

1 Ohjeita

Tässä listassa on ehdotuksia kandidaatintutkielma-kurssin aiheiksi. Valitse listalta joitain kandidaattiaihteita. Listan aiheet ovat suuntaa-antavia; lopullinen rajausta ja otsikko sovitaan kurssin kuluessa. Voit myös ehdottaa omaa aihettasi. Oma aihe on kaikkein paras vaihtoehto, kunhan se sopii aihepiiriin ja siitä löytyy riittävästi sopivaa materiaalia.

Kukin aihe tai aihepiiri on varustettu listalla avainsanoja, jotka kuvaavat aiheeseen liittyviä matemaatiikan ja tietojenkäsittelyn osa-aloja tai hyödyllisiä kursseja. Nämä eivät ole esitietovaatimuksia, mutta on selvää että taustatiedot ja menetelmiin liittyvä kiinnostus saattavat nopeuttaa artikkelien ymmärtämistä ja kirjoitustyötä. Peruslähtökohta on kuitenkin se, että opinto-oppaassa mainitut esitietovaatimukset antavat riittävät valmiudet kaikkiin aiheisiin.

Listalla on kustakin aiheesta mainittu muutama lähdeartikkeli tai kirja, joiden kautta voi päästä helpommin aihepiiriin kiinni. Valintariskin minimoimiseksi kannattaa silmäillä läpi esimerkiksi yksi lähdeartikkeli ennen kuin kiinnität valintasi.

Aiheisiin tai aihepiireihin on merkitty myös ohjaajan arvio kirjaimilla A–C. Kirjain A tarkoittaa aihepiiriä, joka on lähempänä ohjaajan omaa tutkimusta, ja kirjain C taas aihepiiriä, jossa ohjaajakin varmasti oppii uutta. Kaikista aiheista saa tuki ohjausta, myös omasta aiheestasi.

2 Aihteita

2.1 Koneoppimista, tekoälyä, data-analyysia ja tiedon louhintaa

Ohjaajan arvio: A. *Liittyy:* Todennäköisyyslaskentaa. *Kursseja:* Johdatus koneoppimiseen, Johdatus tekoälyyn, Todennäköisyysmallit, Tiedon louhinta, Laskennallinen data-analyysi, Kolme käsitettä -kurssit, Graphical models, todennäköisyyslaskennan ja tilastotieteen kurssit.

Algoritminvalintaongelma ja koneoppimiseen perustuvat algoritmiportfoliot [82, 129]

Annettuna joukko saman laskennallisen ongelman ratkaisevia algoritmeja, mikä algoritmi ratkaisee nopeiten syötteenä annettavan ongelmainstanssin? Tähän klassiseen algoritminvalintaongelmaan on kehitetty koneoppimiseen pohjautuvia menetelmiä useille keskeisille NP-vaikeille laskennallisille ongelmille.

Algoritmeja tietovirroille (*data stream algorithms*) [9, 46]

Sensori tai muu reaaliaikainen datalähde voi tuottaa dataa niin paljon ja nopeasti, että sitä ei ole mahdollista kokonaan tallettaa myöhempää käsittelyä varten. Erilaisia koko dataa koskevia tilastoja voidaan kuitenkin laskea luomalla siitä muistiin vain ”hahmotelma” (*sketch*), joka on kooltaan esim. logaritminen syötteen koon suhteen.

Tulon summaus -ongelma (*sum-product problem*) [8, 140, 141]

Tulon summaus -ongelmassa tehtävänä on laskea moniulotteinen summa funktiosta, joka voidaan esittää alempiulotteisten funktioiden tulona.

Liittyy: Algebra.

Päätöspuiden oppiminen [127]

Heuristiikkoja päätöspuiden muodostamiseen, päätöspuiden oppiminen.

Liittyy: Informaatioteoriaa.

Robottien navigointi [145]

Oppiminen robottien navigoinnissa, robottien navigointialgoritmit.

Liittyy: Matriisilaskenta, Markovin prosessit, Kalman-suodattimet, particle filtering -tyyppiset tilasiirtymäjärjestelmiin liittyvät tilastolliset estimointimenetelmät.

Itseorganisoivat kartat ja niiden sovellukset [98]

Itseorganisoivan kartan tarkoituksena on projisoida suuriulotteisessa vektoriavaruudessa sijaitseva data pieniulotteiseen (esim. kaksiulotteiseen) vektoriavaruuteen esimerkiksi datan visualisointia varten tai siksi että tyypilliset etäisyysmitat käyttäytyvät huonosti suuriulotteisissa avaruuksissa.

Itseorganisoivat kartat voidaan nähdä myös pääkomponenttianalyysin epälineaarisenä vastineena (ei suoria ominaisvektoreita eikä ortogonaalista kantaa).

Liittyy: Matriisilaskentaa.

Riippumattomien komponenttien analyysi [134]

Sokeassa lähteiden erottelussa havaittu signaalivektori x oletetaan toisitaan riippumattomien lähdesignaalien s sekoitteeksi ($x = As$). Tehtävänä on selvittää alkuperäinen signaalivektori S oletusten ja havaitun signaalivektorin perusteella niin pitkälle kuin mahdollista ($s = Wx$). Sovelluksia on mm. laskennallisessa neurotieteessä.

Liittyy: Matriisilaskentaa, informaatioteoriaa, tiedon louhintaa.

Ei-negatiiviset matriisihajotelmat [102]

Ei-negatiivisissa matriisihajotelmissa ei-negatiivinen matriisi hajotetaan kahden ei-negatiivisen matriisin tuloksi. Näin saadaan joissain tilanteissa helpommin tulkittavia hajotelmiä kuin esimerkiksi riippumattomien komponenttien analyysillä. Valitettavasti ei-negatiivisten hajotelmien laskeminen on melko vaikeaa.

Liittyy: Matriisilaskentaa, tiedon louhintaa.

Palauteoppiminen (*reinforcement learning*) [14, 91, 144]

Palauteoppimisessa oppija yrittää muodostaa tehokkaan toimintastrategian ympäristössä, jossa suoritettujen valintojen edut ja haitat näkyvät vasta myöhemmin. Algoritmeja on sovellettu menestyksellä mm. peleissä. Myös erilaisia teoreettisia analyyseja on mahdollista tehdä.

Liittyy: Markovin prosesseja.

Tukivektorikoneet (*support vector machines, SVM*) [135]

Tukivektorikoneet olivat 1990-luvun loppupuolen kuumiin koneoppimisparadigma. Tukivektorikoneita ja muita ydinfunktioihin (*kernel function*) perustuvia data-analyysimenetelmiä on sovellettu menestyksellisesti moniin tehtäviin, joihin aiemmin käytettiin etupäässä neuraaliverkkoja.

Liittyy: Optimointia.

