated with the vector h whose sum of the coordinates is equal to v_β.
Thus, for all u_α and v_β there exists a single common function of which all the sums of the coordinates are even. We will denote by q this function.
At the end of all these operations, we get the vector $x := q$.
This process is performed at most $(n+m)$ times until all the sums $\sum_{i=1}^{n} h(t_{ij})$ for all $j = 1, ..., m$ and $\sum_{j=1}^{m} h(t_{ij})$ for $i = 1, ..., n$ are even. Thus, the new coordinates of the function that we obtain is the component of the vector $x \in D_n \otimes D_m$. An such x is the closest vector of $t \in span(D_n \otimes D_m)$ in $D_n \otimes D_m$.

Algorithm 4 A CVP algorithm for the lattice $D_n \otimes D_m$.

Require: $n, m \geq 2$ and $t = (t_{11}, ..., t_{nm}) \in \texttt{span}(D_n \otimes D_m)$.

Ensure: a closest vector x to t in $D_n \otimes D_m$.

1: $f1 := (\lfloor t_{11} \rceil, ..., \lfloor t_{nm} \rceil)$;
2: $g1 := (f(t_{11}), ..., f(t_{k(l-1)}), f(w_{kl}), f(t_{k(l+1)}), ..., f(t_{nm}))$; (where w_{kl} is defined as in Section 3.4.1);
3: $u = [0, ..., 0]$; $v = [0, ..., 0]$; $c := 0$; $d := 0$;
4: **if** $\sum_{i,j} f(t_{ij})$ even **then**
5:     $p := f1$;
6:     **if** $p := g1$;
7: **end if**
8: **for** $i = 1, \cdots, n$ **do**
9:     $a := \sum_{j=1}^{m} p_{ij}$;
10:     **if** a odd **then**
11:         $c := c + 1$; $u_c := a$;
12:     **end if**
13: **end for**
14: **for** $j = 1, \cdots, m$ **do**
15:     $b := \sum_{i=1}^{n} p_{ij}$;
16:     **if** b odd **then**
17:         $d := d + 1$; $v_d := b$;
18:     **end if**
19: **end for**
20: **if** $c = 0$ and $d = 0$ **then**
21:     $x := p$;
22: **end if**
23: **for** $\alpha = 1, \cdots, c$ and $\beta = 1, \cdots, d$ **do**
24:     compute $f(p_{u_\alpha})$; $g(p_{u_\alpha})$; $f(p_{v_\beta})$; $g(p_{v_\beta})$; (see Subsection 3.4.3)
25:     $x := q$;(see Complexity Analysis 3.4.3 below)
26: **end for**
27: x is a closest vector of x in $D_n \otimes D_m$;

Complexity Analysis

About the complexity of this algorithm, we have:
From line 1 to line 2, we have 2 elementary operations. Indeed, we have only 2 assignments in these steps.

Line 3 has 4 elementary operations. Indeed, we have 4 assignments in this step.
From line 4 to line 8 we have 2 elementary operations. Indeed, we have 1 comparison and 1 assignment.
From line 9 to line 15, we have at most $3n$ elementary operations. Indeed, we have at most 3 operations inside the loop for which goes from 1 to n.
From line 16 to line 22, we have at most $3m$ elementary operations. Indeed, we have at most 3 operations inside the loop for which goes from 1 to m.
From line 23 to line 24, we have at most 3 elementary operations.
From line 26 to line 29, we have $n+m$ operations. Indeed, q is the vector whose coordinates are made up of a part of the coordinates whose sum is even in line 10 of our algorithm, and the rest of the coordinates of q supplemented by the coordinates obtained after line 27 of our algorithm. In this step, the algorithm uses Section 3.4.1 to determine each sub-coordinate for which the sub-vectors of each block are close to the associated target sub-vectors. Indeed, by determining the values whose distances with that of the associated sub-blocks are minimum, we will globally obtain the closest vector to the initial target vector. Given that the only operations used here are the comparisons and the additions, and that we have at most n blocks according to the index i, and at most m blocks according to the index j.
Thus we will have at most $2+4+4+3n+3m+3 = 13+4n+4m$ arithmetic operations;
since $\dfrac{13+4n+4m}{n+m} \longrightarrow cste$ when $n, m \longrightarrow \infty$, then the complexity of this algorithm is $O(n+m)$ arithmetic operations.

Example 3.4.4. *Let $n = m = 2$, and $x = (1.2, -1.2, -1.2, 0.6) \in \mathtt{span}(D_2 \otimes D_2)$.*
We have: $f = (1, -1, -1, 1)$, and $g = (1, -1, -1, 0)$;
since $1-1-1+1 = 0$, then $p := f = (1, -1, -1, 1) \in D_4$;
and because $\sum\limits_{i=1}^{2} p_{i1} = p_{11} + p_{21} = 1 - 1 = 0$, $\sum\limits_{i=1}^{2} p_{i2} = p_{12} + p_{22} = -1 + 1 = 0$, $\sum\limits_{j=1}^{2} p_{1j} = p_{11} + p_{12} = 1 - 1 = 0$ and $\sum\limits_{j=2}^{2} p_{2j} = p_{21} + p_{22} = -1 + 1 = 0$ then $x := p = (1, -1, -1, 1)$.
Therefore, $\boldsymbol{x} = (1, -1, -1, 1)$ is the closest vector of $t = (1.2, -1.2, -1.2, 0.6)$ in $D_2 \otimes D_2$.

Example 3.4.5. *Let $n = 3$ and $m = 2$, and*
$t = (2.8, -2.8, -2.8, 4.6, -2.9, -3.3) \in \mathtt{span}(D_3 \otimes D_2)$.
We have: $f := (3, -3, -3, 5, -3, -3)$ and $g := (3, -3, -3, 4, -3, -3)$;
since $3-3-3+5-3-3 = -4$, then $p := f = (3, -3, -3, 5, -3, -3)$;

For $i = 1, ..., 3$ we have:
$U_1 = \sum\limits_{j=1}^{2} p_{1j} = p_{11} + p_{12} = 3 - 3 = 0$; $U_2 = \sum\limits_{j=1}^{2} p_{2j} = p_{21} + p_{22} = -2 + 4 = 2$;
$U_3 = \sum\limits_{j=1}^{2} p_{3j} = p_{31} + p_{32} = -3 - 3 = -6$;
and for $j = 1, ..., 2$ we have:
$V_1 = \sum\limits_{i=1}^{2} p_{i1} = p_{11} + p_{21} + p_{31} = 3 - 3 - 3 = 3$ *and* $V_2 = \sum\limits_{i=1}^{2} p_{i2} = p_{12} + p_{22} + p_{32} = -3 + 5 - 3 = 3$;
we have V_1 and V_2 odd. For the case of V_1, we take the coordinates p_{11}, p_{21}, p_{31} and we calculate f_1 and g_1 as below:
$f_1 = (3, -3, -3)$ *and* $g_1 = (3, -2, -3)$ *where* $p_{11} = 3$, $p_{21} = -3, -2$ *and* $p_{31} = -3$.
For the case of V_2, we take the coordinates p_{12}, p_{22}, p_{32} and we calculate f_2 and g_2 as below:
$f_2 = (-3, 5, -3)$ *and* $g_2 = (-3, 4, -3)$ *where* $p_{12} = -3$, $p_{22} = 5, 4$ *and* $p_{32} = -3$;
since the sums of the coordinates of the vectors g_1 and g_2 are even, we choose $p_{21} = -2$ and $p_{22} = 4$; thus, $x := (3, -3, -2, 4, -3, -3)$.
Therefore, the vector $x = (3, -3, -2, 4, -3, -3)$ is the closest vector of
$t = (2.8, -2.8, -2.8, 4.6, -2.9, -3.3)$ in the root lattice $D_3 \otimes D_2$.

3.5 Concluding remarks

In this Chapter, we have use associativity and non commutativity of tensor product in lattices to solve the closest vector problem in the tensor product of three root lattices of type A. We have also generalized this work for the case of k ($k \geq 4$) root lattices of type A. We have also successfully constructed a polynomial algorithm to solve the closest vector problem for the case of tensor product of two root lattices D_n and D_m that we noted $D_n \otimes D_m$ ($n, m \geq 2$). Our future work will consist to generalise this algorithm to solve this problem for the case of tensor product of a finite number k of root lattices of type D_n ($n \geq 2$) which we denote by $\bigotimes\limits_{i=1}^{k} D_i$.
Our future work will consist to improve the algorithm for solving the closest vector problem in the tensor product of two and three root lattices of type A. Indeed, a tensor product of two or three root lattices is also a sub lattice of a root lattice with some particular properties. We will use the characterization of the Voronoi relevant vectors and the oriented complete $k-$graphs to solve CVP in the tensor product of k lattices of type A.

Chapter Four

Sieving algorithm for orthogonal integer lattice of dimension n

In this chapter, we propose a new sieve algorithm that we called *Orthogonal-Integer* sieve algorithm for some orthogonal integer lattices and particularly the case of integer lattices $\Lambda \subset \mathbb{Z}^n$, root lattices of type A_n $(n \geq 1)$ and of type D_n $(n \geq 2)$. In these cases, we use the famous LLL algorithm to find the shortest vector of these lattices. Indeed, in general, a sieve algorithm builds a list of short random vectors which are not necessarily in the lattice, and tries to produce short lattice vectors by taking linear combinations of the vectors in the list. But in our case, we built a list of short vectors in the lattice. From the first column of the LLL-reduced basis of the considered basis, we have the list of at least n and at most 2^n short vectors for the general case (where n is the dimension of the lattice) of orthogonal integer lattices $\Lambda \subset \mathbb{Z}^n$. For the lattices $\mathbb{Z}^n$, A_n $(n \geq 1)$ and D_n $(n \geq 2)$, we have respectively $2n$, $n(n+1)$ and $2n(n-1)$ short vectors . The proposed sieve algorithm for integer lattice $\mathbb{Z}^n$ runs in space $O(2n)$ and the *OrthogonalInteger* sieve algorithm performs $O(n2^n)$ arithmetic operations and is polynomial in space. Indeed, we give a list of all short vectors of the particular case of orthogonal integer lattices $\mathbb{Z}^n$. The proposed algorithm is polynomial and requires $O(n)$ in space. We also propose an enumeration algorithm which will allow us to obtain the list of shortest vectors in all orthogonal integer lattices $\Lambda \subseteq \mathbb{Z}^n$. This algorithm runs in $O(n2^n)$ time and can be polynomial in space and the list of short vectors obtained enables to solve the shortest independent vector problem $SIVP$ [7] for some orthogonal integer lattices. This is possible for some integer lattice $\mathbb{Z}^n$, root lattices of type D_n $(n \geq 2)$ and A_n $(n \geq 1)$ and their duals. For correctness, a Maple computer software implementation of the algorithm has been done.

This chapter is organized as follows. In Section 4.1, we recall some key concepts such as orthogonal lattice, some properties of orthogonal lattices that will be useful in the paper. In Section 4.2 we give a polynomial algorithm to determine an orthogonal integer basis for a given integer lattice. In Section 4.3 we recall Gauss sieve algorithm. Our main result of this chapter is presented in Section 4.4 where we describe a polynomial algorithm which returns a list of exactly $2n$ short vectors for the case of the orthogonal integer lattice $\mathbb{Z}^n$. We also present in Section 4.4 an algorithm which gives at least n and at most 2^n short vectors of general orthogonal integer lattices $\Lambda \subset \mathbb{Z}^n$. This algorithm runs in time $O(n2^n)$ and can be polynomial in space. The chapter is concluded in Section 4.5. The result announced in this chapter come mainly from [45, 31, 5, 41, 50, 18].

4.1 Preliminaries

Here we recall some formal definitions that will be used throughout this chapter. All definitions in this section are taken from [10, 11, 49, 41]

Definition 4.1.1. *[10] A* lattice Λ *is said to be* orthogonal *if it has a basis B such that the rows of B are pairwise orthogonal vectors.*
In other words, a lattice Λ *is said to be* orthogonal *if it generated by a set of pairwise orthogonal vectors. We recall that a basis $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ is* orthogonal *if and only if:*

- $\langle b_i, b_i \rangle \neq 0$ *for all i and;*
- $\langle b_i, b_j \rangle = 0$ *for all $i \neq j$.*

Example 4.1.2. $\mathbb{Z}^n$ *is an* orthogonal lattice*.*
Indeed, the basis of $\mathbb{Z}^n$ is $B = (b_1, ..., b_n)$ where $b_1 = (1, 0, ..., 0)$; $b_2 = (0, 1, 0, ..., 0)$; $b_{n-1} = (0, ..., 0, 1, 0)$ and $b_n = (0, ..., 0, 1)$.
Then, for $i, j \in \{1, ..., n\}$ with $i \neq j$, b_i and b_j are orthogonal.
Thus, the rows of the generator matrix of $\mathbb{Z}^n$ are pairwise orthogonal vectors.
Therefore, $\mathbb{Z}^n$ is an orthogonal lattice*.*

Definition 4.1.3. *[11] Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice. We say that Λ' is a sublattice of Λ if $\Lambda' \subseteq \Lambda$ is a lattice as well. If Λ' is a sublattice of Λ, then $\lambda_i(\Lambda) \leq \lambda_i(\Lambda')$ for $i \leq \boldsymbol{dim}(\Lambda')$.*

Definition 4.1.4. *[11] A sublattice Λ' of $\Lambda \subseteq \mathbb{R}^n$ is said to be primitive if there exists a subspace E of $\mathbb{R}^n$ such that $\Lambda' = \Lambda \cap E$.*

Lemma 4.1.5. *[24] Let Λ be a* lattice *and $b_1, ..., b_d \in \Lambda$ be d linearly independent lattice vectors. Then $b_1, ..., b_d$ form a basis of Λ if and only if $\mathcal{P}(b_1, ..., b_d) \cap \Lambda = \{0\}$.*

In the rest of this work, we will use *full-rank lattice*.

Definition 4.1.6. *[49] Let $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ be a basis of a* lattice Λ *of rank n. The* orthogonality defect *of the basis B is the following quantity:*

$$\delta^{\top}(B) = \frac{\prod_{i=1}^{n} \|b_i\|}{\boldsymbol{det}(B)} \tag{4.1}$$

Remark 4.1.7. *$\delta^{\top}(B) \geq 1$ and if B is* orthogonal*, then $\delta^{\top}(B) = 1$. This means that if B is* orthogonal*, then $\boldsymbol{det}(B) = \prod_{i=1}^{n} \|b_i\|$*

We recall that the minimum distance can be equivalently defined as the length of the shortest nonzero lattice vector as below:

$$\lambda(\Lambda) = \inf \{\|v\| \quad : \quad v \in \Lambda \smallsetminus \{0\}\} \tag{4.2}$$

For the case of random lattices, we have an approximation of the minimum distance called Gaussian heuristic. It is defined explicitly as below.

Definition 4.1.8. *For all lattices Λ, the Gaussian heuristic $gh(\Lambda)$ gives the expected first minimum and for a full rank lattice $\Lambda \subseteq \mathbb{R}^n$, $gh(\Lambda)$ is defined as:*

$$gh(\Lambda) = \sqrt{\frac{n}{2\pi e}}.vol(\Lambda)^{1/n}. \tag{4.3}$$

We also denote $gh(n)$ for $gh(\Lambda)$ of n-dimensional lattice Λ of volume 1: $gh(n) = \sqrt{\frac{n}{2\pi e}}$.
The Gaussian heuristic says that a shortest non zero vector in a randomly chosen lattice will satisfy $v_{shortest} \approx gh(\Lambda)$.

Lemma 4.1.9. *Let a* lattice Λ *with a basis B. If $B^{\perp}$ is its* orthogonal basis*, then $\lambda_1(\Lambda) \leq \lambda_1(\Lambda^{\perp})$. Where $\lambda_1(\Lambda)$ and $\lambda_1(\Lambda^{\perp})$ are respectively the minimum distance of the lattices Λ and $\Lambda^{\perp}$.*

Proof. We use the fact that for every *orthogonal lattice*, we have only one operation (swap) for all the vectors of the basis and we have the result. □

Hermite's Theorem:

Every *lattice* Λ of dimension n contains a non zero vector $v \in \Lambda$ satisfying: $\|v\| \leq \sqrt{n}.\,(det(\Lambda))^{\frac{1}{n}}$.

Orthogonal Basis of Integer Lattices

Although the vectors of B^* are rationals, by multiplying the basis B^* by the least common multiple (lcm) of the denominators of the coordinates, we obtain the basis $B^\perp$ (with integer coordinates) with pairwise orthogonal rows. This basis $B^\perp$ is an orthogonal basis of the lattice $\Lambda(B)$.

Example 4.1.10. *Let given the base $B = (b_1, b_2, b_3)$ with $b_1 = (1, 1, 1)$; $b_2 = (-1, 0, 2)$ and $b_3 = (3, 5, 6)$. We want to determine $B^\perp$.*
The Gram-Schmidt Orthogonalization of B is given by: $B^* = (b_1^*, b_2^*, b_3^*)$ with $b_1^* = (1, 1, 1)$; $b_2^* = (-\frac{4}{3}, -\frac{1}{3}, \frac{5}{3})$ and $b_3^* = (-\frac{3}{7}, \frac{9}{14}, -\frac{3}{14})$;
since $lcm(3, 7, 14) = 42$, we have:
$b_1^\perp = 42 \times b_1^* = (42, 42, 42)$; $b_2^\perp = 42 \times b_2^* = (-56, -14, 70)$ and
$b_3^\perp = 42 \times b_3^* = (-18, 27, -9)$. Therefore, $B^\perp = (b_1^\perp, b_2^\perp, b_3^\perp)$ is an orthogonal basis (with integer coordinates) of the lattice $\Lambda(B)$.

Lemma 4.1.11. *Let a lattice Λ with a basis B. If $B^\perp$ is its orthogonal basis, then $\lambda_1(\Lambda) \leq \lambda_1(\Lambda^\perp)$. Where $\lambda_1(\Lambda)$ and $\lambda_1(\Lambda^\perp)$ are respectively the minimum distance of the lattices Λ and $\Lambda^\perp$.*

Proof. We use the fact that for every orthogonal lattice, we have only one operation (swap) for all the vectors of the basis and we have the result. □

In the next subsection, we proceed to lattice reduction assuming that an orthogonal basis is always given.

