Nagyteljesítményű számítási algoritmusok tervezése és megvalósítása vezeték nélküli MIMO rendszerek számára

A doktori disszertáció tézisei

Józsa Csaba Máté, M.Sc.

Témavezetők

Kolumbán Géza, D.Sc.
A Magyar Tudományos Akadémia doktora

Szolgay Péter, D.Sc.
A Magyar Tudományos Akadémia doktora

Pázmány Péter Katolikus Egyetem
Információs Technológiai és Bionikai Kar
Multidiszciplináris Műszaki és Természettudományi Doktori Iskola

Budapest, 2015

1. Bevezetés

A vezeték nélküli kommunikáció fő hajtóerői a magasabb adatátviteli sebesség, nagyobb hálózati kapacitás és a fokozott megbízhatóság. A kommunikációs rendszerek korlátozó tényezői a berendezések költsége, a rádióhullámok terjedése és a rendelkezésre álló szűkős frekvencia spektrum. A nagyobb adatátviteli sebesség iránti igény arra ösztönözte a kutatókat, hogy új módszerek és algoritmusok segítségével érjék el az egy adó és egy vevő antennával rendelkező vezeték nélküli rendszerek kapacitásának elméleti határát. Információelméleti kutatások [8] azt mutatják, hogy jelentős javulás érhető el az adatátviteli sebesség és a megbízhatóság terén, ha több antenna áll rendelkezésre úgy az adó mint a vevő oldalán. Ezeket a rendszereket MIMO rendszereknek nevezzük [9]. A MIMO rendszerek kulcsfontosságú képessége az, hogy a többutas terjedést, ami hagyományosan a vezeték nélküli kommunikáció egyik fő problémaforrása, a felhasználó előnyére fordítja, így az adott sávszélesség mellett, nagyságrendekkel javítani lehet a vezeték nélküli rendszerek teljesítményét. MIMO rendszerekben a hiba valószínűsége akkor minimalizálható, ha ugyanazon adatfolyam különböző reprezentációi kerülnek továbbításra párhuzamos csatornákon, azaz egy tér-időbeli redundancia kerül bevezetésre. A MIMO csatorna kapacitása úgy növelhető, hogy kihasználva a többutas terjedést, független adatfolyamok egyidejűleg és ugyanabban a frekvenciasávban kerülnek továbbításra.

A különböző vevő struktúrákban használt MIMO detektorok komplexitása függ: az antennák számától, a moduláció rendjétől, kódolástól, stb. A hagyományos additív fehér Gauss-zajjal [Additive White Gaussian Noise (AWGN)] terhelt csatornák esetén az optimális bithibaarány [Bit Error Rate (BER)] a legnagyobb valószínűség [Maximum Likelihood (ML)] elvén történő detekcióval érhető el. Az ML detekció kimerítő keresés alapú komplexitása exponenciálisan nő a jelkészlet elemeinek és az antennák számával, így ez a megvalósítás nem kivitelezhető valódi rendszerekben. Egy ígéretes megvalósítás a szférikus detektor [Sphere Detector (SD)], mert az jelentősen képes csökkenti a keresési teret az optimális bithibaarány megtartása mellett. A keresési térben csak olyan rácspontok kerülnek megvizsgálásra, amelyek egy meghatározott hipergömb által befoglalt térben vannak. A hipergömb középpontját a vett szimbólum vektora határozza meg, és sugara a csatornazaj mértékétől függhet. A keresési tér csökkenése nem eredményez szuboptimális detekciót, mert a középponthoz eső legközelebbi rácspont ugyanaz lesz, akár a teljes ke-

resési teret, akár a hipergömb által befoglalt teret vizsgáljuk. Az SD algoritmus hátrányai a következők: (i) komplexitása továbbra is exponenciális nő az antennák számának vagy a moduláció rendjének növelésével, (ii) egy mélységi keresésé transzformálja a MIMO detekciós problémát, mely jellegét tekintve erősen szekvenciális, és (iii) mivel a különböző mélységi keresések során különböző útvonalak kerülnek bejárásra a feldolgozási idő és az algoritmus komplexitása változó lesz.

Többfelhasználós kommunikációs rendszerek esetén, ha a bázisállomás több antennával rendelkezik a térbeli diverzitás akkor is kihasználható, ha a mobil állomásoknak csak egy antennájuk van. Mivel a mobil állomások számára nem minden kommunikációs csatorna ismert, az összes jelfeldolgozási feladat a bázisállomásra hárul, ilyen például a szimbólumok előkódolása, mely a felhasználók közötti interferenciát szünteti meg. Több kutatás is bizonyította, hogy a rácsredukcióval támogatott lineáris és nemlineáris előkódolás jelentősen csökkenti a bithibaarányt a rácsredukcióval nem támogatott előkódoláshoz képest. Továbbá számos kutatás megmutatta, hogy a rácsredukció a lineáris és nemlineáris detekció teljesítményét is tudja fokozni. A rácsredukció során létrejövő új bázis kondíciószáma és ortogonalitási hiba mérséklése lehetővé teszi, hogy a kevésbé komplex lineáris detekciós algoritmusok is teljes rendű diverzitást érjenek el. A rácsredukciós algoritmusok komplexitása a bázis méretétől függ. Mivel a rácsredukciót a MIMO rendszerek csatorna mátrixán kell végrehajtani, a rácsredukció komplexitása, és ezzel együtt a feldolgozási idő, kritikussá válhat nagy MIMO rendszerek esetén.

Nagy MIMO rendszerek esetén az optimális detekció vagy előkódolás számítási komplexitása is rendkívül nagyra nőhet. Ugyanakkor a különböző modulációs eljárások, csatorna modellek, előkódolási, detektálási és dekódolási eljárások kutatása során előfordulhat, hogy az elméleti teljesítményt csak szimulációk által lehet meghatározni. Egy másik megközelítés, hogy előfeldolgozó algoritmusok segítségével javítjuk az adott probléma paramétereit, és ezáltal, a kevésbé komplex jelfeldolgozó algoritmusok is (például a lineáris detekció, előkódolás) jó teljesítményt érnek el. Ebben az esetben, a feldolgozási időt az előfeldolgozási algoritmusok komplexitása határozza meg. A fentiek konklúziója, hogy a megnövelt spektrális hatékonyság MIMO rendszerek esetén az összetettebb hardverelemekkel és a magasabb komplexitású jelfeldolgozó algoritmusokkal érhető el, bár ezen algoritmusok szekvenciális jellege nem teszi lehetővé a párhuzamos architektúrák hatékony használatát.

A számítási architektúrák és programozási modellek terén elért jelentős fejlődés miatt elérhetővé váltak a viszonylag olcsó, nagy teljesítményű, masszívan párhuzamos architektúrák [Massively Parallel Architectures (MPA-k)]: az általános célú grafikus processzor egységek (GP-GPU-k) és a programozható kapu mátrixok [Field Programmable Gate Arrays (FPGA-k)]. Számos tudományterületen végzett kutatás bebizonyította [10], [11], [12], [13], hogy a GP-GPU alkalmazása jelentős rendszerszintű teljesítmény javulást eredményez. Mivel a modern okostelefonok rendelkeznek GP-GPU-val, és elérhetővé váltak a nagy teljesítményű GP-GPU alapú klaszterek, már nem jelent problémát ezen eszközök alkalmazása a nagy komplexitású problémák megoldásában. Ezekkel a hatékony MPA-kal lehetővé válik a viszonylag magas és változó számítási komplexitású algoritmusok valós idejű végrehajtása, továbbá a hosszas szimulációk feldolgozási ideje is csökkenthető. Már trendszerű az MPA-k használata számos, bonyolult jelfeldolgozó feladat esetén. Számításigényes jelfeldolgozó algoritmusok, mint például a detekció [10], [14], a dekódolás [11], [12] és az előkódolás [13] hatékonyan leképezhetők a GP-GPU architektúrákra.

Látható, hogy a probléma megoldásához használt architektúra alapvetően meghatározza a feldolgozási időt és az eredmények minőségét. Mivel a meglévő algoritmusok többnyire szekvenciálisak, szükségessé válik a létező algoritmusok alapvető újratervezése, hogy az új MPA-kat alkalmazni lehessen. Ezen hatékony eszközök felhasználásával új perspektívák nyílnak meg. Ebben a dolgozatban célom, hogy ezen modern, sokprocesszoros, párhuzamos eszközök számára olyan hatékony és masszívan párhuzamos algoritmusokat tervezzek, melyek (i) képesek a nagy bonyolultságú ML detektálási probléma valósidejű megoldására, illetve (ii) melyek alkalmazása a probléma előfeldolgozásaként, mint például a rácsredukciós (Lattice Reduction (LR)) algoritmusok, lehetővé teszik kis komplexitású jelfeldolgozó algoritmusok használatát úgy, hogy a rendszer teljesítménye optimális közeli legyen.