Segmentointi [94]

Segmentoinnin tarkoituksena on sekvenssidatan jakaminen segmentteihin s.e. kunkin segmentin sisältämä data on keskenään samankaltaista ja eroaa naapurisegmenteistä. Algoritmisia kysymyksiä liittyy mm. datan esikäsittelyyn, oikean segmenttimäärän löytämiseen sekä segmenttien kuvauksen määrittelyyn. Sovelluksia esim. DNA-sekvenssin jakaminen geeneihin ja ei-koodaaviin alueisiin, sensoridatan jakaminen systeemin eri tiloja vastaaviin segmentteihin.

Piilo-Markov-mallit (*hidden Markov models, HMM*) [23]

Piilo-Markov-malleja käytetään moniin tietojenkäsittelyongelmiin puheentunnistuksesta geenisekvenssien analysointiin.

Liittyy: Markovin prosesseja. *Kursseja:* Graphical models.

Ryvöstämisen algoritmistiikkaa [13, 48, 116]

Ryvöstämisessä (*clustering*) suuri joukko pisteitä jaetaan pieneen määrään *rypäitä* siten, että samaan rypäeseen tulevat pisteet ovat jonkin metriikan mukaan lähellä toisiaan ja eri rypäisiin kuuluvat kaukana toisistaan. Ryvöstämisestä on useita muunnelmia ja niitä varten erilaisia algoritmeja. Ongelmien tarkka ratkaiseminen on laskennallisesti vaativaa, mutta joitakin approksimointialgoritmeja tunnetaan.

Paikkatiedon ryvöstäminen [114]

Assosiaatiot paikkatiedossa [80]

Liittyy: Tiedonlouhinnan kurssit, assosiaatiosäännöt.

Emerging patterns, change patterns [53]

Ryvöstäminen korkeaulotteisissa avaruuksissa [7]

Ryvöstettäessä hyvin korkeaulotteista dataa kaikkia datan ulottuvuuksia käyttävät metriikat toimivat usein huonosti. Ongelmaan on esitetty ratkaisuksi algoritmeja, jotka tunnistavat rypäitä alkuperäisen avaruuden aliavaruuksissa.

Tiedon louhinta verkoista [101]

Usein datan rakenne on kuvattavissa luontevasti verkkona, jolloin usein toistuvien hahmojen etsiminen muuttuu verkossa usein toistuvien aliverkkojen etsimiseksi. Ongelmien muotoilussa ja algoritmeissa on huomioitava klassisia verkkoteorian kysymyksiä kuten verkkojen isomorfisuus.

Parametriton tiedon louhinta [93]

2.2 Laskennan teoriaa

Ohjaajan arvio: A. Kurseja: Laskennan mallit, Laskennan vaativuus, Laskennan teoria, logiikan kurssit.

Ekspontiaaliaikahypoteesi (ja ajoaika-alarajoja polynomiaikaisille ongelmille) [124]

Ekspontiaaliaikahypoteesin mukaan tietyille NP-täydellisille ongelmille ei ole olemassa muita kuin brute force -algoritmeja. Olettamalla tämä hypoteesi on yllättäen mahdollista osoittaa ajoaika-alarajoja useille keskeisille polynomiaikaisille ongelmille.

PSPACE: laskentaa polynomisessa tilassa [118]

Missä luokka NP sisältää ongelmat, joiden kaikille instanssille annettu mielivaltainen ratkaisu voidaan tarkistaa polynomisessa ajassa, PSPACE sisältää ongelmat, joiden instanssien ratkaisut voidaan tarkistaa polynomisessa tilassa. Näin ollen PSPACE on sisältää NP:n ja lisäksi useita keskeisiä, oletettavasti NP:tä vaikeampia laskennallisia ongelmia.

Kvantifoidut Boolean kaavat (*quantified Boolean formulae, QBF*) [34, 65, 130]

Kvantifoidut Boolean kaavat laajentavat propositiologiikkaa sallimalla eksistentiaali- ja universaali-kvanttorit. Lähdeartikkeleissa tarkastellaan esimerkiksi QBF-kaavojen SAT-ongelmaa eli kaavojen toteutuvuutta.

Resoluutiotodistusten vaikeus [4, 125]

Toteutuvuusongelma on tutkituimpia NP-täydellisiä ongelmia. Viime aikoina on saatu uusia tuloksia toteutuvuustodistusten vaikeudesta.

Kiintoparametrivaativuus (*fixed-parameter complexity*) [37]

Tietyille NP-täydellisille ongelmille on olemassa algoritmeja, joiden aikavaativuus on $O(f(k) \text{ poly}(n))$, missä n on syötteen koko, k ongelman parametri (esimerkiksi solmupeitteen koko), $f(k)$ on funktio, joka ei riipu parametrasta n ja $\text{poly}(\cdot)$ on polynomi. Tällaiset algoritmit pystyvät käsittelemään suuriakin syötteitä, jos parametri k on kiinnitetty riittävän pieneksi.

Laskentapiirien teoriaa [26, 79, 128]

Laskentapiirit ovat laskennan malli, joka jäljittelee loogisista peruskomponenteista muodostuvaa elektronista piiriä. Laskennan piirivaativuus on mielenkiintoista etenkin alarajojen kannalta. Klassinen esimerkki on Hästadin tulos pariteettifunktioille. Ikävä kyllä epätriviaalien alarajojen todistaminen on tässäkin mallissa osoittautunut hyvin vaikeaksi.

Helppojen ongelmien vaativuus [28, 89]

Vaativuusteoriassa ollaan yleensä ensisijaisesti kiinnostuneita siitä, mitkä ongelmat kuuluvat luokkaan P . Kuitenkin myös luokkaan P kuuluvien ongelmien välisiä eroja voidaan tutkia. Eräs mielenkiinnon kohde on logaritmisessa työtilassa ratkeavat ongelmat, joiden voidaan osoittaa olevan tietystä mielessä tehokkaasti rinnakkaistuvia.

Satunnaislaskennan vaativuusteoriaa [88, 89]

Satunnaisuus on tärkeä algoritmien tekniikka, jota käytetään monella lailla eri tyyppisten ongelmien ratkaisussa. Tämän aiheen tarkoituksena on tarkastella satunnaisalgoritmien yleistä teoriaa: millaisia vaativuusluokkia satunnaislaskennassa voidaan määritellä ja miten ne suhtautuvat toisiinsa ja deterministisiin vaativuusluokkiin.

Kolmogorov-kompleksisuus [63, 86, 87]

Kolmogorov-kompleksisuus on eräänlainen satunnaisuuden mittari. Bittijonon monimutkaisuus on yhtä kuin lyhimmän sellaisen tietokoneohjelman pituus, jonka tulosteena on kyseinen bittijono. Toisin sanoen, jos ohjelma on olennaisesti ”yhtä pitkä” kuin bittijono itse, bittijonoa voidaan pitää ”satunnaisena” siinä mielessä ettei sen rakenteessa ollut mitään säännönmukaisuutta, jonka perusteella ohjelma voisi tehdä muuta kuin muistaa bittijonon suoraan sellaisena kuin se tulostetaan.