4.2 Orthogonal Reduced Basis of Integer Lattices

Given an orthogonal basis $B^\perp$ of an integer lattice $\Lambda \subseteq \mathbb{Z}^n$, Algorithm 5 returns a reduced basis $B^{\perp_1}$ of $B^\perp$, i.e a basis with vectors shorter than those of $B^\perp$. We start by calculating the gcd of the components of each vectors of $B^\perp$. After that, we divide all these vectors by this gcd. Finally, we perform permutations between these vectors in order to achieve the successive minima. The following algorithm illustrates this description.

Algorithm 5 Reduced($B^{\perp}$)

Require: The orthogonal basis $B^{\perp}$ of a lattice Λ.
Ensure: A reduced basis $B^{\perp_1}$ of the basis $B^{\perp}$.

for i from 1 to d **do**
  $b_i^{\perp_1} \leftarrow \dfrac{b_i^{\perp}}{\gcd(a_i)}$ (where $a_i's$ are the components of the vector $b_i^{\perp}$);
end for
end for
for j from d to 1 **do**
  if $\|b_j^{\perp_1}\| < \|b_{j-1}^{\perp_1}\|$ **then**
    $swaps(b_j^{\perp_1}, b_{j-1}^{\perp_1})$;
  end if
  end if
end for
end for
return $B^{\perp_1}$

Example 4.2.1. *Let be given the basis* $B = \begin{pmatrix} 1 & -1 & 3 \\ 1 & 0 & 5 \\ 1 & 2 & 6 \end{pmatrix}$ *with* $b_1 = (1,1,1)$*;* $b_2 = (-1,0,2)$ *and* $b_3 = (3,5,6)$.

The Gram-Schmidt Orthogonalization of B is given by: $B^* = \begin{pmatrix} 1 & -\frac{4}{3} & -\frac{3}{7} \\ 1 & -\frac{1}{3} & \frac{9}{14} \\ 1 & \frac{5}{3} & -\frac{3}{14} \end{pmatrix}$

with $b_1^* = (1,1,1)$*;*
$b_2^* = (-\frac{4}{3}, -\frac{1}{3}, \frac{5}{3})$ *and* $b_3^* = (-\frac{3}{7}, \frac{9}{14}, -\frac{3}{14})$*;*
since $lcm(3,7,14) = 42$*, we have:* $B^{\perp} = \begin{pmatrix} 42 & -56 & -18 \\ 42 & -14 & 27 \\ 42 & 70 & -9 \end{pmatrix}$ *with* $b_1^{\perp_1} = \frac{1}{42} \times (42,42,42) = (1,1,1)$*;* $b_2^{\perp_1} = \frac{1}{14} \times (-56,-14,70) = (-4,-1,5)$ *and*
$b_3^{\perp_1} = \frac{1}{9} \times (-18,27,-9) = (-2,3,-1)$*;*
therefore, since $\|b_3^{\perp_1}\| < \|b_2^{\perp_1}\|$ *then,*
$b_2^{\perp_1} = b_3^{\perp_1} = (-2,3,-1)$*; and* $b_3^{\perp_1} = b_2^{\perp_1} = (-4,-1,5)$*;*
since $\|b_1^{\perp_1}\| \leq \|b_2^{\perp_1}\|$*, the vectors* $b_1^{\perp_1}$ *and* $b_2^{\perp_1}$ *remains the same and we have the*

following reduced basis: $B^{\perp_1} = \begin{pmatrix} 1 & -2 & -4 \\ 1 & 3 & -1 \\ 1 & -1 & 5 \end{pmatrix}$

We recall that, the goal of lattice basis reduction is to find a basis with short vectors and orthogonal to each other. We also know that the Gram-Schmidt process does not preserve the structure of integer lattice. It would be interesting to focus on the LLL-reduction which used Gram-Schmidt process and returns integer vectors. The most usual notion of reduction is probably the LLL-reduction. The LLL- reduction is one of the most commonly used. Let $\frac{1}{4} < \delta < 1$, let $B = (b_1, ..., b_n) \in \mathbb{Z}^{n\times n}$ be a basis of a lattice. We say that B is size-reduced if all Gram-Schmidt coefficients satisfy $\mid \mu_{ij} \mid \leq \frac{1}{2}$. We say that B satisfies the Lovàsz conditions if for all $i \in \{1, ..., n\}$ we have $\delta \|b_i^*\|^2 \leq \|b_{i+1}^*\|^2 + \mu_{i+1,i}\|b_i^*\|^2$.
A basis B satisfying both the size-reduced and the Lovàsz conditions is said to be LLL-reduced. The LLL algorithm is given in [49] and it is shown that the number of LLL swaps is $O(n^2 \lg\|B\|)$. The LLL-reduction implies that the norms of the Gram-Schmidt-Orthogonalization vectors never drop too fast. Indeed the vectors are not far from being orthogonal. The most famous problem of lattice theory is the shortest vector problem (SVP), and the LLL-reduction gives a solution of this problem. We can thus deduce the *Hadamard's inequality* which is stated as below.

Hadamard's inequality [49]

Let $b_1, ..., b_n$ be vectors in $\mathbb{R}^n$ and let B be a corresponding $n \times n$ real matrix with the columns $b_1, ..., b_n$. Then *Hadamard's inequality* asserts that: $\mid \texttt{det}(B) \mid \leq \prod_{i=1}^{n} \| b_i \|$.
The most famous problem of *lattice theory* is the *Shortest Vector Problem* (SVP), and the *Closest Vector Problem* (CVP) is its non-homogeneous variant.

For random lattices, one uses the Gaussian heuristic and Gauss reduction to obtain the list of short vectors of the lattice.

Definition 4.2.2. *[41] For two given vectors $u, v \in \Lambda$, if* $\max(\|u\|, \|v\|) \leq \min(\|u - v\|, \|u + v\|)$*, then u and v are called Gauss-reduced.*

Let L be a list of N vectors from a lattice $\Lambda(B)$. If for any two different vectors v_i, v_j $(i, j = 1, ..., N \quad i \neq j)$ in L, v_i and v_j are Gauss-reduced, then the list L is called pairwise-reduced.

When solving the shortest vector problem, $gh(\Lambda)$ is usually regarded as the expected norm of the shortest vector. In the following, we will present the notion of *OrthogonaInteger* sieve which is the exact method in practice to solve the shortest vector problem in orthogonal integer lattices $\Lambda(B) \subset \mathbb{Z}^n$ where n is the dimension of lattice Λ.

In the following, we will present the notion of *sieve* which is the fastest method in practice to solve the *shortest vector problem* in *random lattice*.

4.3 Gauss Sieve algorithm

In this section, we describe the Gauss Sieve algorithm [41] and use it to propose an Orthogonal sieve algorithm for orthogonal lattices and particularly the case of integer lattice $\mathbb{Z}^n$.
It is known that all *sieving algorithm* start by *sampling* lots of lattices vectors into a list L and by shorting it.

4.3.1 List sieve algorithm

The List Sieve algorithm works by iteratively building a list L of lattice points. At every iteration, the algorithm attempts to add a new point to the list. Lattice points already in the list are never modified or removed. The goal of the algorithm is to produce shorter and shorter lattice vectors, until two lattice vectors within distance μ from each other are found, and a lattice vector achieving the target norm can be computed as the difference between these two vectors. At every iteration, a new lattice point is generated by

first picking a (somehow random, in a sense to be specified) lattice point v, and reducing the length of v as much as possible by repeatedly subtracting from it the lattice vectors already in the list L when appropriate. Finally, once the length of v cannot be further reduced, the vector v is included in the list. The main idea behind our algorithm design and analysis is that reducing v with the vector list L ensures that no two points in the list are close to each other. Since v is close to a list vector $u \in L$, then u is subtracted from v before v is considered for inclusion in the list, this immediately gives upper bounds on the space complexity of the algorithm. Moreover, if at every iteration we were to add a new lattice point to the list, we could immediately bound the running time of the algorithm as roughly quadratic in the list size, because the size of L would also be an upper bound on the number of iterations, and each iteration takes time proportional to the list size $|L|$. The problem is that some iterations might give collisions, lattice vectors v that already belong to the list. These iterations leave the list L unchanged, and as a result they just waste time. So the main hurdle in the time complexity analysis is bounding the probability of getting collisions. This is done using the same method as in the original sieve algorithm [5] instead of directly working with a lattice point v, we use a perturbed version of it $p = v + e$, where e is a small random error vector of length $\|e\| \geq \zeta\mu$ for an appropriate value of $\zeta > 0,5$. As before the length of p is reduced using list points, but instead of adding p to the list we add the corresponding lattice vector $v = p + e$. We will see that some points $p = v_1 + e_1 = v_2 + e_2$ correspond to two different lattice points v_1, v_2 at distance precisely $\|v_1 - v_2\| = \lambda_1(B)$ from each other. For example, if s is the shortest nonzero vector in the lattice, then setting $p = -e_1 = e_2 = s/2$ gives such a pair of points $v_1 = 0$; $v_2 = s$. The distance between two points in L is greater than μ or else the algorithm terminates and as a result at most one of the possible lattice vectors v_1; v_2 is in the list. This property can be used to get an upper bound on the probability of getting a collision. Unfortunately the introduction of perturbations comes at a cost. As we have discussed above, sieving produces points that are far from L and as a result we can prove a lower bound on the angles between points of similar norm. Indeed after sieving with L the point p will be far from any point in L. However the point that is actually added to the list is $v = p - e$ which can be closer to L than p by as much as $\|e\| \geq \zeta\mu$. That makes the resulting bounds on the angles worse. This worsening gets more and more significant as the norm of the points gets smaller. Fortunately we can also bound the distance between points in L by μ, which gives a good lower bound on the angles between shorter points. The space complexity of the algorithm is determined by combining these two bounds to obtain a global bound on the angle between any two points of similar norm, for any possible norm.

Sampling: The pair $(p; e)$ is chosen picking e uniformly at random within a ball of radius μ, and setting $p = emodB$. This ensures that, by construction, the ball $B(p; \zeta\mu)$ contains at least one lattice point $v = p - e$. Moreover, the conditional distribution of v (given p) is uniform over all lattice points in this ball. Notice also that for any $\zeta > 0,5$, the probability that $B(p; \zeta\mu)$ contains more than one lattice point is strictly positive: if s is a lattice vector of length $\lambda_1(B)$, then the intersection of $B(0; \zeta\mu)$ and $B(s; \zeta\mu)$ is not empty, and if e falls within this intersection, then both v and $v + s$ are within distance $\zeta\mu$ from p.

List reduction: The vector p is reduced by subtracting (if appropriate) lattice vectors in L from it. The vectors from L can be subtracted in any order. Our analysis applies independently from the strategy used to choose vectors from

L. For each $v \in L$, we subtract v from p only if $\|p - v\| < \|p\|$. Notice that reducing p with respect to v may make p no longer reduced with respect to some other $v' \in L$. So, all list vectors are repeatedly considered until the length of p can no longer be reduced. Since the length of p decreases each time it gets modified, and p belongs to a discrete set $\Lambda(B) - e$, this process necessarily terminates after a finite number of operations. In order to ensure fast termination, as in the LLL algorithm, we introduce a slackness parameter $\gamma < 1$, and subtract v from p only if this reduces the length of p by at least a factor γ. As a result, the running time of each invocation of the list reduction operation is bounded by the list size $|L|$ times the logarithm (to the base $1/\gamma$) of the length of p. For simplicity, we take $\gamma(n) = 1 - 1/n$, so that the number of iterations is bounded by a polynomial $\log(n\|B\|)/\log(1 - 1/n)^{-1} = n^[o(1)]$. The algorithm above illustrate the above description.

4.3.2 Gauss Sieve algorithm

The Gauss Sieve algorithm allows to build a list of shorter and shorter lattice vectors. And then, when a new vector v is added to the list, not only we reduce the length of v using the list vectors, but we also attempt to reduce the length of the vectors already in the list using v. This means that, if $\min(\|v \pm u\|) < \max(\|v\|, \|u\|)$, then we replace the longer of v, u with the shorter of $v \pm u$. As a result, the list of the Gauss Sieve algorithm L always consists of vectors that are pairwise reduced, it means that, they satisfy the condition $\min(\|v \pm u\|) \geq \max(\|v\|, \|u\|)$. The Gauss Sieve algorithm uses a stack data structure S to temporarily remove vectors from the list L. When a new point v is reduced with L, the algorithm checks if any point in L can be reduced with v. All such points are temporarily removed from L, and inserted in S for further reduction. The Gauss Sieve algorithm reduces the points in S with the current list before inserting them in L. When the stack S is empty, all list points are pairwise reduced, and the Gauss Sieve can sample a new lattice point v for insertion in the list L. Since (u, v) is a Gauss reduced basis, the angle between the vectors u and v is at least $\frac{\pi}{3}$. Thus the maximum size of the list can be immediately bound by the *kissing number* τ_n.
In the following, we will present the Gauss Sieve pseudo-code [41].
We will beforehand give two definitions which will allow a better understanding of this algorithm.

Definition 4.3.1. *For vectors $u, v \in \Lambda$, if* $\max(\|u\|, \|v\|) \leq \min(\|u - v\|, \|u + v\|)$*, then u, v are called Gauss-reduced.*

Definition 4.3.2. *Let list L be a set of N vectors from* lattice $\Lambda(B)$*, if for*

Algorithm 6 The List Sieve algorithm(B)

Require: Basis B and parameter μ.
Ensure: The list L

function <u>ListSieve</u>(B, μ)
$L \leftarrow \{0\}$, $\delta \leftarrow 1 - \frac{1}{n}$;
$K \leftarrow 2^{cn}$, $\zeta \leftarrow 0.685$;
for $i = 0$ **to** K **do**
  $(p_i, e_i) \leftarrow Sample(B, \zeta\mu)$;
  $v_i \leftarrow ListReduced(p_i, L, \gamma)$;
  if $vi \notin L$ **then**
    if $\exists v_j \in L$: $\|v_i - v_j\| \geq \mu$ **then**
      return $v_i - v_j$;
    end if
    $L \leftarrow L \cup \{v_i\}$;
  end if
end for
  return L
end function

function <u>Sample</u>(B, d)
$e \leftarrow \mathcal{B}_n(d)$ (random vector e);
$p \leftarrow e \mod B$;
return (p, e);
end function

function <u>ListReduce</u>(p, L, γ)
while$\exists v_i \in L$: $\|p - v_i\| \leq \gamma\|p\|$
  $p \leftarrow p - v_i$;
end while
  return p
end function

any two different vectors v_i, v_j $(i, j = 1, ..., N \quad i \neq j)$ *in* L, v_i *and* v_j *are Gauss-reduced, then list* L *is called* pairwise-reduced.

The GaussSieve algorithm is given as below:

Algorithm 7 GaussSieve(B)

Require: Basis B.
Ensure: $\|v\| : v \in B \wedge \|v\| \leq \lambda_1(B)$

function GaussSieve(B, μ)
$L \leftarrow \{0\}$, $S \leftarrow \{\}$, $K \leftarrow 0$;
while $K < c$ (number of collisions) **do**
  if S is not empty **then**
    $v_{new} \leftarrow S.pop()$;
  else
    $v_{new} \leftarrow SampleGaussian(B)$;
  end if
  $v_{new} \leftarrow GaussReduce(v_{new}, L, S)$
  if $(v_{new} = 0)$ **then**
    $K \leftarrow K + 1$;
  else
    $L \leftarrow L \cup \{v_{new}\}$;
  end if
end while
end function

function GaussReduce(p, L, S)
while $(\exists v_i \in L \quad \|v_i\| \leq \|p\| \quad \wedge \|p - v_i\| \leq \|p\|)$ **do**
  $p \leftarrow p - v_i$;
end while
while $(\exists v_i \in L \quad \|v_i\| > \|p\| \quad \wedge \|p - v_i\| \leq \|v_i\|)$ **do**
  $L \leftarrow L \setminus \{v_i\}$;
  $S.push(v_i - p)$;
end while
return p
end function

4.4 Orthogonal Sieve algorithm

In this Section, we give our main result consisting of a sieve algorithm for integer lattices. We will first define some important notions that we will use. We will denote L, a list to be constructed, containing all vectors of orthogonal integer lattices $\Lambda(B) \subseteq \mathbb{Z}^n$ such that their norm equals to the minimal distance.

Along the way, we denote H a list used to build the list L. It is the set of all vectors obtained by performing permutations of the coordinates of the vectors u and $-u$ (where u is the first short vector obtained with LLL-reduction). We will say that there is a collision if there is a repetition of the vectors in the list H.

We recall that a lattice Λ is said to be orthogonal if it is generated by set of pairwise orthogonal vectors. If two vectors are orthogonal, then the angle between them is equal to $\frac{\pi}{2}$. The number of vectors in canonical basis of the integer lattice $\mathbb{Z}^n$ is n. Considering the structure of an orthogonal basis in dimension 2, the angle between the two vectors u and v is $\frac{\pi}{2}$. We know that an orthogonal basis has generally large integer coordinates because each vector is multiplied by the lcm of the denominators of all the vectors of the basis obtained by the Gram Schmidt Orthogonalization. Since the vectors are pairwise orthogonal, we cannot use reduction coefficient's process to reduce them. Indeed, the coefficients $\mu_{i,j}$ are all zero for each i, j. Thus, for the case of orthogonal lattices, we will only have the permutations process to carry out the successive minima corresponding to this basis. Since the first minima of $LLL-$reduced basis is less than the first minima of an integer orthogonal reduced basis that we have denoted by $B^{\perp_1}$, we will used the LLL-reduced basis to find the list of shortest vectors in the general case of orthogonal integer lattice $\Lambda \subseteq \mathbb{Z}^n$. Therefore, we will initialize the empty list L, and the number of collisions by $C = 0$. After that, we use LLL-reduced basis to obtain a short vector of this lattice. Because the opposite of this short vector is also a short vector, we can use symmetries of different axes to see that all their permutations are also in the lattice, including shortest vectors. The algorithm that we are going to propose in this work will output at least n and at most 2^n shortest vectors by using the first vector obtain from the LLL-reduced basis. Thus, for the case of orthogonal lattice $\mathbb{Z}^n$, we know that $B^{\perp}_{\mathbb{Z}^n} = \{e_1, ..., e_n\}$ where $e_1 = (1, 0, ..., 0)$; $e_2 = (0, 1, ..., 0)$; . . . ; $e_n = (0, 0, ..., 0, 1)$. Thus this algorithm returns the list $L = \{-e_n, ..., -e_1, e_1, ..., e_n\}$; which gives exactly the $2n$ shortest vectors of the lattice $\mathbb{Z}^n$.