2. A kutatásban alkalmazott módszerek és eszközök

Kutatásom célja, hogy a vezeték nélküli kommunikáció területén, magas komplexitású jelfeldolgozási problémákat oldjak meg modern GP-GPU és sokmagos CPU párhuzamos architektúrákon. A fő kihívást az jelentette, hogy a magas komplexitású, szekvenciális problémákat, hogyan lehet különféle létező és újonnan kidolgozott matematikai és algoritmikus transzformációk alkalmazásával alkalmassá tenni a párhuzamos architektúrák számára.

Kutatásom első részében a MIMO rendszerek optimális ML detekcióját tanulmányozom. Az ML detektor komplexitása exponenciálisan növekszik az antennák számával és a moduláció rendjével. A komplexitás jelentős csökkentése érdekében, a szférikus detekciós módszert [15] használtam. A szférikus detektor alapvető célja a keresési tér korlátozása egy hipergömbön belüli rácspont halmazra, melynek középpontját a vett szimbólum vektor jelöli ki. A keresési tér csökkenése nem eredményez szuboptimális detekciót, mert a középponthoz eső legközelebbi rácspont ugyanaz lesz, akár a teljes keresési teret, akár a hipergömb által befoglalt teret vizsgáljuk.

A különböző szimbólum vektorok detekciója során az optimális megoldáshoz különböző keresési útvonalak vezetnek, melyekhez változó feldolgozási idő tartozik. A változó komplexitás kedvezőtlen hatásainak mérséklése érdekében (i) egy oszlop normán alapuló rendezési módszert alkalmaztam és kidolgoztam (ii) egy dinamikus terheléselosztást ütemező algoritmust. Korábban bizonyításra került, hogy a csatornamátrix oszlop normáin alapuló szimbólumok detekciójának sorrendje jelentősen szűkítheti a bejárt útvonalakat [16]. Ha magasabb poszt-detekciós jelzaj viszonnyal rendelkező szimbólumok detekciója a keresési fa felső szintjein valósul meg, akkor nagy lesz a valószínűsége annak, hogy a választott útvonal az optimális útvonal. Következésképpen, nem vezet jelentős visszalépéshez egy nem optimális döntés, ha az a keresési fa egy alacsonyabb szintjén valósul meg.

A szimbólum vektorok változó detekciós ideje a CUDA kernelek szálblokkjai (thread blocks) futási idejének a kiegyensúlyozatlanságához vezethet. Amíg egy kernel szálblokkjai nem fejezik be futásukat, a kernelhez rendelt erőforrások nem szabadulnak fel. A hosszú ideig tartó erőforrás birtoklás gátolja a kernelek párhuzamos végrehajtását a különböző CUDA folyamokon (streams). A dinami-

kus terhelésmegosztással mérsékelhető a szálblokkok futási idejének a kiegyensúlyozatlansága, ezáltal a kernelek párhuzamos végrehajtása hatékonyabb lesz, ami csökkenti a teljes futási időt és elfedi az egyes szimbólumok változó detekciós idejét.

Kutatásom második részének középpontjában az LR módszer [17] áll, mely egy széleskörűen alkalmazható matematikai eszköz. Az LR célja, hogy egy adott pontrács számára, az euklideszi norma értelmében, ortogonálisabb és rövidebb vektorokból álló bázist találjon, mint az eredeti bázist alkotó vektorok. Az LR javítja a bázismátrix kondíciószámát, az ortogonalitási hibametrikáját, illetve a Seysen metrikát.

Az irodalomban számos LR algoritmus létezik, melyek leginkább a számítási komplexitásban és ezzel együtt a létrehozott bázis "jóságában" különböznek. A legelterjedtebb LR algoritmust Lenstra-Lenstra-Lovász (LLL) [18] alkotta meg, mely népszerűségét annak köszönhette, hogy ez volt az első polinomidejű LR algoritmus. Széleskörű alkalmazhatósága és számos előnyös tulajdonsága miatt, kutatásom az LLL algoritmus továbbfejlesztésére, illetve MPA-k számára alkalmassá tételére összpontosult.

Korábban bebizonyították, hogy a lineáris és nemlineáris detektorok is teljes rendű diverzitást érhetnek el, ha a rácsredukciós technikákkal együtt alkalmazzák [19]. A többfelhasználós kommunikációs rendszerekben a bázisállomás megszüntetheti a felhasználók közötti interferenciát előkódolási módszerek alkalmazásával. Megmutatták, hogy a lineáris előkódolási módszerek, mint a Zero-Forcing (ZF) előkódolás, és a nemlineáris előkódolási módszerek, mint például a Tomlinson-Harashima előkódolás és a vektor perturbációs technikák, jobban teljesítenek, ha a mátrix jól kondicionált [20], azaz a mátrix kondíciószáma alacsony. Így teljes rendű diverzitás érhető el nagy rendszerek esetén is.

A fent említett, számítási szempontból kihívást jelentő jelfeldolgozó feladatokhoz használt eszközök: a modern többmagos CPU-k, pl. Intel Core i7-3820, Intel Xeon X5680, Intel Xeon E5-2650 v3, és a masszívan párhuzamos architektúrák, pl. Nvidia GeForce GTX 690, Nvidia Tesla C2075 és K20 GP-GPU-k. Számos párhuzamos programozási modellt is alkalmaztam, hogy a feltárt többszintű párhuzamosság a megfelelő architektúrára kerüljön leképezésre. A durva szemcsézettségű (coarse-grained) osztott memórián alapuló párhuzamossághoz, a többszálú programozást lehetővé tevő OpenMP-t használtam. A finom szemcsézettségű (fine-grained) párhuzamosságot a CUDA modell alapján valósítottam meg.

3. Tudományos eredmények

I. Téziscsoport - A legnagyobb valószínűség elvén alapuló új párhuzamos működési elvű szférikus detektorok tervezése és azok sokprocesszoros architektúrákra való hatékony leképezése.

(Kapcsolódó publikációk: [1], [3].)

Tézis I.a.

A MIMO rendszerek számára kidolgoztam egy új Párhuzamos Szférikus Detekciós (PSD) algoritmust, mely a legnagyobb valószínűség elvén alapul, és optimális bithibaarányt valósít meg additív fehér Gauss-zajjal terhelt csatorna esetén. A PSD algoritmus magas fokú párhuzamosságát biztosító összetevők: a mélységi és szélességi keresések hatékony kombinációján alapuló új, hibrid fakeresési módszer, továbbá a közbülső szinteken megvalósuló útmetrika szerinti rendezést a párhuzamos rendezési hálózatok biztosítják. Megmutattam, hogy a PSD algoritmus lehetővé teszi a hatékony munkamegosztást egy erősen többszálú környezetben úgy, hogy az egy szál által bejárt csúcsok száma $88\% - 96\%$*-al csökken, továbbá az átlagos detekciós teljesítmény,* 4×4 *MIMO rendszerek esetén,* $2 - 50$*-szeres gyorsulást érhet el a különböző jel-zaj viszonyok esetén a szekvenciális algoritmussal szemben.*

A valós jelek felett értelmezett MIMO rendszer modellje leírható egy lineáris egyenlet segítségével

$$\mathbf{y} = \mathbf{H}\mathbf{s}_t + \mathbf{v}$$

ahol $\mathbf{y} \in \mathbb{R}^M$ a vett szimbólum vektor, $\mathbf{v} \in \mathbb{R}^M$ az additív csatornazaj, $\mathbf{s}_t \in \Omega^N$ a küldött szimbólum vektor, Ω a szimbólum készlet, és a továbbított szimbólumok szuperpozícióját a $\mathbf{H} \in \mathbb{R}^{M \times N}$ csatorna mátrix határozza meg. Az optimális ML detektor additív fehér Gauss-zaj esetén a következő egyenlettel definiálható

$$\hat{\mathbf{s}}_{ML} = \arg\min_{\mathbf{s} \in \Omega^N} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2.$$

A küldött szimbólum vektor ML becslése az egész számokon értelmezett legkisebb négyzetek problémájának megoldásával tehető meg, amely ek-

vivalens azzal, hogy egy $\mathbf{\Lambda} = \{\mathbf{Hs} | \mathbf{s} \in \Omega^N\}$ pontrács pontjai között kell megtalálni azt a pontot, amely legközelebb esik egy adott $\mathbf{y}$ vektorhoz.