2.3 Tietorakenteita ja algoritmeja

Ohjaajan arvio: B. Kurseja: Tietorakenteet, Algoritmien suunnittelu ja analyysi, Merkkijonomenetelmät, Tiedon tiivistämisen tekniikat, Advanced data structures.

Joukkojen erilliset yhdisteet [143, 151]

Joukkojen yhdisteiden säilyttäminen ja purkaminen.

Binääriset etsintäpuut [137]

Splay-puu on itsesäätyvä versio perinteisestä binäärisestä hakupuusta.

Liittyy: Tasoitettua analyysiä.

Tietorakenteiden tiivistys [40, 84]

Tiedon tiivistyksessä on tavoitteena esittää tieto (esim. tekstidokumentti) mahdollisimman pienessä tilassa siten, että alkuperäinen tieto voidaan palauttaa virheettömästi. Tietorakennetta tiivistettäessä lisätavoitteena on säilyttää tietorakenteen toimintakyky. Esimerkki: Binääripuu halutaan esittää muodossa, jossa kullekin solmulle löydetään vakioajassa sen vasen ja oikea lapsi. Perinteinen linkein esitetty puu vie $O(n \log n)$ bittiä tilaa, missä n on puun solmujen määrä (jokaiseen solmuun on talletettu 2 linkkikenttää, kukin luokkaa $\log n$ bittiä). On kuitenkin mahdollista esittää puu $O(n)$ bitillä siten, että vasen ja oikea lapsi löytyvät vakioajassa.

Liittyy: Informaatioteoriaa.

Hahmonsovitukset merkkijonoissa [47, 97]

Merkkijonomenetelmien avulla pyritään löytämään tekstistä hahmon osumia eli paikkoja, joissa hahmo esiintyy osin tai kokonaan.

Staattisen tekstin indeksointi loppuosatietorakenteilla [107]

Suuri osa internetin sisällöstä on tekstiä, joka muuttuu suhteellisen hitaasti. Tällaiselle staattiselle tekstille on kehitetty indeksirakenteita, jotka mahdollistavat erittäin nopeat tekstihaut. Loppuosatietorakenteet kuten suffiksipuut ja -taulukot ovat perusesimerkkejä tällaisista rakenteista.

Muistihierarkia-algoritmit [20, 60]

Välimuistin käyttö vaikuttaa usein merkittävästi algoritmien käytännön tehokkuuteen. Eräs mielenkiintoinen idea on suunnitella algoritmit sellaisiksi, että ne käyttävät välimuistia suunnilleen optimaalisesti ilman, että niiden tarvitsee ennakolta tietää välimuistin kokoa. B-puut ovat tuore esimerkki mallin sovelluksista.

Burrows–Wheeler-muunnos [32, 108, 113]

Burrows–Wheeler-muunnos [32] on tärkeä tekstin esikäsittelymenetelmä, jota voidaan käyttää esim. hakemistojen tiivistämisessä [113]. Aiheesta on tehty myös matemaattisia analyyseja [108].

Käytännön algoritmeja lauselogiikan toteutuvuusongelmalle (SAT-ongelmalle) [24]

Teoreettisten tarkastelujen lisäksi keskeinen NP-täydellinen SAT-ongelma mahdollistaa useiden olennaisten käytännön ongelman instanssien ratkaisemisen jopa erikoisalgoritmeja tehokkaammin. Käytännön ongelmille, kuten ohjelmistojen ja mikroprosessorien mallintarkastukseen, tehokkaimmat eksaktit algoritmit pohjautuvat ns. ristiriitapohjaiseen klausuulioppimiseen (CDCL, conflict-driven clause learning). Myös paikallishakuun pohjautuvia algoritmeja on kehitetty.

Reilu jakaminen (*fair allocation*) [30]

Tavaroiden tai hyödykkeiden reilu jakaminen on yleisesti esiintyvä ongelma. Klassinen esimerkki tämäntyyppisistä ongelmista on ns. kakun leikkaus -ongelma, jossa tavoitteena on jakaa annettu kakku henkilöiden kesken mahdollisimman reilusti.

2.4 Kombinatorista optimointia

Ohjaajan arvio: B. *Liittyy:* Kombinatoriikka, verkot, puut, lineaarinen ohjelmointi. *Kursseja:* Algoritmien suunnittelu ja analyysi, Diskreetti optimointi, Kombinatorinen optimointi, Approksimointialgoritmit, Satunnaisalgoritmit, diskreetin matematiikan kurssit.

Simplex-algoritmi [120]

Lineaariohjelmat ovat keskeinen polynomiakaisesti ratkeava luokka rajoiteoptimointiongelmaa. Tyypillisesti käytännössä tehokkain algoritmi lineaariohjelmien ratkaisemiseksi on Simplex, riippumatta siitä, että tämän algoritmin pahimman tapauksen aikavaativuus tiedetään eksponentiaaliseksi. Simplex on myös

Kokonaislukuohjelmointi (*integer programming, IP*) [5, 120]

Kokonaislukuohjelmointi on keskeinen NP-vaikea rajoiteohjelmointiparadigma. Kokonaislukuoptimointialgoritmit pohjautuvat tyypillisesti ns. branch-and-cut -lähestymistapaan, jossa annetun kokonaislukuoptimointi-instanssi lineaarirelaksaation ratkaisujen joukkoa iteratiivisesti muokataan kohti instanssin oikeaa optimaalista ratkaisujen joukkoa. Kokonaislukuoptimointijärjestelmät mahdollistavat hyvien tai jopa optimaalisten ratkaisujen tehokkaan löytämisen useille reaali maailman optimointiongelmiin.

Olomuotosiirtymät laskennallisissa ongelmissa (*phase transitions*) [74]

Luonnossa esiintyville olomuotosiirtymille (faasitransitiot, phase transitions), kuten veden jäätyminen, sulaminen, höyrystyminen ja tiivistyminen, on havaittu löytyvän analogia laskennallisten ongelmien, kuten verkon väriytyksen ja SAT-ongelman, vaikeudessa sitä mukaan, kun laskennallisten ongelmien luontaisten rajoitteiden tiheyttä (kuten kaarten määrää verkossa tai klausuulien määrän konjunkttiivisessä normaalimuodossa olevassa lauselogiikan kaavassa) kasvatetaan, siirtyen ratkaisuja sisältävistä instansseista toteutumattomiin ongelmiin. Tämän siirtymän kynnyksellä myös ongelmainstanssien ratkaisemisen vaikeus on osoitettu huomattavasti kasvavan sekä käytännössä että teoriassa.