Therefore, in this case of integer lattice $\mathbb{Z}^n$, we can obtain the list of all *shortest vectors* by the following simple enumeration algorithm:

Algorithm 8 OrthogonalSieve($\mathbb{Z}^n$)

Require: The dimension n.

Ensure: A list L of *shortest* vectors.

1: $B^{\perp} \leftarrow (e_1, ..., e_n)$ (orthogonal basis of $\mathbb{Z}^n$);

2: $L \leftarrow (-e_n, -e_{n-1}, ..., -e_1, e_1, ..., e_{n-1}, e_n)$;

3: **return** L

Remark 4.4.1. *Indeed, in this case, our* orthogonal basis *is the canonical basis and it does not give all the* shortest vectors *because the opposites of these vectors are also the* shortest vector*. Therefore, to have all the* shortest *vectors of the list L, it must be completed with the opposites of the vectors already present in the* orthogonal basis*.*

Example 4.4.2. *For $n = 4$, the orthogonal basis of $\mathbb{Z}^4$ is given by: $B^{\perp} = (e_1, e_2, e_3, e_4)$ where $e_1 = (1, 0, 0, 0)$; $e_2 = (0, 1, 0, 0)$; $e_3 = (0, 0, 1, 0)$ and $e_4 = (0, 0, 0, 1)$.*
Therefore the list L of shortest vectors is given by:
$L = \{-e_4, -e_3, -e_2, -e_1, e_1, e_2, e_3, e_4\}$.

Lemma 4.4.3. *Let Λ be a full rank integer lattice of dimension n. Λ has at least n and at most $N = n!.2^n$ shortest vectors.*
Particularly,
$1-$ *The integer lattice $\mathbb{Z}^n$ has exactly $2n$ shortest vectors;*
$2-$ *The root lattice of type A_n $(n \geq 1)$ has exactly $n(n + 1)$ shortest vectors;*
$3-$ *The root lattice of type D_n $(n \geq 2)$ has exactly $2n(n - 1)$ shortest vectors.*

Proof. Let Λ be a full rank integer lattice of dimension n. We know that there exists a vector $v = (v_1, ..., v_n) \in \Lambda \setminus \{0\}$ such that $\|v\| = \lambda_1(\Lambda)$. We also know that $u = -v$ is another shortest vector in Λ. Likewise, all the permutations of the coordinates of v and u are a shortest vector of the lattice. The vector v has at most $n!$ permutations and the vector u has also at most $n!$ permutations. Thus, we have at most $(n!)^2$ permutations possible for one of these vectors. Moreover, the vectors u and v have at most 2^n possibilities to combine symmetrically by the different axes. Therefore, we have at most $n!.2^n$ shortest vectors in integer lattices.
For the case of integer lattice $\mathbb{Z}^n$, we know that the n vectors of canonical basis are the shortest vectors of this lattice. Since their opposites are also the shortest vectors of $\mathbb{Z}^n$, we have exactly $2n$ shortest vectors in $\mathbb{Z}^n$.
Since the short vectors of the root lattices of type A_n $(n \geq 1)$ are the permutations of the vector $(+1, -1, 0, ..., 0)$, then we will have exactly $\dfrac{(n+1)!}{(n-1)!} =$

$n(n+1)$ short vectors in this particular lattice. Thus we will have exactly $n(n+1)$ short vectors in root lattices of type A_n.
About the root lattices of type D_n ($n \geq 2$), we also know that all the short vectors are the permutations of the vector $(\pm 1, \pm 1, 0, ..., 0)$ with the condition that the sum of all the components is even. Thus we will have three possible following cases: the permutations of the vector $(+1, -1, 0, ..., 0)$, the permutations of the vector $(+1, +1, 0, ..., 0)$ and the permutations of the vector $(-1, -1, 0, ..., 0)$.
This means that, we will have exactly $\frac{n!}{(n-2)!} + \frac{n!}{2!(n-2)!} + \frac{n!}{2!(n-2)!} = 2n(n-1)$.
Therefore, we will exactly have $2n(n-1)$ short vectors in root lattices of type D_n. □

Corollary 4.4.4. *Given a basis B of the orthogonal lattice $\mathbb{Z}^n$, we can obtain the list L of shortest vectors of this lattice in space $O(2n)$.*

Proof. Let B be a basis of the orthogonal lattice $\mathbb{Z}^n$. The canonical basis permits to obtain exactly $2n$ short vectors of this lattice. Then these vectors will be obtained in space $O(n)$. □

We are now going to propose an enumeration algorithm which will take as input a basis (not orthogonal) of the integer lattice Λ and return a list of at most 2^n shortest vectors of this lattice. Since this lattice is an integer lattice, then the *LLL* algorithm will return a shortest vector of the lattice that we call v. Even if an integer lattice is also an orthogonal lattice, it would be interesting to use a non-orthogonal basis of the lattice. Indeed, by applying the *LLL* algorithm to an orthogonal basis, we obtain the same basis. Consequently, the vectors obtained will not necessarily be the short vectors of the lattice.
Therefore, we will bring out all the possible combinations between the components of the vector v and its opposite $-v$ (this by keeping the position of each component used). The description of our algorithm is given as below.

4.4.1 Description of the Algorithm

Given an orthogonal integer lattice Λ, this algorithm takes as input the (non orthogonal) basis $B = (b_1, b_2, ..., b_n)$ of the lattice (where n is the dimension of Λ)and returns a list L of at least n and at most 2^n short vectors of the lattice Λ and the number of collision C as follows: we start by executing the *LLL* algorithm to the basis B which allows us to obtain a short vector of the lattice which we denote by u. Subsequently, we will use this vector u and its opposite

$v = -u$ to build a list L. To achieve this, we will build a $2^n \times n$ matrix K using an iterative function $Vect$ and an additional $2^{n-1} \times (n-1)$ matrix P. The 2^n rows of our constructed matrix K will be short vectors of the lattice. Now, we will consider the list H whose elements are rows of K. A final list L consisting of short vectors will then be constructed from K, making sure that an element appears only once. The number of collisions will be the number of repetitions of the vectors in the list H.
At the end of the algorithm, we will have the list L which will be made up of at least n and at most 2^n short vectors of the lattice, and the number of collisions C. The following explanations will help to better understand the algorithm.
• The function **Vect** takes as input the vectors p and q, and builds a $2^{n-1} \times (n-1)$ matrix P;
• $K[i,]$ is line number i of the matrix K;
• matrix(0, $nrow = 2^n$, $ncol = n$) is the $2^n \times n$ matrix with 0 everywhere;
• The function **LLL(B)** takes as input the basis B and returns its LLL-reduced basis.

Remark 4.4.5. *We will call the number of collisions that we will denote by C, the total number of repetitions of the vectors that we will have in the auxiliary list H which will make it possible to obtain the list L of short vectors. Thus, if the number of collisions is large, then the size of the list L is small. Indeed, the total number of vectors of the list L will be equal to $2^n - C$.*

The algorithm below illustrates the above description. For correctness, a Maple computer software implementation of the algorithm has been done.

Algorithm 9 Orthogonal integer sieve

Require: The basis B of a lattice Λ and its dimension $n \geq 2$.
Ensure: A list L of *short* vectors v with $\|v\|= \lambda_1(\Lambda(B))$ and integer C.

$L := \{\}$; $C := 0$; "We initialize an empty list L and integer C "
$G := LLL(B)$;
$u := G[,1]$; $v := -u$; " u is the $1st$ column of matrix G"
$p := (0, ..., 0)$; $q := (0, ..., 0)$ "$(n-1)$ times"
for $i = 1, \cdots, n-1$ **do**
  $p_i := u_i$; $q_i := v_i$;
end for
$P := Vect(p, q, n-1)$;
$K := matrix(0,\ nrow = 2^n,\ ncol = n)$; $l := 2^n$; $t := 2^{n-1}$;
for $i = 1, \cdots, t$ **do**
  for $j = 1, \cdots, n-1$ **do**
    $K[i, j] := P[i, j]$;
  end for
end for
for $i = t+1, \cdots, l$ **do**
  for $j = 1, \cdots, n-1$ **do**
    $K[i, j] := P[i-t, j]$;
  end for
end for
for $i = 1, \cdots, t$ **do**
  $K[i, n] := u_n$; "we update the 2^{n-1} first components of column n"
end for
for $i = t+1, \cdots, l$ **do**
  $K[i, n] := v_n$; "we update the last 2^{n-1} components of column n"
end for
end if
$H := (K[1,], ..., K[2^n,])$; $L := L \cup \{H[1]\})$;
for $i = 2, \cdots, 2^n$ **do**
  if $H[i] \notin L$ then **then**
    $L := L \cup \{H[i]\})$; "we remove all copies from the list"
    else $C := C + 1$;
  end if
end for
return (The list L of *shortest* vectors v with $\|v\|= \lambda_1(\Lambda(B))$ and C);

4.4.2 Complexity Analysis

About the complexity of our algorithm, we have:
The line 1 has 2 elementary operations. Indeed, we have only 2 assignments in this step;
line 2 is carried out in polynomial time with complexity $O(n)$ arithmetic operations. Indeed, algorithm LLL runs in $O(n)$ arithmetic operations.
The line 3 has 2 elementary operations (assignments).
Line 4 has $2(n-1)$ arithmetic operations. Indeed, in this line we have 2 affectations inside the loop for which goes from 1 to $n-1$;
from line 5 to line 7, we also have $2(n-1)$ elementary operations. Indeed, we have 2 assignments inside the loop for which goes from 1 to $n-1$;
The line 8 has $(n-1)2^{n-1}$ arithmetic operations. Indeed, we use a recursive algorithm that uses two loops "for", which one goes from 1 to 2^{n-1} and other from 1 to $n-1$;
Line 9 has 3 elementary operations (assignments);
from line 10 to line 14, we have two loops and the first goes from 1 to 2^{n-1}, and inside this one we have another loop for which goes from 1 to $n-1$. Thus, we will have $2^{n-1}(n-1)$ operations from line 10 to line 14.
In the same way, we will have $2^{n-1}(n-1)$ operations from line 15 to line 19;
from line 20 to line 22, we have 2^{n-1} because we have only one operation inside the loop for which goes from 1 to 2^{n-1}. In the same way, we will have 2^{n-1} operations from line 23 to line 26;
line 27 has 2^n+1 operations because we have 1 elementary operation (assignment) and 2^n assignments to build matrix K;
from line 29 to line 34, we have 2 operations (assignment and comparison) which will be automatically executed inside the loop for which goes from 1 to 2^n-1. Thus we will have $2\times(2^n-1)=2^{n+1}-2$ operation from line 29 to line 34.
So we will have $2^{n+1}-2+2^n+1+2^{n-1}+2^{n-1}+(n-1)2^{n-1}+(n-1)2^{n-1}+(n-1)2^{n-1}+2(n-1)+2+n+2$ arithmetic operations;
this means that we have $2^{n+1}-2+2^n+1+2^n+(n-1)2^n+(n-1)2^{n-1}+2(n-1)+n+4$ arithmetic operations;
thus, we have $2^{n+1}+2^{n+1}+(n-1)2^n+(n-1)2^{n-1}+2(n-1)+n+3$;
since $\dfrac{2^{n+2}+(n-1)2^n+(n-1)2^{n-1}+2(n-1)+n+3}{n2^n} \rightarrow \quad cte$ when $n \rightarrow +\infty$, then the complexity of algorithm is $O(n2^n)$.
Therefore, the complexity of our algorithm is $O(n2^n)$ arithmetic operations.

Example 4.4.6. *Let* $B := \begin{pmatrix} 3 & 3 & -3 \\ 1 & 3 & 1 \\ 1 & 4 & -2 \end{pmatrix}$ *be a basis of a lattice* $\Lambda(B) \subset \mathbb{Z}^3$*;*
we have, $G := LLL\,(B) = \begin{pmatrix} 0 & 0 & 3 \\ 2 & 2 & 1 \\ -1 & 3 & 1 \end{pmatrix}$*;*
thus $u = (0, 2, -1)$*,* $v = (0, -2, 1)$ *and* $n = 3$*;*
We have $n \neq 1$*, then* $p = (0, 2)$ *and* $q = (0, -2)$*;*
then P:=Vect($p = (0, 2), q = (0, -2), n = 2$*);*
thus $n = 2 \neq 0$*, this means that we have P:= Vect(*$p = (0), q = (0), n = 1$*);*
therefore, $l = 2^2 = 4$ *and* $t = 2^{2-1} = 2$*; thus*
for $i = 1, 2$ *and* $j = 1$ *we have:* $P[1, 1] = 0$ *and* $P[2, 1] = 0$
for $i = 3, 4$ *and* $j = 1$ *we have:* $P[3, 1] = 0$ *and* $P[4, 1] = 0$
Thus P *is the form* $K := \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$
now we will complete the second column as below:
for $i = 1, 2$ *and* $j = 2$ *we have:* $P[1, 2] = P[2, 2] = u_2 = 2$*;*
for $i = 3, 3$ $j = 2$ *we have:* $P[3, 2] = P[4, 2] = v_2 = -2$*;*

and then, we have $P := \begin{pmatrix} 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 0 & -2 \end{pmatrix}$

now $l = 2^3 = 8$ *and* $t = 2^2 = 4$*;*
thus for $i = 1, ..., 4$ *and* $j = 1, 2$ *we have:* $K[1, 1] = P[1, 1] = 0$*;* $K[1, 2] = P[1, 2] = 2$*;*
$K[2, 1] = P[2, 1] = 0$*;* $K[2, 2] = P[2, 2] = 2$*;* $K[3, 1] = P[3, 1] = 0$*;* $K[3, 2] = P[3, 2] = -2$*;* $K[4, 1] = P[4, 1] = 0$ *and* $K[4, 2] = P[4, 2] = -2$*;*
for $i = 5, ..., 8$ *and* $j = 1, 2$ *we also have:* $K[5, 1] = P[1, 1] = 0$*;* $K[5, 2] = P[1, 2] = 2$*;*
$K[6, 1] = P[2, 1] = 0$*;* $K[6, 2] = P[2, 2] = 2$*;* $K[7, 1] = P[3, 1] = 0$*;* $K[7, 2] = P[3, 2] = -2$*;* $K[8, 1] = P[4, 1] = 4$ *and* $K[8, 2] = P[4, 2] = -2$*;*

Thus K is the form $K := \begin{pmatrix} 0 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & -2 & 0 \\ 0 & -2 & 0 \\ 0 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & -2 & 0 \\ 0 & -2 & 0 \end{pmatrix}$

now we will complete the last column as below:
for $i = 1, ..., 4$ *and* $j = 3$ *we have* $K[1,3] = K[2,3] = K[3,3] = K[4,3] = u_3 = -1$*;*
for $i = 5, ..., 8$ *and* $j = 3$ *we have* $K[5,3] = K[6,3] = K[7,3] = K[8,3] = v_3 = 1$*;*

thus, we have $K = \begin{pmatrix} 0 & 2 & -1 \\ 0 & 2 & -1 \\ 0 & -2 & -1 \\ 0 & -2 & -1 \\ 0 & 2 & 1 \\ 0 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & -2 & 1 \end{pmatrix}$ *Thus,*

$H = \{(0,2,-1),(0,2,-1),(0,-2,-1),(0,-2,-1),(0,2,1),(0,2,1),(0,-2,1),(0,-2,1)\}$.
Therefore, $L = \{0,2,-1),(0,-2,-1),(0,2,1),(0,-2,1)\}$ *and* $C = 4$.

4.5 Concluding remarks

In this chapter, we talked about the notions of orthogonal lattices, integer lattices, gave some properties of this family of lattices. We also recalled the relationship between orthogonal and integer lattices. All this allowed us to construct an enumeration algorithm for integer lattice $\mathbb{Z}^n$ to provide a full list of its shortest vectors. This algorithm runs in space $O(n)$. We also constructed an algorithm which gives at least n and at most 2^n short vectors of a general orthogonal integer lattice $\Lambda \subset \mathbb{Z}^n$. This algorithm runs in time $O(n2^n)$ and can be polynomial in space. We have successfully implemented these algorithms in the Maple computer software 18.0. Our future work will consist in giving an algorithm which will give a list of short vector in general case of any orthogonal lattice.

Chapter Five

CONCLUSION AND FURTHER WORK

In this thesis, we have built a new family of lattice (tensor product of two root lattices of type D) for which the Closest Vector Problem is solved efficiently. Subsequently, we solved the Closest Vector Problem in the tensor product of three root lattices of type A, before generalizing this resolution for the tensor product of a finite number of root lattices of type A. We have also constructed a list of vectors with minimum norm in the orthogonal integer lattices of dimension n and in particular for the case of integer lattice $\mathbb{Z}^n$. We have adopted a natural approach, by focusing on the first two cases on the existing relations with the lattices whose properties are known. To arrive at the results, we have used various techniques, from the classical computation of complexity, to some properties of directed graphs and geometry of numbers.
From an algorithmic point of view, our contributions are the following:

1. We have given a polynomial algorithm to determine the closest vector in the tensor product of two root lattices of type D. To achieve this result, we first characterized the vectors of this new family of lattices, then we established the relationship between this lattice and the root lattices of type D. We used this characterization and the same method for the root lattice of type D to obtain this new polynomial algorithm. Our future work will consist to generalise this algorithm to solve this problem for the case of tensor product of a finite number k of root lattices of type D_n $(n \geq 2)$ which we denote by $\bigotimes_{i=1}^{k} D_i$. We will also characterize the Voronoi region vectors in root lattice $D_n \otimes D_m$ and use it to propose another algorithm to solve Closest Vector Problem in lattice $D_n \otimes D_m \otimes D_p$ $(n, m, p \geq 2)$.