A feltételekhez nem kötött legkisebb négyzetek megoldása $\hat{\mathbf{s}} = \mathbf{H}^{\dagger}\mathbf{y}$, ahol $\mathbf{H}^{\dagger}$ a Moore-Penrose pszeudoinverzet jelöli. A $\mathbf{H} = \mathbf{QR}$ a csatornamátrix QR faktorizációjával, az ML detekció felírható mint

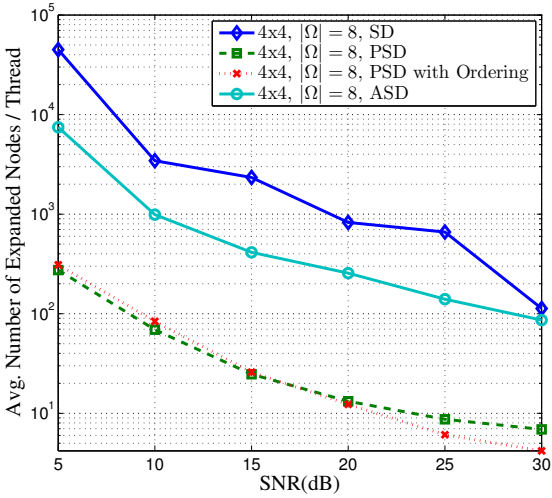
$$\hat{\mathbf{s}}_{ML} = \arg\min_{\mathbf{s}\in\Omega^N} \|\mathbf{R}(\mathbf{s} - \hat{\mathbf{s}})\|^2.$$

Egy $\mathbf{y}$ középpontú, d sugarú $S(\mathbf{y}, d)$ hipergömb tartalmazza a $\mathbf{Hs}$ rácspontot, ha az ekvivalens ML detektor kiértékelő függvényére teljesül az $\|\mathbf{R}(\mathbf{s} - \hat{\mathbf{s}})\|^2 \leqslant d^2$ egyenlőtlenség, azaz ha

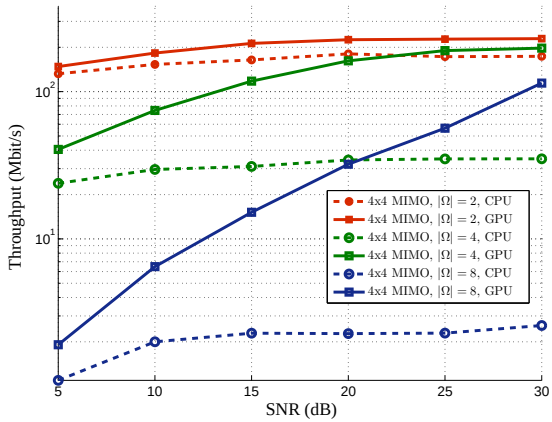
$$\left\| \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1N} \\ 0 & r_{22} & \cdots & r_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{NN} \end{pmatrix} \begin{pmatrix} s_1 - \hat{s}_1 \\ s_2 - \hat{s}_2 \\ \vdots \\ s_N - \hat{s}_N \end{pmatrix} \right\|^2 \leqslant d^2.$$

Az N-ik dimenzióval kezdve, a szimbólum készlet egyik eleme behelyettesítésre kerül a szimbólum vektor soron következő hiányzó szimbólumának helyére, ezt követi az egyenlőtlenségi feltétel teljesülésének ellenőrzése. Mivel a keresési folyamat egy mélységi bejárásnak felel meg, mely jellegét tekintve erősen szekvenciális, a probléma nem oldható meg hatékonyan egy többszálas környezetben.

Az általam kidolgozott PSD algoritmus teljesen kiküszöböli az SD algoritmus szekvenciális részeit. A PSD algoritmus egy új fakeresési algoritmust valósít meg, ahol a párhuzamosságot egy hibrid, szélességi és mélységi fakeresés hatékony kombinációja biztosítja. A hibrid keresés eredménye, hogy a fa lvl_x paraméterekkel jelölt szintjein lévő csúcsok lesznek csak bejárva. Minden csúcs leírható egy részleges vagy teljes szimbólum vektorral. A megjelölt szinteken egyszerre exp_{lvl_x} darab részleges szimbólum vektor kerül kifejtésre. Egy részleges szimbólum vektor kifejtése során $(lvl_{x-1} - lvl_x)$ darab új szimbólummal bővül az eredeti vektor. Ha egyidejűleg $exp_{lvl_{x-1}}$ részleges szimbólum vektor kerül kifejtésre, akkor a következő szinten $eval_{lvl_x} = exp_{lvl_{x-1}} \cdot |\Omega|^{(lvl_{x-1} - lvl_x)}$ darab új szimbólum vektor keletkezik. Ezen a ponton megvalósítom a hibrid keresést, mivel az exp_{lvl_x} paraméterekkel a szélességi keresés, míg a lvl_x paraméterekkel a mélységi keresés kiterjedése szabályozható. Mivel több új (részleges) szimbólum vektor jön létre a kifejtési szakasz során, ezért lehetővé válik az útmetrikák párhuzamos kiértékelése, így az MPA



1. ábra. Összehasonlítása a szálanként átlagban bejárt csúcsok számának egy $|\Omega| = 8$ elemű jelkészletű 4×4 MIMO rendszerben, a szekvenciális, párhuzamos és automatikus szférikus detektor esetén.



2. ábra. A PSD algoritmus GP-GPU megvalósítása és egy többmagú CPU szálanként futtatott szekvenciális SD algoritmus által elért átlagos detekciós teljesítmény összehasonlítása.

erőforrásai is hatékonyan kihasználhatók.

Az 1. ábra mutatja a különböző MIMO konfigurációk átlagban kifejtett csúcsainak szálankénti számát. A PSD algoritmust futtató szálak teljes terhelése 5 dB-es jel-zaj viszony esetében 96%-kal, 20 dB-es jel-zaj viszony esetében ugyanez a terhelés 95%-kal csökken.

A 2. ábrán a PSD algoritmus egy GTX690 GP-GPU megvalósítása és egy többmagú Intel Xeon E5-2650 CPU szálanként futtatott szekvenciális SD algoritmus által elért átlagos detekciós teljesítménye kerül összehasonlításra. Megfigyelhető, hogy 30 dB-es jel-zaj viszony esetén a 4×4 MIMO rendszer átlagos detekciós teljesítménye, $|\Omega| = 4$ elemű jelkészlet esetén 6-szor gyorsabb, $|\Omega| = 8$ elemű jelkészlet esetén 50-szer gyorsabb a GP-GPU architektúrán futtatva.

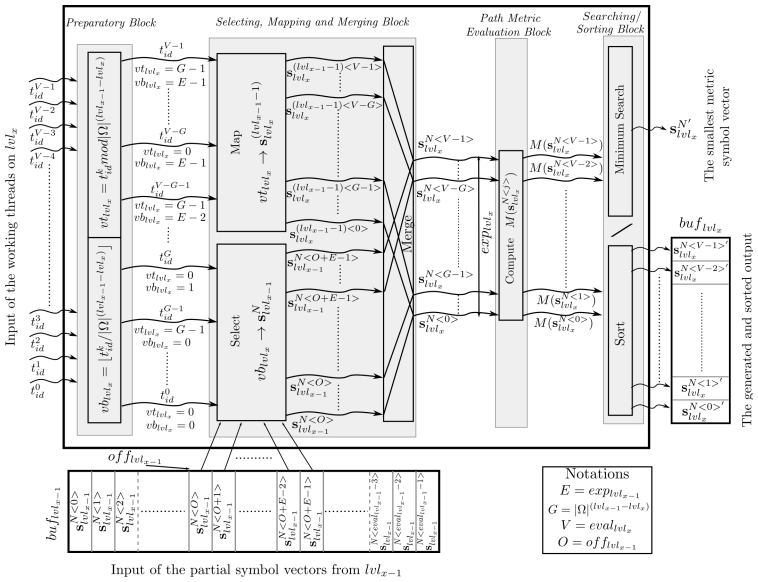
Tézis I.b.