Rajoiteohjelmointi (*constraint programming, CP*) [133]

Rajoiteohjelmointi (constraint programming, CP) viittaa tyypillisesti tiettyyn deklarativiseen lähestymistapaan ratkaista usein NP-kovia hakuongelmia. Rajoiteohjelmoinnissa, toisin kuin yleisesti ottaen kokonaislukuohjelmoinnissa, muuttujat voivat saada arvoja ainoastaan äärellisestä arvojen

joukosta, ja tavoitteena on löytää arvojakelu, joka toteuttaa joukon annettuja (deklaroituja) tarkasti määriteltyjä rajoitteita.

Algoritmisia menetelmiä automatisoituun suunnitteluun (*automated planning*) [66,112]

Automatisoidun suunnittelun tavoitteena on löytää suunnitelma (plan), joka toteuttamalla päästään nykyisestä maailman tilasta tavoitetilaan suorittamalla suunnitelman määrittelemä sekvenssi atomisia operaatioita. Klassinen esimerkki automatisoiduista suunnitteluongelmista ovat ns. STRIPS planning -ongelmat, joille aktiivisesti kehitetään erilaisia eksakteja algoritmisia lähestymistapoja.

Laskennalliset näkökulmat argumentaatioon [35,54]

Abstrakti argumentaatio on tiedonesittämisparadigma, joka on mahdollistaa argumenttien keskinäisten suhteiden mallintamiseen ns. argumentaatioverkkoina. Yksi keskeinen argumentaatioverkkoihin liittyvä laskennallinen ongelma on päätellä mitkä annetun argumentaatioverkon argumentit ovat ristiriidattomia olettaessa tietyt argumentit tosiksi väitteiksi. Tämänkaltaisille, usein NP-koville argumentaatiotutkimuksesta kumpuaville suunnattuihin verkkoihin liittyville haku- ja optimointiongelmille on kehitetty useita erilaisia algoritmisia ratkaisumenetelmiä.

Kauppamatkustajan ongelma (*travelling salesman problem, TSP*) [12,90]

Kauppamatkustajan ongelma on yksi yleisimmistä NP-täydellisistä ongelmista. Sille tunnetaan erilaisia ratkaisuheuristiikkoja ja erikoistapauksille myös approksimointialgoritmeja.

Pienin virittävä puu (*minimum spanning tree*) [36], [142, luku 8]

Pienimmän virittävän puun laskenta on klassinen optimointiongelma. Siihen on löydetty myös uudempiä lähestymistapoja.

Avioliitto-ongelma (*stable marriage problem*) [19,61,62,83]

Avioliitto-ongelmassa pitää parittaa annetut naiset ja miehet toisilleen siten, etteivät he tee syrjähyppyjä, kun kaikkien mieltymykset tunnetaan. Tämän jo klassisen algoritmisen ongelman perustapaus ratkeaa polynomisessa ajassa, mutta laajennuksissa on törmättykin vaikeuksiin.

Verkon murtolukuväritys (*fractional graph coloring*) [71,85,105]

Murtolukuväritysongelmat ovat muunnoksia tunnetuista verkonväritysongelmista. Murtolukuväritystä voi soveltaa esimerkiksi tiedonsiirron ajoittamiseen langattomassa verkossa. Kaarivärityksen murtolukuversio ratkeaa polynomisessa ajassa, mutta solmuvärityksen approksimointikin on vaikeaa millä tahansa vakiosuhteella. Monille verkkojen luokille voidaan kuitenkin löytää approksimointialgoritmeja.

Steiner-puut (*Steiner tree*) [132,148]

Mitä nykyään tiedetään Steiner-puihin liittyvien optimointiongelmiä approksimoituvuudesta ja ei-approksimoituvuudesta?

Simuloitu jäähdytys (*Simulated annealing*) [1,95]

Simuloitu jäähdytys on yleiskäyttöinen satunnaisuuteen perustuva menetelmä, jota voidaan käyttää erilaisten optimointiongelmiä heuristiseen ratkaisemiseen.

Geneettiset algoritmit [68,78,138]

Geneettiset algoritmit ovat toinen satunnaisuuteen perustuva optimointimenetelmä.

Nopeasti sekoittuvat Markovin ketjut (*rapidly mixing Markov chains*) [31], [110, luvut 6.7 ja 11]

Nopeasti sekoittuvat Markovin ketjut tarjoavat keinon vaikeiden laskentaongelmien likimääräiseen ratkaisemiseen suurella todennäköisyydellä. Tyypillisesti algoritmit ovat yksinkertaisia, mutta niiden tehokkuuden todistaminen on mutkikasta.

Liittyy: Markovin prosesseja. *Kursseja:* Satunnaisalgoritmit.

Peitto- ja pakkaus-LP [64]

Peitto- ja pakkaustyyppisten LP-ongelmien sovelluksia ja approksimatiivista ratkaisemista.

Siirtostrategia (*shifting strategy*) [17, 77, 81]

Yleiskäyttöinen tekniikka, jota voidaan käyttää polynomi aikaisen approksimointiskeeman laatimiseen lukuisiin NP-koviin ongelmiin geometrisissa verkoissa (esim. yksikkökiekkoverkossa tai tasoverkossa).

2.5 Laskennallista geometriaa

Ohjaajan arvio: B. Kursseja: Geometriset menetelmät. *Konferensseja:* SoCG.

Leikkaavien janojen ongelma [18]

Monikulmioiden kolmiointialgoritmit [10]

Äärellisen tarkkuuden geometriaa [33, 39, 69, 70, 72, 76, 117]

Käytännön laskennassa geometrinen kappaleiden koordinaatteja ei käsitellä mielivaltaisen tarkkoja reaalilukuina. Miten koordinaatit voidaan pyöristää rajalliseen tarkkuuteen? Millaisia vaatimuksia pyöristämiselle asetetaan sovelluksissa? Millaisilla algoritmeilla nämä vaatimukset saadaan täytettyä?

Ydinjoukot ja geometriset approksimointialgoritmit [6, 73]

Monet laskennallisen geometrian ongelmat liittyvät isoihin pistejoukkoihin: on esimerkiksi löydettävä pienin pistejoukkoa ympäröivä pallo tai laatikko. Tällaisten ongelmien tehokkaaseen likimääräiseen ratkaisemiseen on olemassa kohtalaisen yleinen menetelmä: (1) etsitään pistejoukosta ns. ydinjoukko (*coreset*), joka on tarkasteltavan ongelman kannalta mahdollisimman edustava; (2) ratkaistaan ongelma tässä huomattavasti pienemmässä pistejoukossa. Mihin ongelmiin tämä lähestymistapa soveltuu ja mihin ei?