2. We have given a polynomial algorithm to solve the Closest Vector Problem in the tensor product of three root lattices of type A, and we have also given an algorithm which generalizes this resolution in the tensor product of a finite number of lattices of type A. To achieve this result, we first characterized the vectors of this new family of lattices $(A_n \otimes A_m \otimes A_p)$, then we established the relationship between this lattice and the root lattices of type A. We used this characterization and the same method for the root lattice of two root lattices $A_n \otimes A_m$ to obtain the polynomial algorithm in the case of tensor product of three root lattices of type A. We used associativity and non commutativity of tensor product of root lattice to generalize the result for the case of tensor product of a finite root lattices of type A. As future work, we will improve the algorithm for solving the closest vector problem in the tensor product of two and three root lattices of type A. Indeed, a tensor product of two or three root lattices is also a sub lattice of a root lattice with some particular properties.

3. We have constructed an enumeration algorithm for integer lattice $\mathbb{Z}^n$ to provide a full list of its shortest vectors. We have also constructed an algorithm which gives at least n and at most 2^n short vectors of a general case of orthogonal integer lattice $\Lambda \subset \mathbb{Z}^n$. We used the LLL-reduction algorithm. Indeed, from the first column vector of the LLL-reduced basis of the considered basis, we built the list by permuting the components of this column vector. Our future work will consist in giving an algorithm which will give a list of short vector in general case of any orthogonal lattice.

All the previous algorithms are implemented in Maple software 18.0 to get all the results presented in Chapters 3 and 4.

Conferences attended during this research

During this research we have participated to the conferences and workshops listed below:

1. Conference GIRAGA, International Conference of Mathematics, University of Yaounde 1, Yaounde, Cameroon 13-18 December, 2021.

2. ASCRYPTO and LATINCRYPT, School of Engineering, Science and Technology, University of Del Rosario, Bogota, Colombia 04-08 October, 2021.

3. CRAG 10, Algebra, Arithmetic and Combinational Geometry, Algebraic number and with Applications in Cryptography, University of Dschang, Dschang, Cameroon 19-30 July 2021.

4. Conference CIMY, International Conference of Mathematics, University of Yaounde 1, Yaounde, Cameroon 09-14 September, 2019.

5. 21th Workshop on Algebra and Logic, *Codes, Cryptography, formal concept analysis*, University of Yaounde I, Cameroon, September 28-01, 2019.

6. CIMPA School, Algebraic Geometry, Number Theory and Applications in Cryptography and Robot kinematics , AIMS Cameroon, Limbe, Cameroon 2-12 July 2019.

7. African Mathematical School (AMS) and 8th International Conference on Cryptography, Algebra and Geometry (CRAG-8), University of Yaounde I, Cameroon, July 16-28, 2018.

Bibliography

[1] Divesh Aggarwal, Daniel Dadush, and Noah Stephens-Davidowitz. Solving the closest vector problem in 2^n time - the discrete gaussian strikes again! In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 563–582. IEEE Computer Society, 2015.

[2] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows - theory, algorithms and applications.* Prentice Hall, 1993.

[3] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108. ACM, 1996.

[4] Miklós Ajtai. The shortest vector problem in L_2 is *NP*-hard for randomized reductions (extended abstract). In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 10–19. ACM, 1998.

[5] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 601–610. ACM, 2001.

[6] Johannes Blömer and Alexander May. New partial key exposure attacks on RSA. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 27–43. Springer, 2003.

[7] Johannes Blömer and Jean-Pierre Seifert. On the complexity of computing short linearly independent vectors and short bases in a lattice. In Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton, editors, *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 711–720. ACM, 1999.

[8] Dan Boneh and Glenn Durfee. Cryptanalysis of RSA with private key d less than $n^{0.292}$. *IEEE Trans. Inf. Theory*, 46(4):1339–1349, 2000.

[9] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. Flush, gauss, and reload - A cache attack on the BLISS lattice-based signature scheme. *IACR Cryptol. ePrint Arch.*, page 300, 2016.

[10] Karthekeyan Chandrasekaran, Venkata Gandikota, and Elena Grigorescu. Deciding orthogonality in construction-a lattices. *SIAM J. Discret. Math.*, 31(2):1244–1262, 2017.

[11] Jingwei Chen, Damien Stehlé, and Gilles Villard. Computing an lll-reduced basis of the orthogonal latice. In Manuel Kauers, Alexey Ovchinnikov, and Éric Schost, editors, *Proceedings of the 2018 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2018, New York, NY, USA, July 16-19, 2018*, pages 127–133. ACM, 2018.

[12] John H. Conway and Neil J. A. Sloane. Fast quantizing and decoding and algorithms for lattice quantizers and codes. *IEEE Trans. Inf. Theory*, 28(2):227–231, 1982.

[13] John H. Conway and Neil J. A. Sloane. *Sphere Packings, Lattices and Groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1988.

[14] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition.* MIT Press, 2009.

[15] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition.* MIT Press, 2009.

[16] Thomas Debris-Alazard, Léo Ducas, and Wessel P. J. van Woerden. An algorithmic reduction theory for binary codes: LLL and more. *IEEE Trans. Inf. Theory*, 68(5):3426–3444, 2022.

[17] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654, 1976.

[18] Léo Ducas. Shortest vector from lattice sieving: A few dimensions for free. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 125–145. Springer, 2018.

[19] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. *IACR Cryptol. ePrint Arch.*, page 383, 2013.

[20] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.

[21] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. E fficient identity-based encryption over NTRU lattices. *IACR Cryptol. ePrint Arch.*, page 794, 2014.

[22] Léo Ducas and Wessel P. J. van Woerden. The closest vector problem in tensored root lattices of type A and in their duals. *Des. Codes Cryptogr.*, 86(1):137–150, 2018.

[23] Allen Gersho. Asymptotically optimal block quantization. *IEEE Trans. Inf. Theory*, 25(4):373–380, 1979.

[24] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer, 1997.

[25] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Electron. Colloquium Comput. Complex.*, (2), 1999.

[26] Laurent Grémy, Aurore Guillevic, François Morain, and Emmanuel Thomé. Computing discrete logarithms in $\mathbb{F}_{p^6}$. In Carlisle Adams and Jan Camenisch, editors, *Selected Areas in Cryptography - SAC 2017 - 24th*

International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers, volume 10719 of *Lecture Notes in Computer Science*, pages 85–105. Springer, 2017.

[27] Aurore Guillevic, Simon Masson, and Emmanuel Thomé. Cocks-pinch curves of embedding degrees five to eight and optimal ate pairing computation. *Des. Codes Cryptogr.*, 88(6):1047–1081, 2020.

[28] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 530–547. Springer, 2012.

[29] Nick Howgrave-Graham and Nigel P. Smart. Lattice attacks on digital signature schemes. *Des. Codes Cryptogr.*, 23(3):283–290, 2001.

[30] Hendrik W. Lenstra Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983.

[31] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987.

[32] J. C. Lagarias. Knapsack public key cryptosystems and diophantine approximation. In David Chaum, editor, *Advances in Cryptology, Proceedings of CRYPTO '83, Santa Barbara, California, USA, August 21-24, 1983*, pages 3–23. Plenum Press, New York, 1983.

[33] J. C. Lagarias. The computational complexity of simultaneous diophantine approximation problems. *SIAM J. Comput.*, 14(1):196–209, 1985.

[34] J. C. Lagarias and Andrew M. Odlyzko. Solving low-density subset sum problems. In *24th Annual Symposium on Foundations of Computer Science, Tucson, Arizona, USA, 7-9 November 1983*, pages 1–10. IEEE Computer Society, 1983.

[35] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, 2009.

[36] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, 2010.

[37] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-lwe cryptography. *IACR Cryptol. ePrint Arch.*, 2013:293, 2013.

[38] Alexander May. Cryptanalysis of unbalanced RSA with small crt-exponent. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 242–256. Springer, 2002.

[39] Alexander May. Computing the RSA secret key is deterministic polynomial time equivalent to factoring. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 213–219. Springer, 2004.

[40] Robby G. McKilliam, Alex J. Grant, and I. Vaughan L. Clarkson. Finding a closest point in a lattice of voronoi's first kind. *CoRR*, abs/1405.7014, 2014.

[41] Daniele Micciancio and Panagiotis Voulgaris. Faster exponential time algorithms for the shortest vector problem. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1468–1480. SIAM, 2010.

[42] Phong Q. Nguyen. Can we trust cryptographic software? cryptographic flaws in GNU privacy guard v1.2.3. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 555–570. Springer, 2004.

[43] Kaisa Nyberg and Rainer A. Rueppel. Message recovery for signature schemes based on the discrete logarithm problem. *Des. Codes Cryptogr.*, 7(1-2):61–81, 1996.

[44] Peter Pessl, Leon Groot Bruinderink, and Yuval Yarom. To BLISS-B or not to be: Attacking strongswan's implementation of post-quantum signatures. In Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1843–1855. ACM, 2017.

[45] Michael Pohst. On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. *SIGSAM Bull.*, 15(1):37–44, 1981.

[46] Michael Pohst. A modification of the LLL reduction algorithm. *J. Symb. Comput.*, 4(1):123–127, 1987.

[47] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[48] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[49] Damien Stehlé. *Algorithmique de la réduction de réseaux et application à la recherche de pires cas pour l'arrondi defonctions mathématiques.* PhD thesis, Henri Poincaré University, Nancy, France, 2005.

[50] Panagiotis Voulgaris. *Algorithms for the closest and shortest vector problems on general lattices.* PhD thesis, University of California, San Diego, USA, 2011.

Published papers

During this research, our contributions have been published to the journals as listed below :

1. Arnaud Girès Fobasso Tchinda, Emmanuel Fouotsa, Celestin Nkuimi Jugnia, Sieve Algorithms for Some Orthogonal Integer Lattices, *Discrete Mathematics, Algorithms and Applications*, (2022) https://doi.org/10.1142/S179383022501518

2. Arnaud Girès Fobasso Tchinda, Emmanuel Fouotsa and Celestin Nkuimi Jugnia, A Polynomial Algorithm for Solving the Closest Vector Problem in Tensored Root Lattices of Type D, *SN Computer Science, Springer* (2022) https://doi.org/10.1007s42979-022-01440-2.

3. Arnaud Girès Fobasso Tchinda, Emmanuel Fouotsa, Celestin Nkuimi Jugnia, Generalization of Closest Vector Problem in Tensored Root Lattices of Type A. Under review at *Indian Journal of Pure and Applied Mathematics, Springer*.

Articles

SN Computer Science (2023) 4:19
https://doi.org/10.1007/s42979-022-01440-2

ORIGINAL RESEARCH

A Polynomial Algorithm for Solving the Closest Vector Problem in Tensored Root Lattices of Type D

Arnaud Girès Fobasso Tchinda[1] · Emmanuel Fouotsa[2] · Celestin Nkuimi Jugnia[1]

Received: 4 December 2020 / Accepted: 3 October 2022
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2022

Abstract

The purpose of this work is to propose an efficient algorithm to solve the closest vector problem (CVP) in the tensor product of two root lattices of type D_n $(n \geq 2)$. In 2018, Léo Ducas and Wessel van Woerden proposed a polynomial algorithm allowing to solve this problem in the tensor product of two root lattices of type A_n $(n \geq 1)$. In our present case, we show that the root lattice D_{nm} is a full-rank sub-lattice of the tensor product $D_n \otimes D_m$ $(n, m \geq 2)$ of the root lattices D_n and D_m, enabling us to derive a polynomial algorithm for solving the CVP in D_n $(n \geq 2)$. The proposed algorithm performs at most $O(n + m)$ arithmetic operations.

Keywords Lattice-based cryptography · Tensored root lattices · Closest vector problem

Mathematics Subject Classification 11H71 · 11H06 · 94B35

Introduction

A lattice is a discrete additive subgroup of $\mathbb{R}^n$. A central problem in the theory of lattice is the Closest Vector Problem (CVP). However, the seeking for the closest vector in a lattice is a difficult mathematical problem [10], used in cryptography to build robust and secured cryptosystems resistant to quantum computers [5, 14]. Although CVP is an NP-hard problem for general lattices, it is interesting to design lattices for which CVP can be solved efficiently, while at the same time optimizing other lattices properties like the packing density. Special lattices are, for example, the root lattices A_n $(n \geq 1)$, D_n $(n \geq 2)$, E_n $(n = 6, 7, 8)$, their duals, and the Leech lattice [3, 4, 6–8]. These lattices can be used as the basis for efficient block quantizers for uniformly distributed inputs and to construct code for a band-limited channel with Gaussian noise [4, 9]. Indeed, recent attempts to create lattice-based cryptographic schemes are promising and are mostly based on removing some error to a lattice vector using a CVP algorithm [11, 12]. Léo Ducas and Wessel van Woerden proposed a polynomial algorithm for solving CVP for the case of the lattice $A_n \otimes A_m$ $(n, m \geq 1)$ to give a generalization of resolution of CVP on some case of cyclotomic integer lattices $\mathbb{Z}[]$ (with $\alpha = p.q$, where p and q are prime) and their duals [7]. We build in the same order a new family of lattices that we called tensored root lattice of type D_n $(n \geq 2)$ which CVP is solved in polynomial time. Even though there are some families of lattices for which CVP is solved with a polynomial time algorithm, it would be important to remember that lattices have many applications in cryptography. Indeed, in December 2016, the National Institute of Standards and Technology (NIST) announced a competition to select new quantum resistant public key encryption algorithms that would eventually supersede the classical RSA and other public key cryptography algorithms that may be vulnerable to future quantum computer. For the past 5 years, after the third round of this competition, lattice was selected to continue.

In this work, we propose a polynomial time algorithm to solve CVP in the tensor product $D_n \otimes D_m$ $(n, m \geq 2)$, where D_n and D_m are two root lattices of type D_n $(n \geq 2)$.

✉ Arnaud Girès Fobasso Tchinda
fobass1989@gmail.com

Emmanuel Fouotsa
emmanuelfouotsa@yahoo.fr

Celestin Nkuimi Jugnia
nkuimi@yahoo.co.nk

1 Department of Mathematics, Faculty of Science, University of Yaounde I, Yaoundé, Cameroon

2 Department of Mathematics, Higher Teacher Training College, University of Bamenda, Bamenda, Cameroon

Published online: 19 October 2022

The paper is organized as follows. In the section "General Preliminaries", we introduce and recall some definitions and preliminaries that will be useful in the paper. In the section "The Root Lattice D_n", we present the root lattices of type D_n $(n \geq 2)$ as well as an efficient algorithm to solve the closest vector problem. Our main result is presented in the section "Our Result: The Closest Vector Problem in $D_n \otimes D_m$" where we present a polynomial time algorithm which solves the closest vector problem in $D_n \otimes D_m$. The work is concluded in the section "Conclusion".

General Preliminaries

We recall here the definitions and properties that will be used throughout this work.

Throughout this paper, for any positive integer d, we use the Euclidean product on $\mathbb{R}^d$ that is defined by: $\langle \mathbf{x}, \mathbf{y} \rangle := x_1y_1 + x_2y_2 + \cdots + x_dy_d$ for $\mathbf{x} := (x_1, x_2, \ldots, x_d)$ and $\mathbf{y} := (y_1, y_2, \ldots, y_d)$ in $\mathbb{R}^d$. The Euclidean norm on $\mathbb{R}^d$ is defined as follows: $||\mathbf{x}|| := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$.

We denote by $\mathbb{B}(x, R)$ the closed Euclidean $n-$dimensional ball of radius R centered at x, such that: $\mathbb{B}(x, R) = \{y \in \mathbb{R} : \|x - y\| < R\}$. If no center is specify, then the center is zero $\mathbb{B}(R) = \mathbb{B}(0, R)$. More details about these preliminaries can be found in [1–3, 7, 13].

Basic Properties of Lattices

Definition 1 A *lattice* is a discrete additive subgroup of $\mathbb{R}^d$, for any positive integer d. We deal exclusively with any *lattice* Λ of rank r, which is generated as the set of all integer linear combinations of r linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r$ (for which there are a basis of this *lattice* that we denoted by B) in $\mathbb{R}^d$ as follows:

$$\Lambda = \left\{ \sum_{i=1}^{r} z_i \mathbf{b}_i \ : \ (z_1, z_2, \ldots, z_r) \in \mathbb{Z}^r \right\}. \tag{1}$$

Definition 2 The *rank* of a *lattice* Λ is defined as the number of linearly independent vector in any basis for that lattice. Indeed, in the Definition 1, the *rank* of the *lattice* Λ is r and its *dimension* is d. The lattice Λ is said to be of *full-rank* if $r = d$.

Definition 3 Let Λ be a *lattice* and B its basis, we defined the *fundamental parallelepiped* of Λ, denoted $\mathbb{P}(B)$ as below

$$\mathbb{P}(B) = \left\{ Bx \mid x \in \mathbb{R}^d , \ \forall i \ : \ 0 \leq x_i < 1 \right\}. \tag{2}$$

For any lattice basis B and point x, there exists a unique vector $y \in \mathbb{P}(B)$, such that $y - x \in \Lambda(B)$.

Definition 4 The *determinant* of a lattice Λ denoted $\mathtt{det}(\Lambda)$ is defined as being the *volume* of fundamental parallelepiped $\mathbb{P}(B)$ given by

$$\mathtt{det}(\Lambda) = \mathtt{vol}(\mathbb{P}(B)) = \sqrt{\mathtt{det}(B^T B)},$$

where B^T is the transpose of the matrix B. If the lattice Λ is of full rank, then B is a square matrix, and consequently, we have

$$\mathtt{det}(\Lambda) =\mid \mathtt{det}(B) \mid .$$

Specifically, in a *lattice* Λ, any non-zero vector v has a strictly positive length. However, the problem which arises is that of knowing if this length is relatively small compared to the other vectors of the *lattice*. This leads us to introduce the notion of *minimum distance* in a lattice and more generally the i'th *successive minima* of a *lattice* as below.