A rendelkezésre álló párhuzamosság függvényében dinamikus, masszívan párhuzamos építőelemeket definiáltam a PSD algoritmus csúcs kifejtő és kiértékelő folyamatához. Az építőelemek alapján definiáltam azon paramétereket, melyek meghatározzák a párhuzamosság kiterjedését és az algoritmus memória igényét. Megmutattam, hogy a GP-GPU implementáció átlagos detekció sebessége felülmúlja valamennyi létező optimális ML detektort és számos GP-GPU, ASIC, DSP és FPGA architektúrán megvalósított szuboptimális detektort.

A detekciós folyamat során a legsűrűbben használt műveletek a csúcsok kifejtése és kiértékelése. Ezért megterveztem a párhuzamos csúcs kifejtő és kiértékelő folyamatot [Expansion and Evaluation Pipeline (EEP)], mely feloldja a párhuzamos implementációt akadályozó szűk keresztmetszeteket. Az EEP építőelemei a 3. ábrán láthatók: (i) az előkészítő blokk, (ii) a kiválasztás, leképezés és egyesítés blokk, (iii) az útmetrika kiértékelő blokk, és (iv) a keresés vagy rendezés blokk.

Az *előkészítő blokkban* a rendelkezésre álló szálak vagy processzáló egységek t_{id}^k azonosítójuk alapján előkészítik a szükséges virtuális szál és blokk azonosítókat. Ha az adott párhuzamos architektúrában tt szál áll rendelkezésre, akkor a virtuális azonosítókat az alábbi módon számítom ki:

$$VT_{lvl_x}^k = \{vt_{lvl_x} | vt_{lvl_x} = (t_{id}^k + n \cdot tt) \mod |\Omega|^{(lvl_{x-1} - lvl_x)}, \\ n = 0 : \lceil eval_{lvl_x}/tt \rceil - 1\},$$



3. ábra. A PSD algoritmus csúcs kifejtő és kiértékelő folyamata.

$$VB^k_{lvl_x} = \{vb_{lvl_x} | vb_{lvl_x} = \lfloor (t^k_{id} + n \cdot tt)/|\Omega|^{(lvl_{x-1}-lvl_x)} \rfloor, \\ n = 0 : \lceil eval_{lvl_x}/tt \rceil - 1\}.$$

A *kiválasztás, leképezés és egyesítés blokkban* a már kiértékelt és kiválasztott részleges szimbólum vektorokat bővítem. A kiválasztási fázisban minden szál a saját $vb_{lvl_x} \in VB^k_{lvl_x}$ virtuális blokk azonosítói alapján kiválaszt már korábban kiértékelt részleges szimbólum vektorokat $\mathbf{s}^N_{lvl_{x-1}}$, leképezi a virtuális szál azonosítókat $vt_{lvl_x} \in VT^k_{lvl_x}$ részleges szimbólum vektorokra $\mathbf{s}^{lvl_{x-1}-1}_{lvl_x}$, végül ezek egyesítése $\mathbf{s}^{N<j>}_{lvl_x} = (\mathbf{s}^{lvl_{x-1}-1<j>}_{lvl_x}, \mathbf{s}^{N<j>}_{lvl_{x-1}})$ fogja létrehozni a következő lvl_x kiértékelési szint (részleges) szimbólum vektorait.

Az *útmetrikát kiértékelő blokkban*, a kibővített részleges szimbólum vektorok útmetrikáit frissítem. Ez az egyik legidőigényesebb lépés, viszont az útmetrikákat párhuzamosan több szál is frissíti.

A *keresési vagy rendezési blokk* a detekció folyamatának egyik legfontosabb fázisa. A keresési szinttől függően rendezésre vagy minimum keresésre kerül sor. A minimum keresést a detekció utolsó keresési szintjén alkalmazom, ugyanis ezen a szinten csak a legjobb metrikával

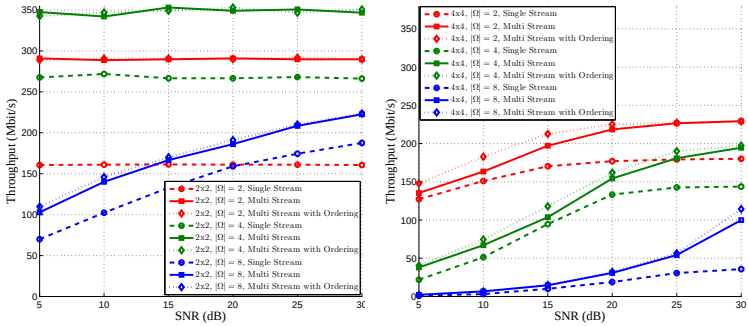
rendelkező teljes szimbólum vektort kell megtalálni. A minimum kereső algoritmust a párhuzamos prefix összegző (parallel prefix sum) párhuzamos minta alapján építettem fel. A költségesebb rendezést a rendező hálózatokkal valósítottam meg. Az adat-független felépítés és teljesen merev műveleti sorrend alkalmassá teszi a rendező hálózatok párhuzamos implementációját a GP-GPU architektúrán.

A konklúzió, hogy a párhuzamos építőelemek és a kismértékű szinkronizáció szükségessége lehetővé teszik az MPA-k nagyon hatékony használatát. Összehasonlítottam a PSD algoritmussal elért átlagos detekciós teljesítményt, az irodalomból ismert optimális ML implementációkkal. A PSD algoritmus felülmúlta mindegyiket. Továbbá összehasonlítottam számos FPGA, DSP, ASIC és GP-GPU megvalósítását szuboptimális detektoroknak. A PSD átlagos detekció teljesítménye sok esetben jobb volt a szuboptimális megoldásokénál. Néhány FPGA és VLSI alapú detektor jobb detekciós időt ért el a bithibaarány jelentős romlásával.

Tézis I.c.

Bevezettem egy dinamikus terheléselosztást ütemező algoritmust, amely hatékonyan ötvözi a rendszerszintű és eszközszintű párhuzamosságot. A kidolgozott ütemezési algoritmus által a szálblokkok és a szimbólum vektorok között egy dinamikus összerendelés valósul meg, ami lehetővé teszi a csökkentett szálblokk számú CUDA kernelek használatát. A szálblokkok számának csökkentésével a "stream" processzorok erőforrásai több kernel által is elérhetők, ezáltal a különböző CUDA folyamokon futó kernelek átlapolodási ideje nagyobb lesz. Így minimálisra csökkentettem a szimbólum detekció változó komplexitása által okozott, feldolgozó egységekben mért üresjárati időt tovább növelve az átlagos detekciós teljesítményt.

A rendszer szintű párhuzamosság egy adatkeret blokk fadinges csatornáinak a párhuzamos feldolgozásaként definiálható. Egy adott csatorna realizációhoz több szimbólum is tartozik. Következésképpen, a futtatott kernelek száma megegyezik a csatornák független realizációinak a számával. A kernelhez rendelt szálblokkok hajtják végre egy adott csatornához tartozó szimbólum vektorok detekcióját. Tehát, a szálblokkok és a szimbólum vektorok közti hozzárendelés meghatározása kritikussá válik, hiszen több szálblokk a GP-GPU több erőforrását foglalja le, és ezáltal csökken az egyes kernelek átlapolodásának a valószínűsége, ami



4. ábra. A PSD algoritmus átlagos detekciós teljesítménye $|\Omega| = 2$, 4 and 8 elemű jelkészletű (a) 2×2, (b) 4×4 MIMO rendszerek esetén.

teljesítmény romláshoz vezet.

Egy naiv hozzárendelési megközelítés, hogy minden szimbólum vektorhoz külön szálblokk tartozzon. Ezáltal a szálblokkok száma nagy lesz, ami a kernelek egyidejű futását korlátozza. Ha kevesebb, azonos terhelésű szálblokk végzi el a szimbólum vektorok detekcióját, a változó komplexitás miatt megjelenhet a végződési effektus (tail effect), ami a számítási egységek kihasználatlanságához vezet.