Konveksin peitteen yleistyksiä [55–57]

Ns. α -muodot ovat pistejoukon konveksin peitteen yleistyksiä.

Liittyy: Mahdollisesti varsin matemaattinen aihe.

Näkyvyysverkot [2, 38, 58, 67]

Monikulmion näkyvyysverkko (*visibility graph*) kuvaa, minkä nurkkapisteen välillä on esteetön näköyhteys. Jos monikulmio on annettu, näkyvyysverkon muodostaminen on suoraviivaista. Mutta jos on annettu verkko, voidaanko tunnistaa, onko se jonkin monikulmion näkyvyysverkko?

2.6 Logiikkaa; ohjelmointikielten syntaksia ja semantiikkaa

Ohjaajan arvio: C. Kursseja: Johdatus funktionaaliseen ohjelmointiin, Johdatus lambda-kalkyyliin, Spesifioinnin ja verifiointin perusteet, logiikan kurssit.

Rajoitetietokantojen kyselykielet (*constraint query languages*) [3, sivut 94–98], [21], [22], [103]

Rajoitetietokantojen kyselykielet käsittelevät sellaisia tietokantoja, joihin on talletettu informaatiota jostakin äärettömästä rakenteesta äärellisenä kokoelmana sitä rajoittavia lauseita. Esimerkiksi kolmiulotteinen avaruus on periaatteessa ääretön, mutta sen osia voidaan kuvailla sellaisilla lauseilla kuin ”kaikki ne pisteet joiden x -koordinaatti on positiivinen”. Mutta miten näin esitetystä informaatiosta vastataan käyttäjän kysymyksiin?

Liittyy: Tietokannat, logiikka.

Rajoitelogiikkaohjelmointi [42]

Rajoitelogiikkakielet.

Liittyy: Prolog-ohjelmointi.

Spesifikaatiot ovat väitteitä, ohjelmat niiden todistuksia [16, 43, 149]

Formaali looginen ohjelmankehitys.

Liittyy: Logiikkaa, spesifiointia ja verifiointia.

Ohjelmien aksiomaattinen semantiikka [75]

Heikoimman ennakkoehdon semantiikka, todistetusti oikeellisten ohjelmien kehittäminen.

Liittyy: Logiikkaa, spesifiointia ja verifiointia.

2.7 Hajautettuja algoritmeja

Ohjaajan arvio: C. Konferensseja: PODC, SPAA, DISC.

Hajautettu verkon väritys [41, 104], [44, luku 30.5]

Verkkoa ei voi värittää hajautetulla algoritmilla vakioajassa, ei edes siinä yksinkertaisessa erikoistapauksessa, että kyseessä on $n:n$ solmun rengas, jossa jokaisella solmulla on yksilöllinen tunniste. Sen sijaan renkaan värittämiseen on olemassa algoritmi, joka toimii ajassa $O(\log^* n)$; tämä on erittäin hitaasti kasvava funktio. Miten nämä tulokset yleistyvät muihin verkkoihin kuin renkaisiin? Mikä on yhteys muihin kombinatorisiin ongelmiin, esimerkiksi riippumattoman joukon etsimiseen?

Paikalliset algoritmit (*local algorithm*) [99, 100, 104, 111, 121–123, 147]

Paikallinen algoritmi on hajautettu algoritmi, joka ratkaisee ongelman vakioajassa riippumatta syötteenä olevan verkon koosta. Verkon kukin solmu suorittaa laskentaa itsenäisesti. Vakioajassa informaatiota voidaan kerätä vain vakioetäisyydeltä; kunkin solmun tekemä päätös on siis funktio syötteestä, joka oli alkutilanteessa saatavilla solmun lähiympäristössä enintään tietyllä etäisyydellä solmusta.

Itsestabiloivat algoritmit (*self-stabilising algorithm*) [15, 50–52, 136]

Hajautettu järjestelmä on itsestabiloiva, jos se palautuu äärellisessä ajassa kelvolliseen suoritukseen riippumatta siitä, mikä on järjestelmän alkutila.

Kursseja: Hajautettujen algoritmien seminaari syksyllä 2007.

Mustan aukon etsintää [59]

Musta aukko on verkon solmu, johon astuva agentti katoaa jälkiä jättämättä. Missä tapauksessa usean agentin joukkue voi paikantaa mustan aukon (uhraamalla osan jäsenistään)?

Johtajan valinta [27], [106, luku 3]

2.8 Tietoturva

Ohjaajan arvio: C. Kursseja: Tietoturva.

Julkisen avaimen salakirjoitus [131]

RSA-algoritmi yms.

Liittyy: Kryptografiaa, algebraa, lukuteoriaa.

Diffie-Hellman-avaimenvaihtoprotokolla [49]

Tiivistefunktiot kryptografiassa [126], [139, luku 12], [150]

Melko tekninen

Monen osapuolen protokollat [11], [29, luku 6], [146]

2.9 Muita aiheita

Ohjaajan arvio: B.

Mekanismisuunnittelu (*mechanism design*) [19, luku 7], [109], [115], [119]

Mekanismisuunnittelu on ”pelien suunnittelua” peliteoreettisessa mielessä. Toisin kuin tyypillisessä peliteorian ongelmassa, tässä ei asetuta itsekään ”pelaajan” asemaan ratkaisemaan optimaalista tapaa pelata peliä, vaan asetutaan ”keskusjohdon” asemaan suunnittelemaan pelin säännöt siten että itsekkäiden pelaajien omaa hyötyä maksimoivasta käytöksestä seuraa keskusjohdon haluama lopputulos. Esim. hyvin suunnitelluista pelisäännöistä voi seurata ettei itsekään pelaajan kannata petkuttaa tai vahingoittaa muita pelaajia.

Tunnettuna esimerkkinä mekanismisuunnittelusta voidaan mainita *Vickreyn huutokauppasääntö*, jota käytettiin esim. Googlen listausannissa.

Tietojenkäsittelytieteellä on liittymäkohtansa taloustieteisiin ja peliteoriaan. Mekanismisuunnittelua voi soveltaa myös Internetiin [119]. Toistuvien pelien tapauksessa peliteoreettisilla aiheilla on liittymäkohtia koneoppimiseen: TD-oppiminen, palauteoppiminen, ja Markovin prosessit.

Onpa maailma pieni -ilmiö (*small world phenomenon*) [96]

Analysoitaessa erilaisia verkostoja on havaittu, että useimmiten mitä tahansa kahta verkon solmua yhdistää lyhyt polku. Esimerkiksi lentoreiteissä, internetissä, geenisäätelyssä ja ihmisten tuttavuuspiireissä tämä ilmiö on havaittavissa. Mutta voidaanko tällaisten verkostojen syntyä selittää algoritmisesti?