Successive Minima

Let $\Lambda(B)$ be a lattice of dimension n. Let $i \leq n$, the i'th minimum of lattice, denoted $\lambda_i(\Lambda)$, is defined by

$$\lambda_i(\Lambda) = \min \{R, \dim((\Lambda \cap \mathbb{B}(R))) = i\}. \tag{3}$$

The successive minima of a given lattice are all reached. There exist vectors of the lattice of norms equal to the successive minima, and can be so in particular by linearly independent vectors. The *minimum distance* of a lattice Λ w.r.t Euclidean norm, denoted $\|\Lambda\|$, is the length of a shortest lattice non-zero vector, i.e., $\|\Lambda\| := \min\limits_{\mathbf{0} \neq \mathbf{x} \in \Lambda} \|\mathbf{x}\|$.

Another lattice Λ^* in $\mathbb{R}^d$ of the same rank r, such that $\Lambda^* \subset \Lambda$ is called a *full-rank sub-lattice* of Λ. A *generator matrix* of Λ^* is a matrix whose rows form a base of Λ.

Definition 5 Let $\Lambda \subseteq \mathbb{R}^d$ be a lattice. We say that Λ' is a sub-lattice of Λ if $\Lambda' \subseteq \Lambda$ is a lattice, as well. If Λ' is a sub-lattice of Λ, then $\lambda_i(\Lambda) \leq \lambda_i(\Lambda')$ for $i \leq \mathtt{dim}(\Lambda')$.

Definition 6 The *span* of a lattice Λ is the linear space spanned by its vectors

$$\mathtt{span}(\Lambda) = \{By \ | \ y \in \mathbb{R}^d\},$$

where d is the *dimension* of the lattice Λ and B its basis.

Definition 7 Let $\Lambda_1 \subseteq \mathbb{R}^n$ and $\Lambda_2 \subseteq \mathbb{R}^m$ be lattices and respective ranks n and m, and let $x_1, \ldots, x_n \in \mathbb{R}^n$ and $y_1, \ldots, y_m \in \mathbb{R}^m$ be respective bases. The *tensor product* $\Lambda_1 \otimes \Lambda_2 \subseteq \mathbb{R}^{nm}$ is defined as the *lattice* with basis $\left\{x_i \otimes y_j : \ i \in \{1, \ldots, n\}, \ j \in \{1, \ldots, m\}\right\}$. We note that $a \otimes b = (a_1, \ldots, a_n) \otimes (b_1, \ldots, b_m)$ with $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ is defined as the natural embedding in $\mathbb{R}^{nm}$ as below

$a \otimes b = (a_1 b_1, a_1 b_2, ..., a_1 b_m, a_2 b_1, ..., a_2 b_m, ..., a_n b_m) \in \mathbb{R}^{nm}$.

Definition 8 (*Closest vector problem*). Let $\Lambda \subset \mathbb{R}^d$ be a lattice. Given an arbitrary vector $\mathbf{t} \in \mathtt{span}(\Lambda)$. The vector $\mathbf{x}$ in Λ that minimizes the distance $||\mathbf{t} - \mathbf{x}||$ is called a *closest vector* to $\mathbf{t}$.

Although the closest vector problem is classified as NP-hard [10], there are some lattices where this problem can be solved efficiently. It is the case of integer lattice $\mathbb{Z}^n$, the root lattices A_n $(n \geq 1)$, D_n $(n \geq 2)$, E_n $(n = 6, 7, 8)$, the Leech lattice, and some cases of cyclotomic integer lattices $\mathbb{Z}[]$ (with $\alpha = p.q$, where p and q are prime).

The Root Lattice D_n

Definition and Basis of D_n

In the following, we recall the definition of the root lattice of type D_n $(n \geq 2)$, and give its generator matrix.

Definition 9 Let n be a positive integer. The subset D_n $(n \geq 2)$ of $\mathbb{R}^n$ defined by

$$D_n := \left\{ \mathbf{x} \in \mathbb{Z}^n \ : \ \langle \mathbf{x}, \overline{1} \rangle \textit{ is } \text{even} \right\}, \tag{4}$$

where $\overline{1} := (1, 1, \dots, 1)$, is a lattice of rank n in $\mathbb{R}^n$.

The shortest vectors in the lattice D_n $(n \geq 2)$ are all the permutations of $(\mp 1, \mp 1, 0, 0, ..., 0)$. The basis of the root lattice D_n is given in the following Lemma 1.

Lemma 1 (Basis of D_n $(n \geq 2)$) *A generator matrix of the lattice D_n is the $n \times n$-matrix* B *given by*

$$\mathrm{B} = \begin{pmatrix} -1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & & & \cdots & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{pmatrix}. \tag{5}$$

Before going on the characterization of the vectors of the root lattice $D_n \otimes D_m$, we will present a polynomial algorithm which solves the CVP in the root lattice D_n.

The Closest Vector Problem in D_n [3]

Given $x \in \mathbb{R}^n$, the closest point to x in D_n is whichever of $f(x)$ and $g(x)$ having an even sum of coordinates (one will have an even sum and the other will have an odd sum), where the function f and g are defined as follows: For an arbitrary $x_i \in \mathbb{R}$, we define the functions $f(x_i)$ and $w(x_i)$ for all $i = 1, ..., n$ as follows:

- if $x_i = 0$, then $f(x_i) = 0$ and $w(x_i) = 1$
- if $0 < m + \frac{1}{2} < x_i < m + 1$, then $f(x_i) = m$ and $w(x_i) = m + 1$
- if $-m - \frac{1}{2} \leq x_i \leq -m$, then $f(x_i) = -m$ and $w(x_i) = -m - 1$
- if $0 < m + \frac{1}{2} < x_i < m + 1$, then $f(x_i) = m + 1$ and $w(x_i) = m$
- if $-m - 1 < x_i < -m - \frac{1}{2}$, then $f(x_i) = -m - 1$ and $w(x_i) = -m$.

We also write $x_i = f(x_i) + \delta(x_i)$, so that $|\delta(x_i)| \leq \frac{1}{2}$ is the distance from x_i to the nearest integer.

Given that $x = (x_1, ..., x_n) \in \mathbb{R}^n$, let $k (1 \leq k \leq n)$, such that $|\delta(x_k)| \leq |\delta(x_i)|$ for all $1 \leq i \leq n$ and $|\delta(x_k)| = |\delta(x_i)|$ implies $k \leq i$. Then, $f(x) = (f(x_1), f(x_2), ..., f(x_k), ..., f(x_n))$ and $g(x)$ is defined by:

$g(x) = (f(x_1), f(x_2), ..., w(x_k), ..., f(x_n))$.

Our Result: The Closest Vector Problem in $D_n \otimes D_m$

We will start this section by the characterization of the vectors of the root lattice $D_n \otimes D_m$ $(n, m \geq 2)$ as below. We first recall the definition of the tensor product:

Definition 10 Let $\Lambda_1 \subseteq \mathbb{R}^{n_1}$ and $\Lambda_2 \subseteq \mathbb{R}^{n_2}$ be lattices of, respectively, ranks n_1 and n_2, let $a_1, ..., a_{n_1} \in \mathbb{R}^{n_1}$ and $b_1, ..., b_{n_2} \in \mathbb{R}^{n_2}$ be their respective bases. The tensor product $\Lambda_1 \otimes \Lambda_2 \subset \mathbb{R}^{n_1 n_2}$ is defined as the lattice with basis $\{a_i \otimes b_j \ : \ i \in \{1, ..., n_1\}, \ j \in \{1, ..., n_2\}\}$.

Here, $x \otimes y = (x_1, ..., x_{n_1}) \otimes (y_1, ..., y_{n_2})$ with $x \in \mathbb{R}^{n_1}$ and $y \in \mathbb{R}^{n_2}$ can be seen as an element of $\mathbb{R}^{n_1 n_2}$ as follows : $(x_1 y_1, x_1 y_2, ..., x_1 y_{n_1}, x_2 y_1, ..., x_{n_1} y_{n_2}) \in \mathbb{R}^{n_1 n_2}$.

Characterization of the Vectors of the Root Lattice $D_n \otimes D_m$

The root lattice $D_n \otimes D_m \subseteq \mathbb{Z}^{nm}$ $(n, m \geq 2)$ consists of all elements $x = (x_{11}, ..., x_{1m}, x_{21}, ..., x_{2m}, ..., x_{n1}, ..., x_{nm}) \in \mathbb{Z}^{nm}$ satisfying the following conditions:

(1) $\sum_{i=1}^{n} x_{ij}$ even for all $j = 1, ..., m$
(2) $\sum_{j=1}^{m} x_{ij}$ even for all $i = 1, ..., n$.

[The notation $x = (x_{11}, ..., x_{1m}, x_{21}, ..., x_{2m}, ..., x_{n1}, ..., x_{nm})$ above means that there exist two vectors $u = (u_1, ..., u_n) \in D_n$ and $v = (v_1, ..., v_m) \in D_m$, such that $x_{ij} = u_i v_j$ for $i = 1, ..., n$ and $j = 1, ..., m$.]

Indeed, we have $(x_{11}, ..., x_{1m}, ..., x_{2m}, ..., x_{n1}, ..., x_{nm}) = (u_1 v_1, ..., u_1 v_m, u_2 v_1, ..., u_2 v_m, ..., u_n v_1, ..., u_n v_m) \in D_n \otimes D_m$. Since the sums $\sum_{i=1}^{n} u_i$ and $\sum_{j=1}^{m} v_j$ are even, then $\sum_{i=1}^{n} u_i v_j$ is even for all $j = 1, ..., m$ and $\sum_{j=1}^{m} u_i v_j$ is even for all $i = 1, ..., n$.

Remark 1 Let D_n and D_m $(n, m \geq 2)$be two root lattices. Then, D_{nm} is a full-rank sub-lattice of the lattice $D_n \otimes D_m$.

Indeed, the vector $x = (0, 0, 2, 1, 1, 0, -1, 1)$ is the vector of the root lattice D_8, because $0 + 0 + 2 + 1 + 1 + 0 - 1 + 1 = 4$, which is even. However, this vector is not in the root lattice $D_2 \otimes D_4$, because $\sum_{j=1}^{4} x_{1j} = x_{11} + x_{12} + x_{13} + x_{14} = 0 + 0 + 2 + 1 = 3$, which is odd.

Lemma 2 (Basis of $D_n \otimes D_m$) *Let D_n and D_m $(n, m \geq 2)$ be two root lattices, the basis $B_{n\otimes m} := \{b^{ij} : i = 1, ..., n \text{ and } j = 1, ..., m\}$ of the root lattice $D_n \otimes D_m$ is given by*

- $b^{11}_{1,1} = b^{11}_{1,2} = b^{11}_{2,1} = b^{11}_{2,2} = 1$
- $b^{i1}_{i-1,1} = b^{i1}_{i-1,2} = 1; b^{i1}_{i,2} = b^{i1}_{i;1} = -1$ for all $i = 2, ..., n$
- $b^{1j}_{1,j-1} = b^{1j}_{2,j-1} = 1; b^{1j}_{1,j} = b^{1j}_{2,j} = -1$ for all $j = 2, ..., m$
- $b^{ij}_{i-1,j-1} = b^{ij}_{i,j} = 1;\ b^{ij}_{i-1,j} = b^{ij}_{i,j-1} = -1$ for all $i = 2, ..., n$ and $j = 2, ..., m$
- 0 otherwise.

A Polynomial Algorithm for Solving the CVP in $D_n \otimes D_m$

We first present a general description of our CVP efficient algorithm in $D_n \otimes D_m$ $(n, m \geq 2)$ as below:

Description of the Algorithm

This algorithm takes as input a vector of a linear space spanned $span(D_n \otimes D_m)$ (where D_n and D_m are two root lattices of type D with $n, m \geq 2$) and returns a closest vector to this vector in $D_n \otimes D_m$ as follows:

Given a vector $t = (t_{11}, ..., t_{1m}, t_{21}, ..., t_{2m}, ..., t_{n1}, ..., t_{nm})$ of $span(D_n \otimes D_m) \subseteq \mathbb{R}^{nm}$.

We will start by determining the closest vector to t in the root lattice D_{nm}. To do this, we will calculate the functions $f(t) = (f(t_{11}), ..., f(t_{1m}), f(t_{21}), ..., f(t_{2m}), ..., f(t_{n1}), ..., f(t_{nm}))$ and $g(t) = (f(t_{11}), ..., f(t_{k(l-1)}), w(t_{kl}), f(t_{k(l+1)}), ..., f(t_{nm}))$ (where $f(t_{ij}) = \lfloor t_{ij} \rceil$ for all $i = 1, ..., n$ and $j = 1, ..., m$; and the function g is obtained by proceeding as in the case of a single root lattice of type D [4]). Given that the two functions f and g differ by only one component, and by the value 1, then either the sum of the function's coordinates f or g will be even .

Then, if the sum of all the coordinates of $f(t)$ is even, then $h := f$, else $h := g$. Thus, $h \in D_{nm}$. After determining the closest vector $h \in D_{nm}$ of t, the closest vector to h in $D_n \otimes D_m$ is obtained as follows:

We carry out the sums $\sum_{i=1}^{n} h(t_{ij})$ for all $j = 1, ..., m$ and $\sum_{j=1}^{m} h(t_{ij})$ for $i = 1, ..., n$. If all these sums are even, then $h \in D_n \otimes D_m$. Therefore, $x := h$. Else, we proceed as follows.

Then, we initialize the counters c, d, α, and β as follows: $c := 0$, $d := 0$, $\alpha := 1$, and $\beta := 1$. We calculate for each $i = 1, ..., n$ the sums $\sum_{j=1}^{m} h(t_{ij})$. Thus, for $i = 1, ..., n$ if $\sum_{j=1}^{m} h(t_{ij})$ odd, then $c := c + 1$; $u_\alpha := \sum_{j=1}^{m} h(t_{ij})$ and $\alpha = \alpha + 1$. We calculate also for each $j = 1, ..., m$ the sums $\sum_{i=1}^{n} h(t_{ij})$. As above, for $j = 1, ..., m$, if $\sum_{i=1}^{n} h(t_{ij})$ odd, then $d := d + 1$; $v_\beta := \sum_{j=1}^{m} h(t_{ij})$ and $\beta = \beta + 1$.

After calculating all the sums above, if $c = 0$ and $d = 0$, then $x := h$. Else, for each $r = 1, ..., c$, we denote by $f(h_{u_\alpha})$ and $g(h_{u_\alpha})$ the corresponding functions to the vector h as defined in the section "The Closest Vector Problem in D_n [3]". Similarly, for each $s = 1, ..., d$, we denote by $f(h_{v_\beta})$ and $g(h_{v_\beta})$ the corresponding functions to the vector h. Here, the functions $f(h_{u_\alpha})$ and $g(h_{u_\alpha})$ are associated with the vector h whose sum of the coordinates is equal to u_α. In the same way, the functions $f(h_{v_\beta})$ and $g(h_{v_\beta})$ are associated with the vector h whose sum of the coordinates is equal to v_β.

Thus, for all u_α and v_β, there exists a single common function of which all the sums of the coordinates are even. We will denote by q this function.

At the end of all these operations, we get the vector $x := q$. This process is performed at most $(n + m)$ times until all the sums $\sum_{i=1}^{n} h(t_{ij})$ for all $j = 1, ..., m$ and $\sum_{j=1}^{m} h(t_{ij})$ for $i = 1, ..., n$ are even. Thus, the news coordinates of the function that we obtain is the component of the vector $x \in D_n \otimes D_m$.

An such x is the closest vector of $t \in span(D_n \otimes D_m)$ in $D_n \otimes D_m$.

Algorithm 1 A CVP algorithm for the lattice $D_n \otimes D_m$.

Require: $n, m \geq 2$ and $t = (t_{11}, \ldots, t_{nm}) \in \mathtt{span}(D_n \otimes D_m)$.
Ensure: a closest vector x to t in $D_n \otimes D_m$.

1: $f1 := (\lfloor t_{11} \rceil, \ldots, \lfloor t_{nm} \rceil)$;
2: $g1 := (f(t_{11}), \ldots, f(t_{k(l-1)}), f(w_{kl}), f(t_{k(l+1)}), \ldots, f(t_{nm}))$; (where w_{kl} is define as in Section 3.2);
3: $u = [0, \ldots, 0]$; $v = [0, \ldots, 0]$;
4: $c := 0$; $d := 0$;
5: **if** $\sum_{i,j} f(t_{ij})$ even then
6: $\quad p := f1$;
7: $\quad$**else** $p := g1$;
8: **end if**;
9: **for** $i = 1, \cdots, n$ **do**
10: $\quad a := \sum_{j=1}^{m} p_{ij}$;
11: $\quad$**if** a odd then
12: $\quad\quad c := c + 1$;
13: $\quad\quad u_c := a$;
14: $\quad$**end if**;
15: **end for**;
16: **for** $j = 1, \cdots, m$ **do**
17: $\quad b := \sum_{i=1}^{n} p_{ij}$;
18: $\quad$**if** b odd then
19: $\quad\quad d := d + 1$;
20: $\quad\quad v_d := b$;
21: $\quad$**end if**;
22: **end for**;
23: **if** $c = 0$ and $d = 0$ then
24: $\quad x := p$;
25: **else**;
26: **for** $\alpha = 1, \cdots, c$ and $\beta = 1, \cdots, d$ **do**
27: $\quad$compute $f(p_{u_\alpha})$; $g(p_{u_\alpha})$; $f(p_{v_\beta})$; $g(p_{v_\beta})$; (see Subsection 4.2)
28: $\quad x := q$;(see Complexity analysis 4.2 below)
29: **end for**;
30: **end if**;
31: x is a closest vector of x in $D_n \otimes D_m$;

Complexity Analysis

About the complexity of this algorithm, we have the following:

From line 1 to line 2, we have 2 elementary operations. Indeed, we have only 2 assignments in these steps.

Line 3 has 4 elementary operations. Indeed, we have 4 assignments in this step.

From line 4 to line 8, we have 2 elementary operations. Indeed, we have 1 comparison and 1 assignment.

From line 9 to line 15, we have at most $3n$ elementary operations. Indeed, we have at most 3 operations inside the loop for which goes from 1 to n.

From line 16 to line 22, we have at most $3m$ elementary operations. Indeed, we have at most 3 operations inside the loop for which goes from 1 to m.

From line 23 to line 24, we have at most 3 elementary operations.