Az általam javasolt dinamikus számítási terhelést ütemező algoritmus lényegesen kevesebb szálblokkot használ, mint az egy-az-egyhez hozzárendelést megvalósító naiv megközelítés. A dinamikus terhelés elosztás lényege, hogy amint egy szálblokk befejezi egy szimbólum vektor detekcióját, azonnal megkezdődik a következő soron következő feldolgozatlan szimbólum vektor detekciója. Mivel a szimbólum vektorok detekciós ideje változó, a szálblokkok különböző számú szimbólum vektorokat dolgoznak fel és a végződési effektus is mérséklődik. Ennek eredményeként az eszköz szintű párhuzamosság fokozható, azaz a kernelek átlapolodási ideje megnő.

A 4. ábrán látható a dinamikus terheléselosztást ütemező algoritmus hatása egy $|\Omega| = 2$, 4 és 8 elemű jelkészletű 2×2 és 4×4 MIMO rendszeren. Az átlagos detekciós teljesítmény $15\% - 30\%$-os növekedése figyelhető meg $|\Omega| = 2$, 4 elemű jelkészlet esetén, illetve $38\% - 64\%$-os teljesítmény javulás $|\Omega| = 8$ elemű jelkészlet esetén.

II. Téziscsoport - MIMO detekciót segítő csatorna előfeldolgozási technikák.

(Kapcsolódó publikációk: [1], [3].)

Tézis II.a.

Kísérletileg igazoltam, hogy az inverz csatorna mátrix sor normái alapján meghatározott szimbólumok detektálási sorrendje csökkenti a PSD algoritmus komplexitását. A rendezés célja, hogy a kisebb jelerősségű szimbólumok detektálása olyan szinteken valósuljon meg, ahol teljes szélességi keresés van, így maximalizálható annak a valószínűsége, hogy a legjobb útmetrikával rendelkező részleges szimbólum vektor egyben az optimális legyen. Megmutattam, hogy az adott inverz csatorna mátrix sor normái alapján bevezetett rendezés növeli az átlagos detekciós teljesítményt és csökkenti a bejárt csúcsok számát.

A szukcesszív interferencia kiküszöbölésen [Successive Interference Cancellation (SIC)] alapuló detektorok bithibaaránya jelentősen függ a szimbólum detekció sorrendjétől. Ha egy hibásan detektált szimbólum hatása kerül kivonásra a vett jelből, akkor a zaj mértéke nő ahelyett, hogy az interferencia mértéke csökkenne. Az irodalomban számos rendezési metrikát vezettek be [16]. A legfontosabb rendezési metrikák a jel-zaj viszonyra, a jel-interferencia-plusz-zaj viszonyra és a csatorna mátrix oszlop normáira épülnek.

A legkisebb számítási komplexitása a csatorna mátrix oszlop norma alapú metrikának van, mely a MIMO modell következő felbontásából számítható

$$\mathbf{y} = \mathbf{H}\mathbf{s}_t + \mathbf{v} = \mathbf{h}_1 s_1 + \mathbf{h}_2 s_2 + \cdots + \mathbf{h}_n s_n + \mathbf{v} \tag{1}$$

ahol $\mathbf{h}_i$ jelöli a $\mathbf{H}$ csatorna mátrix i-edik oszlopát. Ennek eredményeként, a vett jel erőssége arányos a rendezési metrikával.

A SIC alapú algoritmusoknál először a legnagyobb energiájú szimbólumok detektálása szükséges, ugyanis ebben az esetben a hiba valószínűsége minimális. Azonban a PSD algoritmus a detektálási folyamatot a legalacsonyabb metrikájú szimbólummal kezdi, ugyanis az első keresési szinten egy teljes szélességi bejárás valósul meg, és mivel a legjobb útmetrikával rendelkező szimbólum vektorokkal folytatódik a de-

tekció, a hiba valószínűsége jelentősen csökkenthető. A 4. ábrán látható, ahogy az inverz csatorna mátrix sor normái alapján meghatározott szimbólum detekciós sorrend $5 - 10\%$-al növeli az átlagos detekciós teljesítményt.

III. Téziscsoport - Csökkentett komplexitású párhuzamos rácsredukciós algoritmusok leképezése masszívan párhuzamos és heterogén architektúrákra.

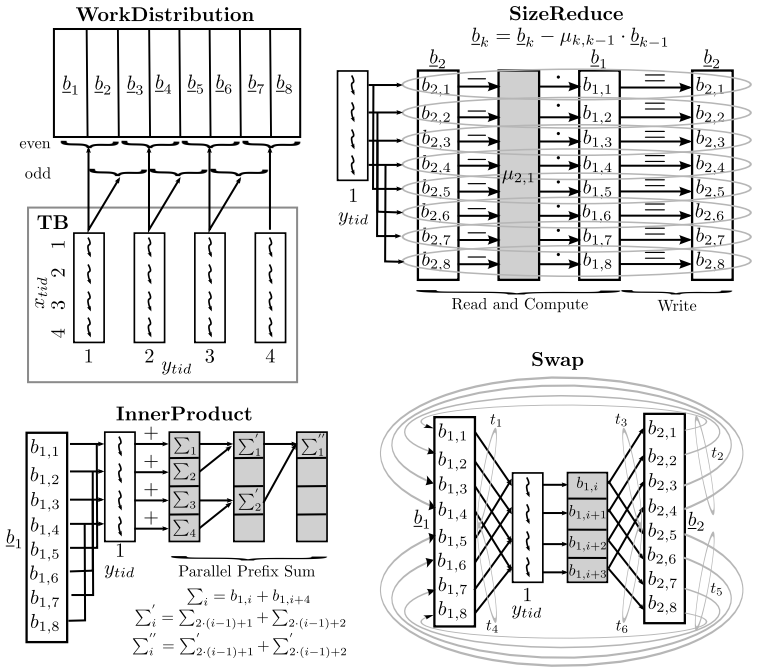
(Kapcsolódó publikációk: [2], [4], [5].)

Tézis III.a.

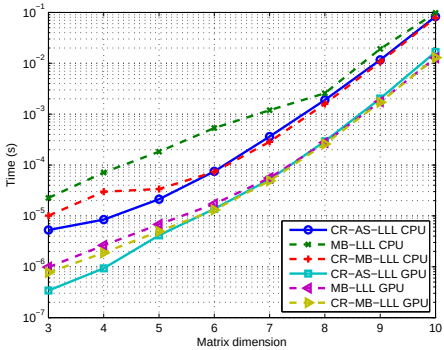
Bevezettem a párhuzamos költségcsökkentett egyidejű oszlopcseréken alapuló LLL [Cost-Reduced All-Swap LLL (CR-AS-LLL)] rácsredukciós algoritmust, amely a költségcsökkentés, a méret csökkentés és oszlopcsere procedúrák futtatása után késlelteti az átlón kívüli Gram-Schmidt együtthatók aktualizálását. Egy kétdimenziós szálblokk konfiguráció alapján leképeztem a CR-AS-LLL algoritmust a GP-GPU architektúrára, ezáltal, egy hatékony szálak közötti munkamegosztás, memória hozzáférés, skalárszorzat számítás és méret csökkentés végezhető el. A GP-GPU-ra való leképezés egy nagyságrenddel javítja az átlagos futási időt a többmagos CPU implementációval szemben.

A párhuzamos "All-Swap LLL" algoritmus esetén a Gram-Schmidt együtthatókat aktualizálom minden mátrix oszlopcsere és méret csökkentési procedúra után. Mivel a gyakori oszlopcsere és méret csökkentés több alkalommal módosítja a Gram-Schmidt együtthatók értékét, ezért egyes együtthatók aktualizálása feleslegessé válik. Az általam javasolt CR-AS-LLL algoritmusban csak a diagonális feletti $\mu_{k,k-1}$ Gram-Schmidt együtthatók vannak rendszeresen frissítve, mivel az LLL feltételek kiértékelése csak ezektől függ. A többi együttható csak akkor aktualizálódik, ha már nincs szükség további oszlopcsere és méretcsökkentési műveletre.

A CR-AS-LLL algoritmus teljesítménye a GP-GPU architektúrára való leképzés esetén a szálak közötti munkamegosztástól és a legfontosabb műveletek, mint a skalárszorzat számítás, méret csökkentés és oszlopcsere, implementációjának hatékonyságától függ. A 5. ábrán látható a CR-AS-LLL algoritmus fő műveleteinek egy lehetséges leképezése. Az



5. ábra. CR-AS-LLL algoritmus műveleteinek leképezése GP-GPU architektúrára.