Nopea Fourier- ja Möbius-muunnos [45, luku 30], [25, 92]

Kahden n -bittisen luvun tulon laskemiseen riittää lähes $n \log_2 n$ bittioperaatiota. Kuinka tämä on mahdollista? Entä, jos kerrotaan kaksi polynomia tai jos lasketaan kahden funktion konvoluutio?

Liittyy: Lineaarialgebraa, kombinatoriikkaa, konvoluutio.

Viitteet

- [1] Emile Aarts and Jan Korst. *Simulated Annealing and Boltzmann Machines: a Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley, 1989.
- [2] James Abello and Krishna Kumar. Visibility graphs and oriented matroids. *Discrete & Computational Geometry*, 28(4):449–465, 2002.
- [3] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [4] Dimitris Achlioptas, Paul Beame, and Michael S. O. Molloy. A sharp threshold in proof complexity yields lower bounds for satisfiability search. *Journal of Computer and System Sciences*, 68(2):238–268, 2004.
- [5] Tobias Achterberg. SCIP: solving constraint integer programs. *Math. Program. Comput.*, 1(1):1–41, 2009.
- [6] Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *Journal of the ACM*, 51(4):606–635, 2004.
- [7] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghawan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. 1998 ACM SIGMOD International Conference on Management of Data*, pages 94–105. ACM Press, 1998.
- [8] Srinivas M. Aji and Robert J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, 2000.
- [9] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

- [10] Nancy M. Amato, Michael T. Goodrich, and Edgar A. Ramos. A randomized algorithm for triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 26(2):245–265, 2001.
- [11] Yair Amir, Yongdae Kim, Cristina Nita-Rotaru, and Gene Tsudik. On the performance of group key agreement protocols. *ACM Transactions on Information and System Security*, 7(3):457–488, 2004.
- [12] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.
- [13] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
- [14] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.
- [15] Baruch Awerbuch and George Varghese. Distributed program checking: A paradigm for building self-stabilizing distributed protocols. In *Proc. 32nd Annual Symposium on Foundations of Computer Science*, pages 258–267. IEEE, 1991.
- [16] Roland C. Backhouse. Constructive type theory: A perspective from computing science. In Edsger W. Dijkstra, editor, *Formal Development of Programs and Proofs*, chapter 1, pages 1–32. Addison-Wesley, 1990.
- [17] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.
- [18] Ivan J. Balaban. An optimal algorithm for finding segments intersections. In *Proc. 11th Annual Symposium on Computational Geometry*, pages 211–219. ACM Press, 1995.
- [19] Michel Balinski and Guillaume Ratier. Of stable marriages and graphs, and strategy and polytopes. *SIAM Review*, 39(4):575–604, 1997.
- [20] Michael A. Bender, Erik D. Demaine, and Martin Farach-Colton. Cache-oblivious B-trees. *SIAM Journal on Computing*, 35(2):341–358, 2005.
- [21] Michael Benedikt and Leonid Libkin. Relational queries over interpreted structures. *Journal of the ACM*, 47(4):644–680, 2000.
- [22] Michael Benedikt and Leonid Libkin. Aggregate operators in constraint query languages. *Journal of Computer and System Sciences*, 64(3):628–654, 2002.
- [23] Yoshua Bengio. Markovian models for sequential data. *Neural Computing Surveys*, 2:129–162, 1999.
- [24] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.
- [25] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Fourier meets Möbius: Fast subset convolution. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC ’07, pages 67–74, New York, NY, USA, 2007. ACM.
- [26] Ravi B. Boppana and Michael Sipser. The complexity of finite functions. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity, pages 757–804. Elsevier, 1990.
- [27] Jacir Luiz Bordim, Yasuaki Ito, and Koji Nakano. Randomized leader election protocols in noisy radio networks with a single transceiver. In *Proc. 4th International Symposium on Parallel and Distributed Processing and Applications*, volume 4330 of *Lecture Notes in Computer Science*, pages 246–256. Springer-Verlag, 2006.

- [28] Allan Borodin. On relating time and space to size and depth. *SIAM Journal on Computing*, 6(4):733–744, 1977.
- [29] Colin Boyd and Anish Mathuria. *Protocols for Authentication and Key Establishment*. Springer-Verlag, 2003.
- [30] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.
- [31] Russ Bubley, Martin Dyer, Catherine Greenhill, and Mark Jerrum. On approximately counting colorings of small degree graphs. *SIAM Journal on Computing*, 29(2):387–400, 1999.
- [32] Michael Burrows and David Wheeler. A block sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, 1994.
- [33] Lilian Buzer. Optimal simplification of polygonal chain for rendering. In *Proc. 23rd Annual Symposium on Computational Geometry*, pages 168–174. ACM Press, 2007.
- [34] Marco Cadoli, Marco Schaerf, Andrea Giovanardi, and Massimo Giovanardi. An algorithm to evaluate quantified Boolean formulae and its experimental evaluation. *Journal of Automated Reasoning*, 28(2):101–142, 2002.
- [35] Günther Charwat, Wolfgang Dvorák, Sarah Alice Gaggl, Johannes Peter Wallner, and Stefan Woltan. Methods for solving reasoning problems in abstract argumentation - A survey. *Artif. Intell.*, 220:28–63, 2015.
- [36] Bernard Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *Journal of the ACM*, 47(6):1028–1047, 2000.
- [37] Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: Further observations and further improvements. *Journal of Algorithms*, 41(2):280–301, 2001.
- [38] Seung-Hak Choi, Sung Yong Shin, and Kyung-Yong Chwa. Characterizing and recognizing the visibility graph of a funnel-shaped polygon. *Algorithmica*, 14(1):27–51, 1995.
- [39] J. Chun, M. Korman, M. Nellenburg, and T. Tokuyama. Consistent digital rays. In *Proc. 24th Annual Symposium on Computational Geometry*, pages 355–364. ACM Press, 2008.
- [40] David Clark. *Compact PAT Trees*. PhD thesis, University of Waterloo, 1997.
- [41] Richard Cole and Uzi Vishkin. Deterministic coin tossing with applications to optimal parallel list ranking. *Information and Control*, 70(1):32–53, 1986.
- [42] Alain Colmerauer. An introduction to Prolog III. *Communications of the ACM*, 33(7):69–90, 1990.
- [43] Thierry Coquand, Bengt Nordström, Jan M. Smith, and Björn von Sydow. Type theory and programming. *Bulletin of the EATCS*, 52:203–228, 1994.
- [44] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, 1 edition, 1990.
- [45] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2 edition, 2001.
- [46] Graham Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.
- [47] Maxime Crochemore and Wojciech Rytter. *Jewels of Stringology*. World Scientific, Singapore, 2002.
- [48] Sanjoy Dasgupta and Philip M. Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569, 2005.
- [49] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654, September 1976.