From line 26 to line 29, we have $n + m$ operations. Indeed, q is the vector whose coordinates are made up of a part of the coordinates whose sum is even in line 10 of our algorithm, and the rest of the coordinates of q supplemented by the coordinates obtained after line 27 of our algorithm. In this step, the algorithm uses the section "The Closest Vector Problem in D_n [3]" to determine each sub-coordinate for which the sub-vectors of each block are close to the associated target sub-vectors. Indeed, by determining the values whose distances with that of the associated sub-blocks are minimum, we will globally obtain the closest vector to the initial target vector. Given that the only operations used here are the comparisons and the additions, and that we have at most n blocks according to the index i, and at most m blocks according to the index j.

Thus, we will have at most $2 + 4 + 4 + 3n + 3m + 3 = 13 + 4n + 4m$ arithmetic operations;

since $\dfrac{13+4n+4m}{n+m} \longrightarrow cste$ when $n, m \longrightarrow \infty$, then the complexity of this algorithm is $O(n+m)$ arithmetic operations.

Example 1 Let $n=m=2$, and $x=(1.2,-1.2,-1.2,0.6) \in \mathtt{span}(D_2 \otimes D_2)$.

We have: $f=(1,-1,-1,1)$, and $g=(1,-1,-1,0)$;

since $1-1-1+1=0$, then $p:=f=(1,-1,-1,1) \in D_4$; and because $\sum_{i=1}^{2} p_{i1}=p_{11}+p_{21}=1-1=0$, $\sum_{i=1}^{2} p_{i2}=p_{12}+p_{22}=-1+1=0$, $\sum_{j=1}^{2} p_{1j}=p_{11}+p_{12}=1-1=0$ and $\sum_{j=2}^{2} p_{2j}=p_{21}+p_{22}=-1+1=0$, then $x:=p=(1,-1,-1,1)$.

Therefore, $\mathbf{x}=(1,-1,-1,1)$ is the closest vector of $t=(1.2,-1.2,-1.2,0.6)$ in $D_2 \otimes D_2$.

Example 2 Let $n=3$ *and* $m=2$, and $t=(2.8,-2.8,-2.8,4.6,-2.9,-3.3) \in \mathtt{span}(D_3 \otimes D_2)$.

We have: $f:=(3,-3,-3,5,-3,-3)$ and $g:=(3,-3,-3,4,-3,-3)$;

since $3-3-3+5-3-3=-4$, then $p:=f=(3,-3,-3,5,-3,-3)$;

For $i=1,\dots,3$, we have: $U_1=\sum_{j=1}^{2} p_{1j}=p_{11}+p_{12}=3-3=0$;

$U_2=\sum_{j=1}^{2} p_{2j}=p_{21}+p_{22}=-2+4=2$;

$U_3=\sum_{j=1}^{2} p_{3j}=p_{31}+p_{32}=-3-3=-6$; and for $j=1,\dots,2$, we have: $V_1=\sum_{i=1}^{2} p_{i1}=p_{11}+p_{21}+p_{31}=3-3-3=3$ and $V_2=\sum_{i=1}^{2} p_{i2}=p_{12}+p_{22}+p_{32}=-3+5-3=3$;

we have V_1 and V_2 odd. For the case of V_1, we take the coordinates p_{11}, p_{21}, p_{31} and we calculate f_1 and g_1 as below:

$f_1=(3,-3,-3)$ and $g_1=(3,-2,-3)$ where $p_{11}=3$, $p_{21}=-3,-2$ and $p_{31}=-3$.

For the case of V_2, we take the coordinates p_{12}, p_{22}, p_{32} and we calculate f_2 and g_2 as below:

$f_2=(-3,5,-3)$ and $g_2=(-3,4,-3)$ where $p_{12}=-3$, $p_{22}=5,4$ and $p_{32}=-3$;

since the sums of the coordinates of the vectors g_1 and g_2 are even, we choose $p_{21}=-2$ and $p_{22}=4$; thus, $x:=(3,-3,-2,4,-3,-3)$.

Therefore, the vector $x=(3,-3,-2,4,-3,-3)$ is the closest vector of $t=(2.8,-2.8,-2.8,4.6,-2.9,-3.3)$ in the root lattice $D_3 \otimes D_2$.

Conclusion

In this work, we successfully constructed a polynomial algorithm to solve the closest vector problem for the case of tensor product of two root lattice D_n and D_m that we noted $D_n \otimes D_m$ $(n, m \geq 2)$. Our future work will consist to generalize this algorithm to solve this problem for the case of tensor product of a finite number k of root lattices of type D_n $(n \geq 2)$ which we denote by $\bigotimes_{i=1}^{k} D_i$. We will also characterize the Voronoi region vectors in root lattice $D_n \otimes D_m$ and use it to propose another algorithm to solve Closest Vector Problem in lattice $D_n \otimes D_m \otimes D_p$ $(n, m, p \geq 2)$. After having proposed this, it will also be a question of comparing this new algorithm with that of this work.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Aggarwal D, Dadush D, Stephens-Davidowitz N. Solving the closest vector problem in 2^n time: the discrete gaussian strikes again! In: Guruswami V, editor. IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015. IEEE Computer Society; 2015. p. 563–582.
2. Ahuja RK, Magnanti TL, Orlin JB. Network flows-theory, algorithms and applications. Hoboken: Prentice Hall; 1993.
3. Conway JH, Sloane NJA. Fast quantizing and decoding and algorithms for lattice quantizers and codes. IEEE Trans Inf Theory. 1982;28(2):227–31.
4. Conway JH, Sloane NJA. Sphere Packings, Lattices and Groups, volume 290 of Grundlehren der mathematischen Wissenschaften. Berlin: Springer; 1988.
5. Dachman-Soled D, Ducas L, Gong H, Rossi M. LWE with side information: attacks and concrete security estimation. In: Micciancio D, Ristenpart T, editors. Advances in Cryptology-CRYPTO 2020-40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II, volume 12171 of Lecture Notes in Computer Science. Springer; 2020. p. 329–358.
6. Ducas L. Shortest vector from lattice sieving: a few dimensions for free. In: Nielsen JB, Rijmen V, editors. Advances in Cryptology-EUROCRYPT 2018-37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part I, volume 10820 of Lecture Notes in Computer Science. Springer; 2018. p. 125–145.
7. Ducas L, van Woerden WPJ. The closest vector problem in tensored root lattices of type A and in their duals. Des Codes Cryptogr. 2018;86(1):137–50.
8. Ducas L, van Woerden WPJ. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In: Dunkelman O, Dziembowski S, editors. Advances in Cryptology-EUROCRYPT 2022-41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30-June 3, 2022, Proceedings, Part III, volume 13277 of Lecture Notes in Computer Science. Springer; 2022. p. 643–673.
9. Gersho A. Asymptotically optimal block quantization. IEEE Trans Inf Theory. 1979;25(4):373–80.
10. Goldreich O, Goldwasser S, Halevi S. Public-key cryptosystems from lattice reduction problems. In: Kaliski Jr BS, editor. Advances in Cryptology-CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August

17–21, 1997, Proceedings, volume 1294 of Lecture Notes in Computer Science. Springer; 1997. p. 112–131.
11. Lyubashevsky V, Peikert C, Regev O. On ideal lattices and learning with errors over rings. In: Gilbert H, editor. Advances in Cryptology-EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco/French Riviera, May 30-June 3, 2010. Proceedings, volume 6110 of Lecture Notes in Computer Science. Springer; 2010. p. 1–23.
12. Lyubashevsky V, Peikert C, Regev O. A toolkit for ring-lwe cryptography. IACR Cryptol ePrint Arch. 2013;2013:293.
13. Micciancio D, Voulgaris P. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In: Schulman LJ, editor. Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010. ACM; 2010. p. 351–358.
14. Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput. 1997;26(5):1484–509.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Discrete Mathematics, Algorithms and Applications
(2022) 2250151 (15 pages)
© World Scientific Publishing Company
DOI: 10.1142/S1793830922501518

World Scientific
www.worldscientific.com

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

Sieve algorithms for some orthogonal integer lattices

Arnaud Gires Fobasso Tchinda
Department of Mathematics, The University of Yaounde 1
P.O. Box 812 Yaounde Cameroon
fobass1989@gmail.com

Emmanuel Fouotsa*
Department of Mathematics, Higher Teacher Training College
The University of Bamenda. P.O. Box 39 Bambili Cameroon
emmanuelfouotsa@yahoo.fr

Celestin Nkuimi Jugnia
Department of Mathematics, The University of Yaounde 1
P.O. Box 812 Yaounde Cameroon
nkuimi@yahoo.co.uk

Received 7 March 2022
Revised 20 July 2022
Accepted 15 August 2022
Published 26 September 2022

Communicated by Zhipeng Cai

We propose in this work a Sieve algorithm that we called *OrthogonalInteger* sieve algorithm for some orthogonal integer lattices and particularly the case of integer lattices $\Lambda \subset \mathbb{Z}^n$, root lattices of type A_n $(n \geq 1)$ and of type D_n $(n \geq 2)$. In these cases, we use the famous LLL algorithm to find the shortest vector of these lattices. Indeed, in general, a sieve algorithm builds a list of short random vectors which are not necessarily in the lattice, and try to produce short lattice vectors by taking linear combinations of the vectors in the list. But in our case, we built a list of short vectors in the lattice. From the first column of the LLL-reduced basis of the considered basis, we have the list of at least n and at most 2^n short vectors for the general case (where n is the dimension of the lattice) of orthogonal integer lattices $\Lambda \subset \mathbb{Z}^n$. For the lattices $\mathbb{Z}^n$, A_n $(n \geq 1)$ and D_n $(n \geq 2)$, we have, respectively, $2n$, $n(n+1)$ and $2n(n-1)$ short vectors. The proposed sieve algorithm for integer lattice $\mathbb{Z}^n$ runs in space $O(2n)$ and the *OrthogonalInteger* sieve algorithm performs $O(n2^n)$ arithmetic operations and is polynomial in space.

Keywords: Lattices; sieving; orthogonal lattice; integer lattice; shortest vector problem.

Mathematics Subject Classification 2020: 11H71, 11H06

*Corresponding author.

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

1. Introduction

A lattice is a mathematical object which takes a set of vectors in $\mathbb{R}^n$ and combines them in all possible integer linear combinations. One of the central problems of lattices theory is the Shortest Vector Problem (SVP) which consists in finding the shortest nonzero vector in the lattice. SVP has been extensively studied as purely mathematical problem, being central in the study of the geometry of numbers and as algorithm problems, having many applications in communication theory and computer science. There are two main algorithmic techniques for solving exact SVP: enumeration and sieving. Enumeration algorithms were initiated by Pohst [14] in 1981 and one of the best enumeration algorithm was given by Kannan in 1983 [11]. This method runs in $n^{o(n)}$ time but polynomial in space. The main idea of Sieve Algorithm is to randomly select lattice vectors, then compare them in order to end up getting the shortest lattice vectors, running the algorithm for many steps. This method was introduced by Ajtai *et al.* in 2001 [1] lowering the time complexity of the SVP to $2^{o(n)}$, but required $2^{o(n)}$ space and randomness. In 2010, Micciancio *et al.* presented GaussSieve [12], the first sieving heuristic that outperformed enumeration routines. In 2011, Panagiotis proposed a new heuristic sieving algorithm [17] that performed quite well in the practice with estimated running time $2^{0,52n}$ and space complexity $2^{0,2n}$. In 2017, Leo Ducas [8] exploits the fact that *sieving* returns many short vectors, rather than only one to propose a new practical improvement for sieve algorithms. In this work, we give a list of all short vectors of the particular case of orthogonal integer lattices $\mathbb{Z}^n$. The proposed algorithm is polynomial and requires $O(2n)$ in space. We also propose an enumeration algorithm which will allow us to obtain the list of shortest vectors in all orthogonal integer lattices $\Lambda \subseteq \mathbb{Z}^n$. This algorithm runs in $O(n2^n)$ time and can be polynomial in space and the list of short vectors obtained enable to solve the shortest independent vector problem (SIVP) "which is an NP-Hard problem in cryptography" [2] for some orthogonal integer lattices. Indeed, when we obtain the list of short vectors in some orthogonal integer lattice of dimension n, we can extract a family of n independent vectors with equal norms. This family of vectors is a solution to the shortest independent vector problem in the lattice. Note however that when the dimension n is large, the list of shortest vectors becomes larger, and consequently the search for independent vectors of this list also becomes more complex. This is possible for some integer lattice $\mathbb{Z}^n$, root lattices of type D_n $(n \geq 2)$ and A_n $(n \geq 1)$ and their duals. For correctness, a Maple computer software implementation of the algorithm has been done.

The paper is organized as follows. In Sec. 2, we recall some key concepts such as successive minima, Minkowski's theorem, and some properties of orthogonal lattices that will be useful in the paper. In Sec. 3, we recall the Gram Schmidt process, the LLL-reduction process and we propose a polynomial algorithm to determine an orthogonal integer basis for a given integer lattice. Our main result is presented in Sec. 4, where we describe a polynomial algorithm which returns a list of exactly $2n$ short vectors for the case of the orthogonal integer lattice $\mathbb{Z}^n$. We also present in Sec. 4 an algorithm which gives at least n and at most 2^n short vectors of general

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

orthogonal integer lattices $\Lambda \subset \mathbb{Z}^n$. This algorithm runs in $O(n2^n)$ time and can be polynomial in space. The work is concluded in Sec. 5.

2. Preliminaries on Lattices

In this section, we recall some key concepts such as successive minima, Minkowski's theorem and some properties of orthogonal lattices.

Throughout this work, for any positive integer n, we use the Euclidean inner product on $\mathbb{R}^n$ which is defined by $\langle \mathbf{x}, \mathbf{y} \rangle := x_1y_1 + x_2y_2 + \cdots + x_ny_n$ for $\mathbf{x} := (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} := (y_1, y_2, \ldots, y_n)$ in $\mathbb{R}^n$. The Euclidean norm on $\mathbb{R}^n$ is defined as follows: $\|\mathbf{x}\| := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. We denote by $B(\mathbf{x}, r)$ the closed Euclidean n-dimensional ball of radius r centered at $\mathbf{x}$ such that: $B(\mathbf{x}, r) = \{\mathbf{y} \in \mathbb{R} : \|\mathbf{x}-\mathbf{y}\| < r\}$. The ball centered at zero will be simply denoted $B(r)$.

2.1. *Basic definition of lattices*

More details about these definitions can be found in [10, 16]. A lattice is a discrete additive subgroup of $\mathbb{R}^n$, for any positive integer n. We deal exclusively with any lattice Λ of rank d, which is generated by the set of all integer linear combinations of d linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d$ in $\mathbb{R}^n$ as follows:

$$\Lambda = \left\{ \sum_{i=1}^{d} z_i \mathbf{b}_i : (z_1, z_2, \ldots, z_d) \in \mathbb{Z}^d \right\}. \tag{2.1}$$

The set of vectors $B = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d\}$ is called the basis of the lattice. A lattice has several invariant such as rank, dimension, volume, the determinant of the lattice, the first minimum of the lattice, and the nth-successive minima of the lattice. We will define these notions and give some associated properties in the following.

The rank of a lattice Λ is defined as the number of linearly independent vector in any basis for that lattice. A lattice Λ is said to be a full-rank lattice when $n = d$. The determinant (volume) of a lattice Λ of dimension n and rank d, denoted $\mathtt{det}(\Lambda)$ is defined by

$$\mathtt{det}(\Lambda) = \sqrt{\mathtt{det}(B^T B)}, \tag{2.2}$$

where B^T is the transpose of the matrix B.

If the lattice Λ is of full rank, then B is a square matrix and consequently, we have

$$\mathtt{det}(\Lambda) = |\mathtt{det}(B)|. \tag{2.3}$$

Remark 2.1. The determinant of a lattice is independent of the choice of the basis B.

Let Λ be a lattice and B one basis, the fundamental parallelepiped of Λ, denoted $\mathbf{P}(B)$ is defined as

$$\mathbf{P}(B) = \{Bx \,|\, x \in \mathbb{R}^n, \forall\, i : 0 \leq x_i < 1\}. \tag{2.4}$$

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

Lemma 2.2. *Let Λ be a lattice and $b_1, \ldots, b_d \in \Lambda$ be d linearly independent lattice vectors. Then $b_1, \ldots, b_d$ form a basis of Λ if and only if $\boldsymbol{P}(b_1, \ldots, b_d) \cap \Lambda = \{0\}$.*

In the rest of this work, we will use full-rank lattice. Specifically, in a lattice Λ, any nonzero vector v has a strictly positive length. But the problem which arises is that of knowing if this length is relatively small compared to the other vectors of the lattice. This leads us to introduce the notion of successive minima of a lattice as below.

2.2. *Successive minima*

For a given lattice Λ, we denote $\lambda_1(\Lambda)$ the minimum Euclidean norm of vectors in $\Lambda\backslash\{0\}$. More generally, for all $1 \leq i \leq n$, we define the ith- minimum as follows: $\lambda_i(\Lambda) = \min_{v_1,\ldots,v_i \in \Lambda} \max_{j \leq i} \parallel v_j \parallel$ (where $v_1, \ldots, v_i$ are linearly independent).

Definition 2.3 ([4]). For any lattice Λ with a basis B, the minimum distance of Λ is the smallest distance between any two lattices points given as follows:

$$\lambda(\Lambda) = \inf\{\|x - y\| : x, y \in \Lambda, x \neq y\}.$$

Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice. We say that Λ' is a sublattice of Λ if $\Lambda' \subseteq \Lambda$ is a lattice as well. If Λ' is a sublattice of Λ, then $\lambda_i(\Lambda) \leq \lambda_i(\Lambda')$ for $i \leq \mathtt{dim}(\Lambda')$ (where $\mathtt{dim}(\Lambda')$ is the dimension of lattice Λ').

Theorem 2.4 ([4]). (*First theorem of Minkowski*) *For any full-rank lattice $\Lambda \subseteq \mathbb{R}^n$, we have*

$$\lambda_1(\Lambda) \leq \sqrt{n}(\boldsymbol{det}(\Lambda))^{1/n}, \tag{2.5}$$

where $\lambda_1(\Lambda)$ denote the minimum Euclidean norm of vectors in $\Lambda\backslash\{0\}$.