6. ábra. A CR-AS-LLL, MB-LLL és CR-MB-LLL rácsredukciós algoritmusok futási idejének összehasonlítása $2^3 - 2^{10}$ dimenziójú mátrixok esetén.

egy kernelben futtatott szálblokkok száma egyenlő az egyidejűleg feldolgozott rácsbázisok számával. A szálblokkok kétdimenziós szál konfigurációval rendelkeznek, ahol T_x az x, és T_y az y dimenzió szálainak számát jelölik. A T_y szálak száma a bázis mérete alapján van meghatározva, azaz $T_y = \min(n/2, 32)$. Azáltal, hogy az x dimenzióban engedélyezett $T_x = \min(n, 32)$ szál használata, az azonos y dimenzióhoz tartozó szálak egy láncot (warp) alkotnak. Következésképpen, a globális memóriában tárolt $\mathbf{B}, \mathbf{B}^*$ mátrixok olvasása és írása az összekapcsolt (coalesced) hozzáférési minta segítségével valósul meg, amely kihasználja a teljes memória sávszélességet. A gyors elérésű osztott memória mérete korlátozott, ezért az LLL feltételek kiértékeléséhez szükséges Gram-Schmidt együtthatókat tárolja. A skalárszorzatok kiszámításában és az oszlopcserék gyorsításában is fontos szerepet tölt be.

Az 6. ábra megmutatja a CR-AS-LLL algoritmus GP-GPU és CPU implementációjának átlagos számítási idejét. A GP-GPU minden mátrix dimenzió esetén 6-15-ször gyorsabb a CPU-nál.

Tézis III.b.

Bevezettem a költségcsökkentett blokk alapú LLL [Cost-Reduced Modified-Block LLL (CR-MB-LLL)] rácsredukciós algoritmust, amely egy két szintű párhuzamosságot valósít meg, ezáltal csökkentve a magasabb dimenziójú rácsbázisok redukciós idejét. A magasabb szintű párhuzamosság alapja a blokkredukciós koncepció, amely az eredeti bázist több, alacsonyabb dimenziójú almátrixra osztja, majd az almátrixok rácsredukcióját a párhuzamos CR-AS-LLL algoritmus végzi. Megmutattam, hogy nagyméretű mátrixok esetén a CR-MB-LLL algoritmus hatékonyabb, mint a CR-AS-LLL algoritmus.

A párhuzamosság egy szintjének lehet tekinteni, ha egy probléma felbontható egyidejűleg végrehajtható részfeladatokra. A párhuzamosság második szintjének tekinthető, ha egy részfeladat ki tudja használni a többszálú környezet lehetőségeit. Az eddigi párhuzamos LR megvalósítások csak a többmagos architektúrákra fókuszáltak. A modern processzorok által kínált alacsony számú szálnak fő hátránya (szemben a GP-GPU-val), hogy az alacsony szintű párhuzamosságot nem lehet hatékonyan kihasználni. Egy algoritmus tervezése során az alacsony szintű párhuzamosság általában elhagyható, ezért a párhuzamossági szintek száma is korlátozott lesz. A GP-GPU-k esetében az ezres nagyságrendű CUDA magok számos szálat futtatnak párhuzamosan, ami

lehetővé teszi az alacsony szintű párhuzamosság kiaknázását, ezért jelentős teljesítmény növekedés érhető el.

A CR-MB-LLL algoritmus az eredeti bázist alacsonyabb dimenziójú almátrixokra osztja, majd az almátrixok rácsredukcióját a párhuzamos CR-AS-LLL algoritmus végzi. Mivel az egyes almátrixok rácsredukciója és a szomszédos határok ellenőrzése egymástól függetlenül végezhető el, nincs szükség a szálak gyakori szinkronizálása. A CR-MB-LLL algoritmus GP-GPU-ra való leképezése hasonló a CR-AS-LLL algoritmusnál bemutatott esettel, mert az elvégzendő műveletek ugyanazok.

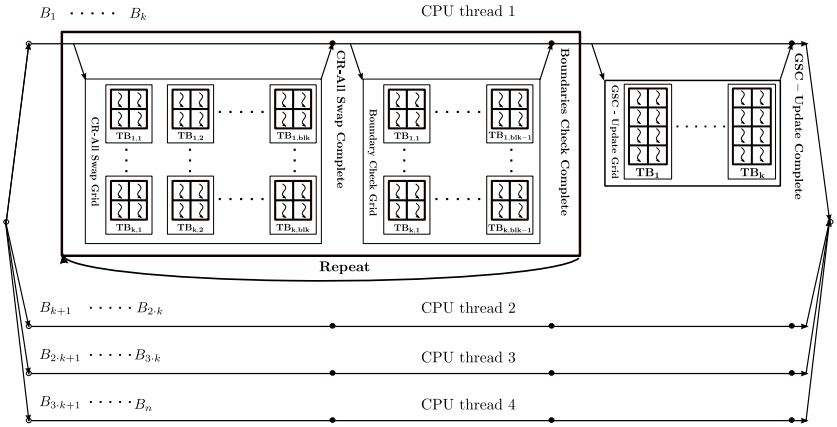
A CR-MB-LLL algoritmus tovább csökkenti a MB-LLL algoritmus számítási komplexitását az LLL feltételek ideiglenes lazításával. Az MB-LLL algoritmus esetén az almátrixok mindig teljesítik az LLL feltételeket. Ez azt jelenti, hogy ha két szomszédos almátrix határain oszlopcsere történik, akkor a Gram-Schmidt együtthatók aktualizálása is bekövetkezik az egyes almátrixokban. A CR-MB-LLL algoritmus komplexitásának csökkenése az almátrixok Gram-Schmidt együtthatóinak frissítésének kiiktatásával és egy egyszerűsített csere eljárással érhető el.

A 6. ábrán a többszintű párhuzamosságot implementáló algoritmusok számítási ideje kerül összehasonlításra. A CR-MB-LLL futási ideje $25-40\%$-al jobb a kis és közepes méretű mátrixok esetén az MB-LLL algoritmus futási idejénél. Továbbá nagy mátrixok esetén, a CR-MB-LLL által megvalósított blokk koncepció 30%-os gyorsítást ér el a CR-AS-LLL algoritmussal szemben.

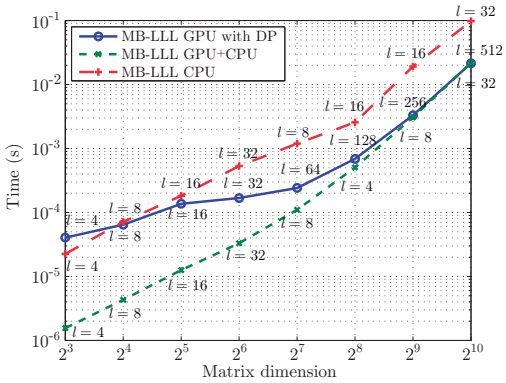
Tézis III.c.

A CR-MB-LLL algoritmushoz terveztem egy heterogén platformot, ahol a kernelek ütemezését a CPU szálak végzik, és a redukciós műveleteket a GP-GPU kernelek hajtják végre. Összehasonlítottam a tervezett heterogén platform teljesítményét a dinamikus párhuzamosságot kihasználó GP-GPU leképezéssel és egy párhuzamos CPU implementációval. Megmutattam, hogy a heterogén platform esetén az átlagos feldolgozási idő egy nagyságrenddel jobb a kis és közepes méretű mátrixoknál.

A heterogén platform vázlata a 7. ábrán látható. A CPU szálak feladata a CR-AS-LLL algoritmus, a határ feltételek kiértékelése, valamint a Gram-Schmidt együtthatók aktualizálását és csökkentését implementáló kernelek dinamikus ütemezése és futtatása.



7. ábra. A CR-MB-LLL algoritmust futtató kernelek ütemezése a heterogén platformon.



8. ábra. Az MB-LLL algoritmus futási ideje különböző architektúrákon, ahol l az optimális blokk méretet jelöli.

Minden CPU szálhoz egyedi CUDA folyamot rendelek, amely lehetővé teszi a kernelek egyidejű futtatását és csökkenti a CUDA magok készenléti idejét. A almátrixok feldolgozottsági szintje a GP-GPU globális memóriájában folyamatosan frissül és ezt a CPU minden iterációban figyelembe veszi, azaz a CR-AS-LLL és a határ feltételeket kiértékelő kernelek szálblokkjainak száma dinamikusan változik a még nem redukált almátrixok számának függvényében. A Gram-Schmidt együtthatók aktualizálására és frissítésére csak akkor kerül sor, ha az egy CPU szálhoz tartozó összes almátrix teljesíti az LLL feltételeket, és nincs szükség további oszlopcserére az almátrixok között.