- [50] Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Communications of the ACM*, 17(11):643–644, 1974.
- [51] Edsger W. Dijkstra. A belated proof of self-stabilization. *Distributed Computing*, 1(1):5–6, 1986.
- [52] Shlomi Dolev. *Self-Stabilization*. The MIT Press, 2000.
- [53] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 43–52. ACM Press, 1999.
- [54] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.
- [55] H. Edelsbrunner. Shape reconstruction with Delaunay complex. In *Proc. 3rd Latin American Symposium on Theoretical Informatics*, volume 1380 of *Lecture Notes in Computer Science*, pages 119–132. Springer-Verlag, 1998.
- [56] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983.
- [57] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.
- [58] H. J. Everett. *Visibility Graph Recognition*. PhD thesis, University of Toronto, 1990.
- [59] Paola Flocchini, David Ilcinkas, and Nicola Santoro. Ping pong in dangerous graphs: Optimal black hole search with pure tokens. In *Proc. 22nd International Symposium on Distributed Computing*, volume 5218 of *Lecture Notes in Computer Science*, pages 227–241. Springer-Verlag, 2008.
- [60] Matteo Frigo, Charles E. Leiserson, Harald Prokop, and Sridhar Ramachandran. Cache-oblivious algorithms. In *Proc. 40th Annual Symposium on Foundations of Computer Science*, pages 285–298. IEEE Computer Society Press, 1999.
- [61] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1968.
- [62] David Gale and Marilda Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11(3):223–232, 1985.
- [63] Alexander Gammerman and Vladimir Vovk. Kolmogorov complexity: Sources, theory and applications. *The Computer Journal*, 42(4):252–255, 1999.
- [64] Naveen Garg and Jochen Koenemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proc. 39th Annual Symposium on Foundations of Computer Science*, pages 300–309. IEEE Computer Society Press, 1998.
- [65] Ian P. Gent and Toby Walsh. Beyond NP: the QSAT phase transition. In *Proc. 16th National Conference on Artificial Intelligence*, pages 648–653. AAAI Press, July 1999.
- [66] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.
- [67] Subir Kumar Ghosh. On recognizing and characterizing visibility graphs of simple polygons. *Discrete & Computational Geometry*, 17(2):143–162, 1997.
- [68] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [69] M. Goodrich, L. J. Guibas, J. Hershberger, and P. Tanenbaum. Snap rounding line segments efficiently in two and three dimensions. In *Proc. 13th Annual ACM Symposium on Computational Geometry*, pages 284–293. ACM Press, 1997.

- [70] Leonidas Guibas and David Marimont. Rounding arrangements dynamically. *International Journal of Computational Geometry and Applications*, 8:157–176, 1998.
- [71] Bruce Hajek and Galen Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34(5):910–917, 1988.
- [72] D. Halperin and E. Packer. Iterated snap rounding. *Computational Geometry: Theory and Applications*, 23(2):209–225, 2002.
- [73] Sarel Har-Peled. No coresets, no cry. In *Proc. 24th International Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 3328 of *Lecture Notes in Computer Science*, pages 324–335. Springer-Verlag, 2005.
- [74] Alexander K. Hartmann and Martin Weigt. *Phase Transitions in Combinatorial Optimization Problems: Basics, Algorithms and Statistical Mechanics*. Wiley, 2005.
- [75] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.
- [76] J. D. Hobby. Practical segment intersection with finite precision output. *Computational Geometry: Theory and Applications*, 13(4):199–214, 1999.
- [77] Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32(1):130–136, 1985.
- [78] Abdollah Homaifar, Shanguchuan Guan, and Gunar E. Liepins. A new approach on the traveling salesman problem by genetic algorithms. In *Proc. 5th International Conference on Genetic Algorithms*, pages 460–466. Morgan Kaufmann, 1993.
- [79] Johan Håstad. Almost optimal lower bounds for small depth circuits. In Silvio Micali, editor, *Advances in Computing Research*, volume 5: Randomness and Computation, pages 143–170. JAI Press, 1989.
- [80] Huang, Shekhar, and Xiong. Discovering colocation patterns from spatial data sets: A general approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(12):1472–1485, 2004.
- [81] Harry B. Hunt, III, Madhav V. Marathe, Venkatesh Radhakrishnan, S. S. Ravi, Daniel J. Rosenkrantz, and Richard E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *Journal of Algorithms*, 26(2):238–274, 1998.
- [82] Frank Hutter, Lin Xu, Holger H. Hoos, and Kevin Leyton-Brown. Algorithm runtime prediction: Methods & evaluation. *Artif. Intell.*, 206:79–111, 2014.
- [83] Kazuo Iwama, David Manlove, Shuichi Miyazaki, and Yasufumi Morita. Stable marriage with incomplete lists and ties. In *Proc. 26th International Colloquium on Automata, Languages and Programming*, volume 1644 of *Lecture Notes in Computer Science*, pages 443–452. Springer-Verlag, 1999.
- [84] Guy Jacobson. *Succinct Static Data Structures*. PhD thesis, Carnegie Mellon University, 1989.
- [85] Klaus Jansen. Approximate strong separation with application in fractional graph coloring and preemptive scheduling. *Theoretical Computer Science*, 302(1–3):239–256, 2003.
- [86] Tao Jiang, Ming Li, and Paul M. B. Vitányi. New applications of the incompressibility method. *The Computer Journal*, 42(4):287–293, 1999.
- [87] Tao Jiang, Ming Li, and Paul M. B. Vitányi. Average-case analysis of algorithms using Kolmogorov complexity. *Journal of Computer Science and Technology*, 15(5):402–408, 2000.
- [88] David S. Johnson. The NP-completeness column: An ongoing guide. *Journal of Algorithms*, 5(3):433–447, 1984.