The proof of this theorem requires the following results.

Theorem 2.5. *Let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice. Then for any symmetric central set S, if $\boldsymbol{vol}(S) > 2^n \boldsymbol{det}(\Lambda)$, then S contains a nonzero point of the lattice.*

Proposition 2.6. *The volume of a ball of dimension n and radius r is $\boldsymbol{vol}(B(r)) \geq (\frac{2r}{\sqrt{n}})^n$.*

The above results enable to conclude that the minimum distance can be equivalently defined as the length of the shortest nonzero lattice vector as follows:

$$\lambda(\Lambda) = \inf\{\|v\| : v \in \Lambda\diagdown\{0\}\}. \tag{2.6}$$

For the case of random lattices, we have an approximation of the minimum distance called Gaussian heuristic. It is defined explicitly as follows.

Definition 2.7 ([12, 13]). For all lattices Λ, the Gaussian heuristic $gh(\Lambda)$ gives the expected first minimum and for a full rank lattice $\Lambda \subseteq \mathbb{R}^n$, $gh(\Lambda)$ is defined as

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

follows:

$$gh(\Lambda) = \sqrt{\frac{n}{2\pi e}}.\mathrm{vol}(\Lambda)^{1/n}. \tag{2.7}$$

We also denote $gh(n)$ for $gh(\Lambda)$ of n-dimensional lattice Λ of volume 1: $gh(n) = \sqrt{\frac{n}{2\pi e}}$.

The Gaussian heuristic says that a shortest nonzero vector in a randomly chosen lattice will satisfy $v_{\text{shortest}} \approx gh(\Lambda)$.

In the following, we will define the particular lattices A_n ($n \geq 1$) and D_n ($n \geq 2$), also called root lattices.

Definition 2.8 ([5]). Let $n \geq 2$ be an integer, the root lattice $D_n \subset \mathbb{R}^n$ of rank n is defined as follows:

$$D_n := \left\{ (x_1, \ldots, x_n) \in \mathbb{Z}^n : \sum_{i=1}^{n} x_i \text{ is even} \right\}. \tag{2.8}$$

Let n be a positive integer, the root lattice $A_n \subset \mathbb{R}^n$ of rank n is defined as follows:

$$A_n := \left\{ (x_1, \ldots, x_{n+1}) \in \mathbb{Z}^{n+1} : \sum_{i=1}^{n+1} x_i = 0 \right\}. \tag{2.9}$$

From this, the shortest vectors of root lattice of type D_n and A_n are, respectively, all the permutations of the vectors $(\pm 1, \pm 1, 0, \ldots, 0)$ and $(1, -1, 0, \ldots, 0)$.

In the following, we will define orthogonal lattices and give the relation with integer lattices.

2.3. *Orthogonal lattices*

Definition 2.9 ([3]). A lattice Λ is said to be orthogonal if it has a basis B such that the rows of B are pairwise orthogonal vectors. In other words, a lattice Λ is said to be orthogonal if it is generated by set of pairwise orthogonal vectors. We recall that a basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ is orthogonal if and only if:

- $\langle b_i, b_i \rangle \neq 0$ for all i and;
- $\langle b_i, b_j \rangle = 0$ for all $i \neq j$.

Example 2.10. $\mathbb{Z}^n$ is an orthogonal lattice. Indeed, the basis of $\mathbb{Z}^n$ is $B = (b_1, \ldots, b_n)$ where $b_1 = (1, 0, \ldots, 0)$; $b_2 = (0, 1, 0, \ldots, 0)$; $b_{n-1} = (0, \ldots, 0, 1, 0)$ and $b_n = (0, \ldots, 0, 1)$.

Definition 2.11 ([16]). Let $B = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ be a basis of a lattice Λ of rank n. The orthogonality defect of the basis B is the following quantity:

$$\delta^{\top}(B) = \frac{\prod_{i=1}^{n} \|b_i\|}{\mathtt{det}(B)}. \tag{2.10}$$

Remark 2.12. $\delta^{\top}(B) \geq 1$ and if B is orthogonal, then $\delta^{\top}(B) = 1$. Thus if B is orthogonal, then $\mathtt{det}(B) = \prod_{i=1}^{n} \|b_i\|$

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

3. Lattice Reduction

In this section, we will recall some lattice reductions allowing either to determine a short vector, or a list of short vectors. We will also propose an algorithm which determines the orthogonal basis of a given integer lattice. We start with the description of Gram–Schmidt Orthogonalization.

Gram–Schmidt orthogonalization [10, 15, 16]

The Gram–Schmidt orthogonalization algorithm is an iterative approach for orthogonalizing vectors of a given basis. The first vector b_1 of a given basis B is taken as a reference and the second vector b_2 is projected onto an $(n-1)-$ hyper plane perpendicular to b_1. The third vector b_3 is projected onto a $(n-2)-$ hyper plane perpendicular to the plane defined by b_1 and b_2. This process continues in an iterative way until all degrees of freedom are exhausted. The new orthogonal vectors are denoted by b_i^* and the orthogonal basis obtained is denoted as B^*.

$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^* \text{for all } 1 \leq j < i \leq n, \tag{3.1}$$

where $\mu_{ij} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}$.

3.1. *Orthogonal basis of integer lattices*

Although the vectors of B^* are over rational field, by multiplying the basis B^* by the least common multiple (lcm) of the denominators of the coordinates, we obtain the basis $B^\perp$ (with integer coordinates) with pairwise orthogonal rows. This basis $B^\perp$ is an orthogonal basis of the lattice $\Lambda(B)$.

Example 3.1. Given the base $B = (b_1, b_2, b_3)$ with $b_1 = (1, 1, 1)$; $b_2 = (-1, 0, 2)$ and $b_3 = (3, 5, 6)$. We want to determine $B^\perp$.

The Gram-Schmidt Orthogonalization of B is given by: $B^* = (b_1^*, b_2^*, b_3^*)$ with $b_1^* = (1, 1, 1)$; $b_2^* = (-\frac{4}{3}, -\frac{1}{3}, \frac{5}{3})$ and $b_3^* = (-\frac{3}{7}, \frac{9}{14}, -\frac{3}{14})$; since $\mathrm{lcm}(3, 7, 14) = 42$, we have $b_1^\perp = 42 \times b_1^* = (42, 42, 42)$; $b_2^\perp = 42 \times b_2^* = (-56, -14, 70)$ and $b_3^\perp = 42 \times b_3^* = (-18, 27, -9)$. Therefore, $B^\perp = (b_1^\perp, b_2^\perp, b_3^\perp)$ is an orthogonal basis (with integer coordinates) of the lattice $\Lambda(B)$.

Lemma 3.2. *Let a lattice Λ with a basis B. If $B^\perp$ is its orthogonal basis, then $\lambda_1(\Lambda) \leq \lambda_1(\Lambda^\perp)$. Where $\lambda_1(\Lambda)$ and $\lambda_1(\Lambda^\perp)$ are, respectively, the minimum distance of the lattices Λ and $\Lambda^\perp$.*

Proof. We use the fact that for every orthogonal lattice, we have only one operation (swap) for all the vectors of the basis and we have the result. □

In section, we proceed to lattice reduction assuming that an orthogonal basis is always given.

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

3.2. *Orthogonal reduced basis of integer lattices*

Given an orthogonal basis $B^{\perp}$ of an integer lattice $\Lambda \subseteq \mathbb{Z}^n$, Algorithm 1 returns a reduced basis $B^{\perp_1}$ of $B^{\perp}$, i.e., a basis with vectors shorter than those of $B^{\perp}$. We start by calculating the gcd of the components of each vectors of $B^{\perp}$. After that, we divide all these vectors by this gcd. Finally, we perform permutations between these vectors in order to achieve the successive minima. The following algorithm illustrates this description.

Example 3.3. Given the basis $B = \begin{pmatrix} 1 & -1 & 3 \\ 1 & 0 & 5 \\ 1 & 2 & 6 \end{pmatrix}$ with $b_1 = (1,1,1)$; $b_2 = (-1,0,2)$ and $b_3 = (3,5,6)$. The Gram–Schmidt orthogonalization of B is given by: $B^* = \begin{pmatrix} 1 & -\frac{4}{3} & -\frac{3}{7} \\ 1 & -\frac{1}{3} & \frac{9}{14} \\ 1 & \frac{5}{3} & -\frac{3}{14} \end{pmatrix}$ with $b_1^* = (1,1,1)$; $b_2^* = (-\frac{4}{3}, -\frac{1}{3}, \frac{5}{3})$ and $b_3^* = (-\frac{3}{7}, \frac{9}{14}, -\frac{3}{14})$; since $lcm(3,7,14) = 42$, we have: $B^{\perp} = \begin{pmatrix} 42 & -56 & -18 \\ 42 & -14 & 27 \\ 42 & 70 & -9 \end{pmatrix}$ with $b_1^{\perp_1} = \frac{1}{42} \times (42,42,42) = (1,1,1)$; $b_2^{\perp_1} = \frac{1}{14} \times (-56,-14,70) = (-4,-1,5)$ and $b_3^{\perp_1} = \frac{1}{9} \times (-18,27,-9) = (-2,3,-1)$; therefore, since $\|b_3^{\perp_1}\| < \|b_2^{\perp_1}\|$ then, $b_2^{\perp_1} = b_3^{\perp_1} = (-2,3,-1)$; and $b_3^{\perp_1} = b_2^{\perp_1} = (-4,-1,5)$; since $\|b_1^{\perp_1}\| \leq \|b_2^{\perp_1}\|$, the vectors $b_1^{\perp_1}$ and $b_2^{\perp_1}$ remains the same and we have the following reduced basis: $B^{\perp_1} = \begin{pmatrix} 1 & -2 & -4 \\ 1 & 3 & -1 \\ 1 & -1 & 5 \end{pmatrix}$

We recall that the goal of lattice basis reduction is to find a basis with short vectors and orthogonal to each other. We also know that the Gram–Schmidt process does not preserve the structure of integer lattice. It would be interesting to focus on the LLL-reduction which used Gram–Schmidt process and returns integer vectors. The most usual notion of reduction is probably the LLL-reduction. The LLL-

Algorithm 1. Reduced($B^{\perp}$)

Require: The orthogonal basis $B^{\perp} = (b_1^{\perp}, \ldots, b_n^{\perp})$ of a lattice Λ.
Ensure: A reduced basis $B^{\perp_1}$ of the basis $B^{\perp}$.

```
for i from 1 to n do
    b_i^{⊥1} ← b_i^⊥ / gcd(a_i); (where a_i's are the components of the vector b_i^⊥)
end for
for j from n to 1 do
    if ||b_j^{⊥1}|| < ||b_{j−1}^{⊥1}|| then then
        swaps(b_j^{⊥1}, b_{j−1}^{⊥1}); (permutation between vectors b_j^{⊥1} and b_{j−1}^{⊥1})
    end if
end for
return B^{⊥1}
```

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

reduction is one of the most commonly used. Let $\frac{1}{4} < \delta < 1$, let $B = (b_1, \ldots, b_n) \in \mathbb{Z}^{n\times n}$ be a basis of a lattice. We say that B is size-reduced if all Gram–Schmidt coefficients satisfy $|\mu_{ij}| \leq \frac{1}{2}$. We say that B satisfies the Lovàsz conditions if for all $i \in \{1, \ldots, n\}$ we have $\delta\|b_i^*\|^2 \leq \|b_{i+1}^*\|^2 + \mu_{i+1,i}\|b_i^*\|^2$.

A basis B satisfying both the size-reduced and the Lovàsz conditions is said to be LLL-reduced. The LLL algorithm is given in [7, 16] and it is shown that the number of LLL swaps is $O(n^2 \lg\|B\|)$. The LLL-reduction implies that the norms of the Gram–Schmidt-orthogonalization vectors never drop too fast. Indeed the vectors are not far from being orthogonal. The most famous problem of lattice theory is the shortest vector problem (SVP), and the LLL-reduction gives a solution of this problem.

3.3. *Shortest vector problem* (*SVP*)

The most important computational problem in lattices is the shortest vector problem. The shortest vector problem asks to find a non zero lattice vector of small norm for a given lattice basis as input. This norm is called the first minimum $\lambda_1(\Lambda)$ or the minimum distance and is in general unique up to the sign. This means that: given a basis of a lattice Λ, find a lattice vector whose norm is exactly $\lambda_1(\Lambda)$.

This problem is classified as NP-hard [6, 10]. Minkowski's theorem gives a simple way to bound the length of the shortest lattice vector. Another variant of this problem is the shortest independent vector problem (SIVP) [9] which asks to find a linearly independent set $\{v_1, \ldots, v_n\}$ such that all vectors have length at most $\gamma \cdot \lambda_1(\Lambda(B))$ for a given lattice basis B as input (where $\gamma \geq 1$) [2]. LLL-reduction does not solve this problem for all lattices. Indeed, for random lattices, one uses the Gaussian heuristic and Gauss reduction to obtain the list of short vectors of the lattice.

Definition 3.4 ([12]). For two given vectors $u, v \in \Lambda$, if $\max(\|u\|, \|v\|) \leq \min(\|u - v\|, \|u + v\|)$, then u, v are called Gauss-reduced.

Let L be a list of N vectors from a lattice $\Lambda(B)$. If for any two different vectors v_i, v_j $(i, j = 1, \ldots, N\ i \neq j)$ in L, v_i and v_j are Gauss-reduced, then the list L is called pairwise-reduced.

When solving the shortest vector problem, $gh(\Lambda)$ is usually regarded as the expected norm of the shortest vector. In the following, we will present the notion of *orthogonaInteger* Sieve algorithm which is the exact method in practice to solve the shortest vector problem in orthogonal integer lattices $\Lambda(B) \subset \mathbb{Z}^n$, where n is the dimension of lattice Λ.

4. Our Main Result: OrthogonalInteger Sieve

In this section, we give our main result consisting of a Sieve algorithm for integer lattices. We will first define some important notions that we will use. We will denote L, a list to be constructed, containing all vectors of orthogonal integer lattices

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

$\Lambda(B) \subseteq \mathbb{Z}^n$ such that their norm equal to the minimal distance. Along the way, we denote H a list used to build the list L. It is the set of all vectors obtained by performing permutations of the coordinates of the vectors u and $-u$ (where u is the first short vector obtained by LLL-reduction). We will say that there is a collision if there is a repetition of the vectors in the list H.

We recall that a lattice Λ is said to be orthogonal if it is generated by a set of pairwise orthogonal vectors. If two vectors are orthogonal, then the angle between them is equal to $\frac{\pi}{2}$. The number of vectors in canonical basis of the integer lattice $\mathbb{Z}^n$ is n. Considering the structure of an orthogonal basis in dimension 2, the angle between the two vectors u and v is $\frac{\pi}{2}$. We know that an orthogonal basis has generally large integer coordinates because each vectors is multiplied by the lcm of the denominators of all the vectors of the basis obtained by the Gram–Schmidt Orthogonalization. Since the vectors are pairwise orthogonal, we cannot use reduction coefficient's process to reduce them. Indeed, the coefficients $\mu_{i,j}$ are all zero for each i, j. Thus, for the case of orthogonal lattices, we will only have the permutations process to carry out the successive minima corresponding to this basis. Since the first minima of LLL-reduced basis is less than the first minima of an integer orthogonal reduced basis that we have denoted by $B^{\perp_1}$, then we will use the LLL-reduced basis to find the list of shortest vectors in the general case of orthogonal integer lattice $\Lambda \subseteq \mathbb{Z}^n$. Therefore, we will initialize the empty list L, and the number of collisions by $C = 0$. After that, we use LLL-reduced basis to obtain a short vector of this lattice. Because the opposite of this short vector is also a short vector, we can use symmetry of different axes to see that all their permutations are also in the lattice, including shortest vectors. The algorithm that we are going to propose in this work, will output at least n and at most 2^n shortest vectors by using the first vector obtained from the LLL-reduced basis.

Thus, for the case of orthogonal lattice $\mathbb{Z}^n$, we know that $B^{\perp}_{\mathbb{Z}^n} = \{e_1, \ldots, e_n\}$, where $e_1 = (1, 0, \ldots, 0)$, $e_2 = (0, 1, \ldots, 0), \ldots, e_n = (0, 0, \ldots, 0, 1)$. Thus this algorithm returns the list $L = \{-e_n, \ldots, -e_1, e_1, \ldots, e_n\}$; which gives exactly the $2n$ shortest vectors of the lattice $\mathbb{Z}^n$.

Therefore, in the case of integer lattice $\mathbb{Z}^n$, we can obtain the list of all *shortest vectors* by the following simple enumeration algorithm:

Example 4.1. For $n = 4$, the orthogonal basis of $\mathbb{Z}^4$ is given by: $B^{\perp} = (e_1, e_2, e_3, e_4)$ where $e_1 = (1, 0, 0, 0)$, $e_2 = (0, 1, 0, 0)$, $e_3 = (0, 0, 1, 0)$ and $e_4 = (0, 0, 0, 1)$.

Algorithm 2. OrthogonalSieve($\mathbb{Z}^n$)

Require: The dimension n.
Ensure: A list L of *shortest* vectors.

1: $B^{\perp} \leftarrow (e_1, \ldots, e_n)$ (orthogonal basis of $\mathbb{Z}^n$);
2: $L \leftarrow (-e_n, -e_{n-1}, \ldots, -e_1, e_1, \ldots, e_{n-1}, e_n)$;
3: **return** L

Therefore the list L of shortest vectors is given by: $L = \{-e_4, -e_3, -e_2, -e_1, e_1, e_2, e_3, e_4\}$.

Lemma 4.2. *Let Λ be a full rank integer lattice of dimension n. Λ has at least n and at most $N = n!.2^n$ shortest vectors. Particularly,*

(1) *The integer lattice $\mathbb{Z}^n$ has exactly $2n$ shortest vectors*;
(2) *The root lattice of type A_n ($n \geq 1$) has exactly $n(n+1)$ shortest vectors*;
(3) *The root lattice of type D_n ($n \geq 2$) has exactly $2n(n-1)$ shortest vectors.*

Proof. Let Λ be a full rank integer lattice of dimension n. We know that there exists a vector $v = (v_1, \ldots, v_n) \in \Lambda \backslash \{0\}$ such that $\|v\| = \lambda_1(\Lambda)$. We also know that $u = -v$ is another shortest vector in Λ. Likewise, all the permutations of the coordinates of v and u are a shortest vector of the lattice. The vector v has at most $n!$ permutations and the vector u has also at most $n!$ permutations. Thus, we have at most $(n!)^2$ permutations possible for one of these vectors. Moreover, the vectors u and v have at most 2^n possibilities to combine symmetrically by the different axes. Therefore, we have at most $n!.2^n$ shortest vectors in integer lattices.