A 8. ábra az MB-LLL algoritmus különböző architektúrákon való futásának eredményét mutatja be. A teljesítmény meghatározásához egy Tesla K20 GP-GPU és egy Intel Core i7-3820 processzort használtam. A heterogén platform egyértelműen felülmúlja a dinamikus párhuzamosság alapú GP-GPU implementációt kis mátrixok esetén és minden esetben a CPU megvalósítást. A konklúzió, hogy a heterogén rendszer által megkövetelt adatátvitel a CPU és a GP-GPU között kevésbé időigényes, mint a dinamikus párhuzamosság alapú kernelek indítása és ütemezése, és az ebből fakadó CUDA folyamok korlátozása.

4. Az eredmények alkalmazhatósága

A rácsredukció egy nagyon fontos matematikai eszköze a pontrácsokra épülő problémák megoldásának. A rácsredukció számos területen betöltött kulcsszerepét, elméleti és gyakorlati alkalmazhatóságát az irodalomban megjelenő számos publikáció bizonyítja. Mivel a pontrácsok és a rácsredukciós módszerek fontos szerepet töltenek be számos területen, ezért a célom az volt, hogy a polinomrendű LLL rácsredukciós algoritmus teljesítményét tovább növeljem.

A III. téziscsoportban elért eredmények bizonyítják, hogy ezt a célt sikerült elérni, mivel csökkentettem az LLL algoritmus komplexitását, azonosítottam és kihasználtam az algoritmusban rejlő több szintű párhuzamosságot, ami lehetővé tette az algoritmus leképezését masszívan párhuzamos architektúrákra és heterogén platformokra. A párhuzamos architektúrák és a heterogén rendszerek erőforrásainak kiaknázásával a rácsredukció végrehajtási ideje jelentősen csökkent. A következő felsorolás összefoglalja, hogy a III. téziscsoportban elért eredmények mely területeken alkalmazhatók:

- A *vezeték nélküli kommunikáció* területén eredményeim gyorsítják: (i) a frekvencia-szelektív csatornák kiegyenlítését [21], (ii) az előkódolt ortogonális frekvencia-osztásos multiplex rendszerek kiegyenlítését [22], (iii) a többantennás rendszerek esetén a forrás és csatorna kódolást [23], és (iv) a szférikus detekció előfeldolgozását [24]. Rácsredukció alkalmazásával az alacsonyabb komplexitású lineáris és nemlineáris detektorok és előkódolok is teljes rendű diverzitást érnek el [19], [20]. Ezen rendszerek számítási komplexitását a rácsredukció határozza meg, viszont a III. téziscsoportban bemutatott eredményeim alkalmazásával a rácsredukció komplexitása csökkenthető ezáltal egy gyorsabb feldolgozási sebesség is elérhető.
- Eredményeim alkalmazhatók a *képfeldolgozás* területén is a radar és mágneses rezonancia alapú képalkotás, és a JPEG képek színtér becslésének [25] és [26] gyorsításában is.
- A *kombinatorikus matematika* területén is több probléma visszavezethető pontrácsokon megfogalmazott problémákra. Pontrács alapú problémák megfogalmazhatók az egészértékű lineáris programozás [27], a hátizsák (knapsack) probléma [28], a racionális együtthatójú polinom faktorizáció [18] és a Diophantine közelítés témakörökben is. Ezen problémák megoldásának gyorsításában is alkalmazhatók a III. téziscsoport eredményei.
- A *kriptográfia* területén is kulcsszerepe van a rácsredukciónak [29],

ahol a feldolgozási idő kritikus szerepet tölt be.

Információelméleti kutatások során kiderült, hogy jelentős adatátviteli sebesség növekedést lehet elérni, ha az adó és vevő is több antennával rendelkezik [8]. A megnövekedett teljesítmény a felmerülő jelfeldolgozási problémák komplexitásának növekedését eredményezi. A MIMO rendszerek optimális ML detekciójának komplexitása exponenciálisan nő az adóantennák számával és a moduláció rendjével, így gyakorlati rendszerekben való alkalmazása nem hatékony. A szférikus detektor megalkotásával és továbbfejlesztésével [15], [28], [24] a keresési tér jelentősen csökkenthető, viszont az SD erősen szekvenciális jellege nem teszi lehetővé annak hatékony implementációját a masszívan párhuzamos architektúrákon.

Az I. téziscsoportban bemutatott PSD algoritmussal megszűntettem a szférikus detektor szekvenciális komponenseit, ezáltal lehetővé vált az algoritmus hatékony leképezése masszívan párhuzamos architektúrákra. A II. téziscsoportban alkalmazott csatorna mátrix alapú rendezési metrikák segítségével a PSD algoritmus detekciós teljesítményét tovább javítottam. Ezen eredmények alkalmazásával a magasabb rendű MIMO rendszerek optimális bithibaarány meghatározása sokkal gyorsabb lett.

Korábban bizonyításra került, hogy a szférikus detektort megvalósító algoritmus megoldja a legközelebbi rácspont problémát [Closest Lattice Point Problem (CLP)], vagy az ezzel egyenértékű legrövidebb vektor problémát [Shortest Vector Problem (SVP)] [24], [30], [31]. Mivel több optimális rácsredukciós módszer és kriptográfiai probléma is visszavezethető a CLP és SVP problémákra, az I. és II. téziscsoportok eredményei használhatók ezen problémák hatékonyabb megoldására.

5. Köszönetnyilvánítás

Elsősorban hálás vagyok témavezetőimnek Kolumbán Géza és Szolgay Péter professzor uraknak a doktori képzés során nyújtott útmutatásaikért, türelmükért és támogatásukért. Őszinte hálával tartozom Roska Tamás professzor úrnak az inspiráló és gondolatébresztő vitákért és önzetlen segítségéért. Hálával és köszönettel tartozom Csurgay Árpád professzor és Oláh Andrásnak docens uraknak a folytonos bátorításért és motivációért, ami mindig erőt adott a folytatáshoz.

Kiemelt köszönet illeti Vidal Antonio professzor urat, hogy a Valenciai Műszaki Egyetem keretében működő "Institute of Telecommunications and Multimedia Applications" (iTEAM) kutatócsoportjába teljes értékű tagként befogadott. Hálásan köszönöm Piñero Gema professzor asszony, González Alberto és Martínez-Zaldívar Francisco professzor urak vezeték nélküli kommunikáció területén nyújtott segítségüket.

Szeretnék köszönetet mondani minden kedves barátnak és kollégának, akikkel az elmúlt éveket töltöttem: *Bihary Dóra, Borbély Bence, Gelencsér Zsolt, Hiba Antal, Jákli Balázs, Laki András, László Endre, Sárkány Norbert, Reguly István, Rudán János, Tuza Zoltán, Fülöp Tamás, Horváth András, Koller Miklós, Radványi Mihály, Rák Ádám, Stubendek Attila, Tornai Gábor, Zsedrovits Tamás, Balogh Ádám, Fekete Ádám, Füredi László, Tornai Kálmán, Tar Ákos, Tisza Dávid, Treplán Gergely, Veres József, Bojárszky András, Karlócai Balázs, Krébesz Tamás.* Külön köszönet Reguly Istvánnak az órákig tartó szakmai vitákért és folyamatos együttműködésért.

Köszönet spanyol kollégáimnak, hogy bevezettek a spanyol kultúra szépségeibe és értékessé tették az ott töltött időt *Aguilera Emanuel, Belloch J. Antonio, Domene Fernando, Fuster Laura, Gutiérrez Pablo, Lorente Jorge, Maciá Luis, Martí Amparo, Ramiro Carla.*

Köszönöm a Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionika Karának, hogy messzemenően támogatta képzésemet a TÁMOP-4.2.1/B-11/2/KMR-2011-0002 és TÁMOP-4.2.2/B-10/1-2010-0014 ösztöndíjak által.

Végül, hálával tartozom szüleimnek Enikőnek és Máténak, testvéremnek Istvánnak, nagyszüleimnek Böbének, Évának és Mártonnak, hogy elviselték távollétemet és mindeközben minden elképzelhető módon támogattak.

Végül, de nem utolsósorban, szeretném kifejezni őszinte hálámat és köszönetemet feleségemnek Ildikónak, hogy nagy szeretettel és türelemmel támogatott a végtelennek tűnő, hosszú, rögös út során.