- [89] David S. Johnson. Catalog of complexity classes. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity, pages 67–162. Elsevier, 1990.
- [90] David S. Johnson. Local optimization and the traveling salesman problem. In *Proc. 17th International Colloquium on Automata, Languages and Programming*, volume 443 of *Lecture Notes in Computer Science*, pages 446–461. Springer-Verlag, 1990.
- [91] Leslie Pack Kaelbling, Michael L. Littman, and Andrew P. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [92] Robert Kennes. Computational aspects of the M?bius transformation of graphs. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):201–223, 1992.
- [93] Keogh, Lonardi, and Ratanamahatana. Towards parameter-free data mining. In *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 206–215. ACM Press, 2004.
- [94] Eamonn J. Keogh, Selina Chu, David Hart, and Michael J. Pazzani. An online algorithm for segmenting time series. In *Proc. 1st IEEE International Conference on Data Mining*, pages 289–296. IEEE Computer Society Press, 2001.
- [95] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [96] Jon M. Kleinberg. The small-world phenomenon: An algorithm perspective. In *Proc. 32nd Annual ACM Symposium on Theory of Computing*, pages 163–170. ACM Press, 2000.
- [97] Donald E. Knuth, James H. Morris, and Vaughan R. Pratt. Fast pattern matching in strings. *SIAM Journal on Computing*, 6(2):323–350, 1977.
- [98] Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Salojärvi, Jukka Honkela, Vesa Paatero, and Antti Saarela. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11(3):574–585, 2000.
- [99] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. The price of being near-sighted. In *Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 980–989. ACM Press, 2006.
- [100] Fabian Kuhn and Roger Wattenhofer. Constant-time distributed dominating set approximation. *Distributed Computing*, 17(4):303–310, 2005.
- [101] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proc. 1st IEEE International Conference on Data Mining*, pages 313–320. IEEE Computer Society Press, 2001.
- [102] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
- [103] Leonid Libkin. Query languages with arithmetic and constraint databases. *SIGACT News*, 30(4):41–50, 1999.
- [104] Nathan Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.
- [105] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.
- [106] Nancy Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.
- [107] Udi Manber and Eugene W. Myers. Suffix arrays: A new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.
- [108] Giovanni Manzini. An analysis of the Burrows–Wheeler transform. *Journal of the ACM*, 48(3):407–430, 2001.

- [109] Dov Monderer and Moshe Tennenholtz. Distributed games: From mechanisms to protocols. In *Proc. 16th National Conference on Artificial Intelligence*, pages 32–37. AAAI Press, July 1999.
- [110] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [111] Moni Naor and Larry Stockmeyer. What can be computed locally? *SIAM Journal on Computing*, 24(6):1259–1277, 1995.
- [112] Alexander Nareyek, Eugene C. Freuder, Robert Fourer, Enrico Giunchiglia, Robert P. Goldman, Henry A. Kautz, Jussi Rintanen, and Austin Tate. Constraints and AI planning. *IEEE Intelligent Systems*, 20(2):62–72, 2005.
- [113] Gonzalo Navarro and Veli M?kinen. Compressed full-text indexes. *ACM Computing Surveys*, 39(1):2, 2007.
- [114] Neill and Moore. Rapid detection of significant spatial clusters. In *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 256–265. ACM Press, 2004.
- [115] Noam Nisan. Algorithms for selfish agents. In *Proc. 16th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1563 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 1999.
- [116] Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. The effectiveness of Lloyd-type methods for the k -means problem. In *Proc. 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 165–176. IEEE, 2006.
- [117] Eli Packer. Iterated snap rounding with bounded drift. *Computational Geometry: Theory and Applications*, 40(3):231–251, 2008.
- [118] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
- [119] Christos H. Papadimitriou. Algorithms, games, and the Internet. In *Proc. 33rd Annual ACM Symposium on Theory of Computing*, pages 749–753. ACM Press, 2001.
- [120] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.
- [121] Christos H. Papadimitriou and Mihalis Yannakakis. On the value of information in distributed decision-making. In *Proc. 10th Annual ACM Symposium on Principles of Distributed Computing*, pages 61–64. ACM Press, 1991.
- [122] Christos H. Papadimitriou and Mihalis Yannakakis. Linear programming without the matrix. In *Proc. 25th Annual ACM Symposium on Theory of Computing*, pages 121–129. ACM Press, 1993.
- [123] Michal Parnas and Dana Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theoretical Computer Science*, 381(1–3):183–196, 2007.
- [124] Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. 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 1065–1075. SIAM, 2010.
- [125] Toniann Pitassi and Ran Raz. Regular resolution lower bounds for the weak pigeonhole principle. *Combinatorica*, 24(3):503–524, 2004.
- [126] Bart Preneel. *Analysis and Design of Cryptographic Hash Functions*. PhD thesis, Katholieke Universiteit Leuven, 1993.
- [127] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
- [128] Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

- [129] John R. Rice. The algorithm selection problem. *Advances in Computers*, 15:65–118, 1976.
- [130] Jussi Rintanen. Improvements to the evaluation of quantified boolean formulae. In *Proc. 16th International Joint Conference on Artificial Intelligence*, pages 1192–1197. Morgan Kaufmann, 1999.
- [131] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [132] Gabriel Robins and Alexander Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19(1):122–134, 2005.
- [133] Francesca Rossi, Peter van Beek, and Toby Walsh, editors. *Handbook of Constraint Programming*, volume 2 of *Foundations of Artificial Intelligence*. Elsevier, 2006.
- [134] V. David Sánchez. Frontiers of research in BSS/ICA. *Neurocomputing*, 49(1–4):7–23, 2002.
- [135] Bernhard Schölkopf and Alex J. Smola. A short introduction to learning with kernels. In Shoham Mendelson and Alex J. Smola, editors, *Advanced Lectures on Machine Learning, Machine Learning Summer School 2002*, volume 2600 of *Lecture Notes in Artificial Intelligence*, pages 41–64. Springer-Verlag, 2002.
- [136] Marco Schneider. Self-stabilization. *ACM Computing Surveys*, 25(1):45–67, 1993.
- [137] Daniel Dominic Sleator and Robert Endre Tarjan. Self-adjusting binary search trees. *Journal of the ACM*, 32(3):652–686, 1985.
- [138] M. Srinivas and Lalit M. Patnaik. Genetic algorithms: a survey. *Computer*, 27(6):17–26, 1994.
- [139] William Stallings. *Cryptography and Network Security: Principles and Practice*. Prentice Hall, 3 edition, 2003.
- [140] Richard Edwin Stearns. Deterministic versus nondeterministic time and lower bound problems. *Journal of the ACM*, 50(1):91–95, 2003.
- [141] Richard Edwin Stearns and Harry B. Hunt. An algebraic model for combinatorial problems. *SIAM Journal on Computing*, 25(2):448–476, 1996.
- [142] Robert Endre Tarjan. *Data Structures and Network Algorithms*. Society for Industrial and Applied Mathematics, 1983.
- [143] Robert Endre Tarjan and Jan van Leeuwen. Worst-case analysis of set union algorithms. *Journal of the ACM*, 31(2):245–281, 1984.
- [144] Gerald Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.
- [145] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
- [146] Wade Trappe, Yuke Wang, and K. J. Ray Liu. Resource-aware conference key establishment for heterogeneous networks. *IEEE/ACM Transactions on Networking*, 13(1):134–146, 2005.
- [147] Jorge Urrutia. Local solutions for global problems in wireless networks. *Journal of Discrete Algorithms*, 5(3):395–407, 2007.
- [148] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2003.
- [149] Philip Wadler. Old ideas form the basis of advancements in functional programming, December 2000.
- [150] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In *Proc. 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer-Verlag, 2005.
- [151] Jeffery Westbrook and Robert Endre Tarjan. Amortized analysis of algorithms for set union with backtracking. *SIAM Journal on Computing*, 18(1):1–11, 1989.