For the case of integer lattice $\mathbb{Z}^n$, we know that the n vectors of canonical basis are the shortest vectors of this lattice. Since their opposites are also the shortest vectors of $\mathbb{Z}^n$, we have exactly $2n$ shortest vectors in $\mathbb{Z}^n$.

Since the short vectors of the root lattices of type A_n ($n \geq 1$) are the permutations of the vector $(+1, -1, 0, \ldots, 0)$, then we will have exactly $\frac{(n+1)!}{(n-1)!} = n(n+1)$ short vectors in this particular lattice. Thus we will have exactly $n(n+1)$ short vectors in root lattices of type A_n.

About the root lattices of type D_n ($n \geq 2$), we also know that all the short vectors are the permutations of the vector $(\pm 1, \pm 1, 0, \ldots, 0)$ with the condition that the sum of all the components is even. Thus we will have three possible following cases: the permutations of the vector $(+1, -1, 0, \ldots, 0)$, the permutations of the vector $(+1, +1, 0, \ldots, 0)$ and the permutations of the vector $(-1, -1, 0, \ldots, 0)$.

This means that, we will have exactly $\frac{n!}{(n-2)!} + \frac{n!}{2!(n-2)!} + \frac{n!}{2!(n-2)!} = 2n(n-1)$. Therefore, we will have exactly $2n(n-1)$ short vectors in root lattices of type D_n. □

Corollary 4.3. *Given a basis B of the orthogonal lattice $\mathbb{Z}^n$, we can obtain the list L of shortest vectors of this lattice in space $O(2n)$.*

Proof. Let B be a basis of the orthogonal lattice $\mathbb{Z}^n$. The canonical basis permits to obtain exactly $2n$ short vectors of this lattice. Then these vectors will be obtained in space $O(2n)$. □

We are now going to propose an enumeration algorithm which will take as input a basis (not orthogonal) of the integer lattice Λ and return a list of at most 2^n shortest vectors of this lattice. Since this lattice is an integer lattice, then the *LLL* algorithm will return a shortest vector of the lattice that we call v. Even if an integer

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

lattice is also an orthogonal lattice, it would be interesting to use a non-orthogonal basis of the lattice. Indeed, by applying the LLL algorithm to an orthogonal basis, we obtain the same basis. Consequently, the vectors obtained will not necessarily be the short vectors of the lattice. Therefore, we will bring out all the possible combinations between the components of the vector v and its opposite $-v$ (this by keeping the position of each component used). The description of our algorithm is given as follows.

4.1. *Description of the algorithm*

Given an orthogonal integer lattice Λ, this algorithm takes as input the (non-orthogonal) basis $B = (b_1, b_2, \ldots, b_n)$ of the lattice (where n is the dimension of Λ) and returns a list L of at least n and at most 2^n short vectors of the lattice Λ and the number of collision C as follows: We start by executing the LLL algorithm to the basis B which allows us to obtain a short vector of the lattice which we denote by u. Subsequently, we will use this vector u and its opposite $v = -u$ to build a list L. To achieve this, we will build a $2^n \times n$ matrix K using an iterative function $Vect$ and an additional $2^{n-1} \times (n-1)$ matrix P. The 2^n rows of our constructed matrix K will be short vectors of the lattice. Now, we will consider the list H whose elements are rows of K. A final list L consisting of short vectors will then be constructed form K, making sure that an element appears only once. The number of collisions is be the number of repetitions of the vectors in the list H.

At the end of the algorithm, we will have the list L which will be made up of at least n and at most 2^n short vectors of the lattice, and the number of collisions C.

Remark 4.4. We will call the number of collisions that we will denote by C, the total number of repetitions of the vectors that we will have in the auxiliary list H which will make it possible to obtain the list L of short vectors. Thus, if the number of collisions is large, then the size of the list L is small. Indeed, the total number of vectors of the list L will be equal to $2^n - C$.

The algorithm below illustrates the above description. For correctness, a Maple computer software implementation of the algorithm has been done.

4.2. *Complexity analysis*

About the complexity of our algorithm, we have

Line 1 has 2 elementary operations. Indeed, we have only 2 assignments in this step; line 2 is carried out in polynomial time with complexity $O(n)$ arithmetic operations. Indeed, algorithm LLL runs in $O(n)$ arithmetic operations. Line 3 has 2 elementary operations (assignments). Line 4 has $2(n-1)$ arithmetic operations. Indeed, in this line we have 2 affectations inside the loop for which goes from 1 to $n-1$; from line 5 to line 7, we also have $2(n-1)$ elementary operations. Indeed,

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

we have 2 assignments inside the loop for which goes from 1 to $n-1$; Line 8 has $(n-1)2^{n-1}$ arithmetic operations. Indeed, we use a recursive algorithm that uses two loops "for", which one goes from 1 to 2^{n-1} and the other from 1 to $n-1$; Line 9 has 3 elementary operations (assignments); from line 10 to line 14, we have two loops and the first goes from 1 to 2^{n-1}, and inside this one we have another loop for which it goes from 1 to $n-1$. Thus, we will have $2^{n-1}(n-1)$ operations from line 10 to line 14. In the same way, we will have $2^{n-1}(n-1)$ operations from line 15 to line 19; from line 20 to line 22, we have 2^{n-1} because we have only one operation inside the loop for which it goes from 1 to 2^{n-1}. In the same way, we will have 2^{n-1} operations from line 23 to line 26; line 27 has 2^n+1 operations because we have 1 elementary operation (assignment) and 2^n assignments to build matrix K; from line 29 to line 34, we have 2 operations (assignment and comparison) which will be automatically executed inside the loop for which it goes from 1 to 2^n-1. Thus we will have $2\times(2^n-1)=2^{n+1}-2$ operation from line 29 to line 34.

So we will have $2^{n+1}-2+2^n+1+2^{n-1}+2^{n-1}+(n-1)2^{n-1}+(n-1)2^{n-1}+(n-1)2^{n-1}+2(n-1)+2+n+2$ arithmetic operations; this means that we have $2^{n+1}-2+2^n+1+2^n+(n-1)2^n+(n-1)2^{n-1}+2(n-1)+n+4$ arithmetic operations; thus, we have $2^{n+1}+2^{n+1}+(n-1)2^n+(n-1)2^{n-1}+2(n-1)+n+3$; since $\frac{2^{n+2}+(n-1)2^n+(n-1)2^{n-1}+2(n-1)+n+3}{n2^n}\to\ cte$ when $n\to+\infty$, then the complexity of algorithm is $O(n2^n)$.

Therefore, the complexity of our algorithm is $O(n2^n)$ arithmetic operations.

Example 4.5. Let $B:=\begin{pmatrix}3&3&-3\\1&3&1\\1&4&-2\end{pmatrix}$ be a basis of a lattice $\Lambda(B)\subset\mathbb{Z}^3$; we have, $G:=LLL(B)=\begin{pmatrix}0&0&3\\2&2&1\\-1&3&1\end{pmatrix}$; thus $u=(0,2,-1)$, $v=(0,-2,1)$ and $n=3$; We have $n\neq1$, then $p=(0,2)$ and $q=(0,-2)$; then $P:=\mathrm{Vect}(p=(0,2),q=(0,-2),n=2)$; thus $n=2\neq0$, this means that we have $P:=\mathrm{Vect}(p=(0),q=(0),n=1)$; therefore, $l=2^2=4$ and $t=2^{2-1}=2$; thus for $i=1,2$ and $j=1$ we have: $P[1,1]=0$ and $P[2,1]=0$ for $i=3,4$ and $j=1$ we have: $P[3,1]=0$ and $P[4,1]=0$. Thus P is the form $K:=\begin{pmatrix}0&0\\0&0\\0&0\\0&0\end{pmatrix}$ now we will complete the second column as folows: for $i=1,2$ and $j=2$ we have: $P[1,2]=P[2,2]=u_2=2$; for $i=3,3$ $j=2$ we have: $P[3,2]=P[4,2]=v_2=-2$; and then, we have $P:=\begin{pmatrix}0&2\\0&2\\0&-2\\0&-2\end{pmatrix}$, now $l=2^3=8$ and $t=2^2=4$; thus for $i=1,\dots,4$ and $j=1,2$, we have: $K[1,1]=P[1,1]=0$; $K[1,2]=P[1,2]=2$; $K[2,1]=P[2,1]=0$; $K[2,2]=P[2,2]=2$; $K[3,1]=P[3,1]=0$; $K[3,2]=P[3,2]=-2$; $K[4,1]=P[4,1]=0$ and $K[4,2]=P[4,2]=-2$; for $i=5,\dots,8$ and $j=1,2$, we also

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

Algorithm 3. Orthogonal integer sieve

Require: The basis B of a lattice Λ and its dimension $n \geq 2$.

Ensure: A list L of *short* vectors v with $\|v\| = \lambda_1(\Lambda(B))$ and the number of collisions C.

$L := \{\}$; $C := 0$; "We initialize a empty list L and the number of collision C"
$G := LLL(B)$; "$LLL(B)$ takes as input the basis B and returns its reduced basis"
$u := G[,1]$; $v := -u$; " u is the $1st$ column of matrix G and v is its opposite"
$p := (0, \ldots, 0)$; $q := (0, \ldots, 0)$ "$(n-1)$ times"
for $i = 1, \ldots, n-1$ **do**
  $p_i := u_i$; $q_i := v_i$;
end for
$P := Vect(p, q, n-1)$; "The function Vect takes as input the vectors p and q, and builds a $2^{n-1} \times (n-1)$ matrix P"
$K := matrix(0,\ nrow = 2^n,\ ncol = n)$; $l := 2^n$; $t := 2^{n-1}$; "We initialize the $2^n \times n$ matrix with 0 everywhere"
for $i = 1, \ldots, t$ **do**
  for $j = 1, \ldots, n-1$ **do**
    $K[i,j] := P[i,j]$;
  end for
end for
for $i = t+1, \ldots, l$ **do**
  for $j = 1, \ldots, n-1$ **do**
    $K[i,j] := P[i-t, j]$;
  end for
end for
for $i = 1, \ldots, t$ **do**
  $K[i,n] := u_n$; "we update the 2^{n-1} first components of column n of the matrix K"
end for
for $i = t+1, \ldots, l$ **do**
  $K[i,n] := v_n$; "we update the last 2^{n-1} components of column n of matrix K"
end for
end if
$H := (K[1,], \ldots, K[2^n,])$; $L := L \cup \{H[1]\}$); "$K[i,]$ is line number i of the matrix K"
for $i = 2, \ldots, 2^n$ **do**
  if $H[i] \notin L$ then **then**
    $L := L \cup \{H[i]\}$); "we remove all copies from the list"
  else
    $C := C + 1$;
  end if
end for
return (The list L of *shortest* vectors v with $\|v\| = \lambda_1(\Lambda(B))$ and C);

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.

have: $K[5,1] = P[1,1] = 0$; $K[5,2] = P[1,2] = 2$; $K[6,1] = P[2,1] = 0$; $K[6,2] = P[2,2] = 2$; $K[7,1] = P[3,1] = 0$; $K[7,2] = P[3,2] = -2$; $K[8,1] = P[4,1] = 4$ and $K[8,2] = P[4,2] = -2$; Thus K is the form $K := \begin{pmatrix} 0 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & -2 & 0 \\ 0 & -2 & 0 \\ 0 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & -2 & 0 \\ 0 & -2 & 0 \end{pmatrix}$, now we will complete the last column as follows: for $i = 1, \ldots, 4$ and $j = 3$, we have $K[1,3] = K[2,3] = K[3,3] = K[4,3] = u_3 = -1$; for $i = 5, \ldots, 8$ and $j = 3$, we have $K[5,3] = K[6,3] = K[7,3] = K[8,3] = v_3 = 1$; thus, we have $K = \begin{pmatrix} 0 & 2 & -1 \\ 0 & 2 & -1 \\ 0 & -2 & -1 \\ 0 & -2 & -1 \\ 0 & 2 & 1 \\ 0 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & -2 & 1 \end{pmatrix}$. Thus, $H = \{(0,2,-1), (0,2,-1), (0,-2,-1), (0,-2,-1), (0,2,1), (0,2,1), (0,-2,1), (0,-2,1)\}$. Therefore, $L = \{0,2,-1), (0,-2,-1), (0,2,1), (0,-2,1)\}$ and $C = 4$.

5. Conclusion

In this work, we talked about the notions of orthogonal lattices, integer lattices, gave some properties of this family of lattice. We also recalled the relationship between orthogonal and integer lattices. All this allowed us to construct an enumeration algorithm for integer lattice $\mathbb{Z}^n$ to provide a full list of its shortest vectors. This algorithm runs in space $O(2n)$. We also constructed an algorithm which give at least n and at most 2^n short vectors of a general orthogonal integer lattice $\Lambda \subset \mathbb{Z}^n$. This algorithm runs in time $O(n2^n)$ and can be polynomial in space. We have successfully implemented these algorithms in the Maple computer software 18.0. Our future work will consist in giving an algorithm which will give a list of short vectors in general case of any orthogonal lattice.

Acknowledgments

Authors are supported by the Simons Foundation through the project PREMA, Subsaharan Africa. The second author acknowledges the support of TWAS UNESCO under the Grant 20-063 RG/MATHS/AF/AC-I.

References

[1] M. Ajtai, R. Kumar and D. Sivakumar, A sieve algorithm for the shortest lattice vector problem, in *Proc. 33rd Annual ACM Symp. Theory of Computing*, eds. J. S. Vitter, P. G. Spirakis and M. Yannakakis, July 6–8 Heraklion, Crete, Greece (ACM, 2001), pp. 601–610.

[2] J. Blömer and J. Seifert, On the complexity of computing short linearly independent vectors and short bases in a lattice, in *Proc. 31 Annual ACM Symp. Theory of Computing*, eds. J. S. Vitter, L. L. Larmore and F. T. Leighton, May 1–4, Atlanta, Georgia, USA (ACM, 1999), pp. 711–720.

[3] K. Chandrasekaran, V. Gandikota and E. Grigorescu, Deciding orthogonality in construction-a lattices, *SIAM J. Discret. Math.* **31**(2) (2017) 1244–1262.

[4] J. Chen, D. Stehlé and G. Villard, Computing an lll-reduced basis of the orthogonal latice, in *Proc. 2018 ACM on Int. Symp. Symbolic and Algebraic Computation*, eds. M. Kauers, A. Ovchinnikov and É. Schost, ISSAC 2018, New York, USA, July 16–19(ACM, 2018), pp. 127–133.

[5] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, Grundlehren der mathematischen Wissenschaften, Vol. 290 (Springer, 1988).

[6] D. Dachman-Soled, L. Ducas, H. Gong and M. Rossi, LWE with side information: Attacks and concrete security estimation, in *Advances in Cryptology — CRYPTO 2020 — 40th Annual Int. Cryptology Conference, CRYPTO 2020*, eds. D. Micciancio and T. Ristenpart, Proceedings, Part II, Lecture Notes in Computer Science, Vol. 12171, Santa Barbara, CA, USA, August 17–21, (Springer, 2020), pp. 329–358.

[7] T. Debris-Alazard, L. Ducas and W. P. J. van Woerden, An algorithmic reduction theory for binary codes: LLL and more, *IEEE Trans. Inf. Theory* **68**(5) (2022) 3426–3444.

[8] L. Ducas, Shortest vector from lattice sieving: A few dimensions for free, in *Advances in Cryptology —EUROCRYPT 2018 — 37th Annual Int. Conf. Theory and Applications of Cryptographic Techniques*, eds. J. B. Nielsen and V. Rijmen, Tel Aviv, Israel, April 29–May 3, 2018 Proceedings, Part I, Lecture Notes in Computer Science, Vol. 10820 (Springer, 2018), pp. 125–145.

[9] L. Ducas, M. Plançon and B. Wesolowski, On the shortness of vectors to be found by the ideal-svp quantum algorithm, *IACR Cryptol. ePrint Arch.* **11692** (2019) 322–351.

[10] O. Goldreich, S. Goldwasser and S. Halevi, Public-key cryptosystems from lattice reduction problems, in *Advances in Cryptology - CRYPTO '97, 17th Annual Int. Cryptology Conference*, Santa Barbara, California, USA, August 17–21, Proceedings, Lecture Notes in Computer Science, Vol. 1294 (Springer, 1997), pp. 112–131.

[11] R. Kannan, Minkowski's convex body theorem and integer programming, *Math. Oper. Res.* **12**(3) (1987) 415–440.

[12] D. Micciancio and P. Voulgaris, Faster exponential time algorithms for the shortest vector problem, in *Proc. 21 Annual ACM-SIAM Symp. Discrete Algorithms, SODA 2010*, ed. M. Charikar, Austin, Texas, USA, January 17–19 (SIAM, 2010), pp. 1468–1480.

[13] P. Q. Nguyen and T. Vidick, Sieve algorithms for the shortest vector problem are practical, *J. Math. Cryptol.* **2**(2) (2008) 181–207.

[14] M. Pohst, On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications, *SIGSAM Bull.* **15**(1) (1981) 37–44.

[15] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* **26**(5) (1997) 1484–1509.

[16] D. Stehlé, Algorithmique de la réduction de réseaux et application à la recherche de pires cas pour l'arrondi defonctions mathématiques, Ph.D. thesis, Henri Poincaré University, Nancy, France (2005).

[17] P. Voulgaris, Algorithms for the closest and shortest vector problems on general lattices, Ph.D. thesis, University of California, San Diego, USA (2011).

Discrete Math. Algorithm. Appl. Downloaded from www.worldscientific.com
by ZHEJIANG NORMAL UNIVERSITY on 10/06/22. Re-use and distribution is strictly not permitted, except for Open Access articles.