References

Author's journal publications

[1] **Csaba M. Józsa**, Géza Kolumbán, Antonio M. Vidal, Francisco J. Martínez-Zaldívar, and Alberto González. "Parallel Sphere Detector algorithm providing optimal MIMO detection on massively parallel architectures". In: *Concurrency and Computation: Practice and Experience* (2015). DOI: 10.1002/cpe.3488.

[2] **Csaba M. Józsa**, Fernando Domene, Antonio M. Vidal, Gema Piñero, and Alberto González. "High performance lattice reduction on heterogeneous computing platform". In: *The Journal of Supercomputing* (2014), pp. 1–14. ISSN: 0920-8542. DOI: 10.1007/s11227-014-1201-2.

Author's conference publications

[3] **Csaba M. Józsa**, Géza Kolumbán, Antonio M. Vidal, Francisco-José Martínez-Zaldívar, and Alberto González. "New Parallel Sphere Detector Algorithm Providing High-Throughput for Optimal MIMO Detection". In: *2013 International Conference on Computational Science (ICCS 2013)*. Vol. 18. Barcelona, Spain, 2013, pp. 2432 –2435. DOI: http://dx.doi.org/10.1016/j.procs.2013.05.417.

[4] **Csaba M. Józsa**, Fernando Domene, Gema Piñero, Alberto González, and Antonio M. Vidal. "Efficient GPU implementation of Lattice-Reduction-Aided Multiuser Precoding". In: *Wireless Communication Systems (ISWCS 2013), Proceedings of the Tenth International Symposium on*. Ilmenau, Germany, Aug. 2013, pp. 1–5. ISBN: 978-3-8007-3529-7.

[5] Fernando Domene, **Csaba M. Józsa**, Antonio M. Vidal, Gema Piñero, and Alberto González. "Performance analysis of a parallel Lattice Reduction algorithm on many-core architectures". In: *The 13th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE 2013)*. Vol. 2. Almeria, Spain, June 2013, pp. 535–542. ISBN: 978-84-616-2723-3.

[6] Tamás Krébesz, **Csaba M. Józsa**, and Géza Kolumbán. "New carrier generation techniques and their influence on bit energy in UWB radio". In: *Circuit Theory and Design (ECCTD), 2011 20th European Conference on*. IEEE. Aug. 2011, pp. 801–804. DOI: 10.1109/ECCTD.2011.6043838.

[7] Tamás Krébesz, Géza Kolumbán, and **Csaba M. Józsa**. "Ultra-wideband impulse radio based on pulse compression technique". In: *Circuit Theory and Design (ECCTD), 2011 20th European Conference on*. IEEE. Aug. 2011, pp. 797–800. DOI: 10.1109/ECCTD.2011.6043839.

Related publications

[8] Emre Telatar. "Capacity of Multi-antenna Gaussian Channels". In: *European Transactions on Telecommunications* 10.6 (1999), pp. 585–595. ISSN: 1541-8251.

[9] Ezio Biglieri, Robert Calderbank, Anthony Constantinides, Andrea Goldsmith, Arogyaswami Paulraj, and H. Vincent Poor. *MIMO Wireless Communications*. New York, NY, USA: Cambridge University Press, 2007. ISBN: 0521873282.

[10] Michael Wu, Yang Sun, Siddharth Gupta, and Joseph R. Cavallaro. "Implementation of a High Throughput Soft MIMO Detector on GPU". In: *J. Signal Process. Syst.* 64.1 (July 2011), pp. 123–136. ISSN: 1939-8018.

[11] Rongchun Li, Yong Dou, Dan Zou, Shi Wang, and Ying Zhang. "Efficient graphics processing unit based layered decoders for quasicyclic low-density parity-check codes". In: *Concurrency and Computation: Practice and Experience* 27.1 (2013), pp. 29–46. ISSN: 1532-0634.

[12] Rongchun Li, Yong Dou, and Dan Zou. "Efficient parallel implementation of three-point viterbi decoding algorithm on CPU, GPU, and FPGA". In: *Concurrency and Computation: Practice and Experience* 26.3 (2014), pp. 821–840. ISSN: 1532-0634.

[13] Fernando Domene, Sandra Roger, Carla Ramiro, Gema Pinero, and Alberto Gonzalez. "A reconfigurable GPU implementation for Tomlinson-Harashima precoding". In: *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. 2012.

[14] Wang Hongyuan and Chen Muyi. “A Fixed-Complexity Sphere Decoder for MIMO Systems on Graphics Processing Units”. In: *Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference on*. Dec. 2010.

[15] M. Pohst. “On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications”. In: *ACM SIGSAM Bulletin* 15.1 (1981), pp. 37–44.

[16] P.W. Wolniansky, G.J. Foschini, G.D. Golden, and R. Valenzuela. “V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel”. In: *Signals, Systems, and Electronics, 1998. ISSSE 98. 1998 URSI International Symposium on*. IEEE. Sept. 1998, pp. 295–300.

[17] D. Wubben, D. Seethaler, J. Jalden, and G. Matz. “Lattice Reduction”. In: *Signal Processing Magazine, IEEE* 28.3 (May 2011), pp. 70–91. ISSN: 1053-5888.

[18] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. “Factoring polynomials with rational coefficients”. In: *Mathematische Annalen* 261.4 (1982), pp. 515–534.

[19] Huan Yao and Gregory W. Wornell. “Lattice-reduction-aided detectors for MIMO communication systems”. In: *Global Telecommunications Conference, 2002. GLOBECOM ’02. IEEE.* Vol. 1. Nov. 2002, pp. 424–428.

[20] Christoph Windpassinger, Robert FH Fischer, Tomáš Vencel, and Johannes B Huber. “Precoding in multiantenna and multiuser communications”. In: *IEEE Trans. Wireless Commun.* 3.4 (2004), pp. 1305–1316.

[21] Wai Ho Mow. “Maximum likelihood sequence estimation from the lattice viewpoint”. In: *Information Theory, IEEE Transactions on* 40.5 (Sept. 1994), pp. 1591–1600. ISSN: 0018-9448.

[22] Xiaoli Ma, Wei Zhang, and A. Swami. “Lattice-reduction aided equalization for OFDM systems”. In: *Wireless Communications, IEEE Transactions on* 8.4 (Apr. 2009), pp. 1608–1613. ISSN: 1536-1276.

[23] R. Zamir, S. Shamai, and U. Erez. “Nested linear/lattice codes for structured multiterminal binning”. In: *Information Theory, IEEE Transactions on* 48.6 (2002), pp. 1250–1276. ISSN: 0018-9448.

[24] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. “Closest point search in lattices”. In: *Information Theory, IEEE Transactions on* 48.8 (2002).

[25] A. Hassibi and S. Boyd. “Integer parameter estimation in linear models with applications to GPS”. In: *Signal Processing, IEEE Transactions on* 46.11 (Nov. 1998), pp. 2938–2952. ISSN: 1053-587X.

[26] R.N. Neelamani, R.G. Baraniuk, and Ricardo de Queiroz. “Compression color space estimation of JPEG images using lattice basis reduction”. In: *Image Processing, 2001. Proceedings. 2001 International Conference on*. Vol. 1. 2001, pp. 890–893.

[27] Ravi Kannan. “Improved algorithms for integer programming and related lattice problems”. In: *Proceedings of the fifteenth annual ACM symposium on Theory of computing*. ACM. 1983, pp. 193–206.

[28] C. P. Schnorr and M. Euchner. “Lattice basis reduction: Improved practical algorithms and solving subset sum problems”. In: *Mathematical Programming* 66 (1 1994), pp. 181–199.

[29] PhongQ. Nguyen and Jacques Stern. “Lattice Reduction in Cryptology: An Update”. English. In: *Algorithmic Number Theory*. Ed. by Wieb Bosma. Vol. 1838. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2000, pp. 85–112. ISBN: 978-3-540-67695-9.

[30] M.O. Damen, H. El Gamal, and G. Caire. “On maximum-likelihood detection and the search for the closest lattice point”. In: *Information Theory, IEEE Transactions on* 49.10 (2003), pp. 2389–2402.

[31] B. Hassibi and H. Vikalo. “On the sphere-decoding algorithm I. Expected complexity”. In: *Signal Processing, IEEE Transactions on* 53.8 (2005).