



Kokeellisesta algoritmitutkimuksesta Tutkimusmenetelmät-kurssi, s-2004

Pekka Kilpeläinen

Kuopion yliopisto

Tietojenkäsittelytieteen laitos

Hyvä kokeellinen algoritmitutkimus?

Päälähde: D.S. Johnson: A theoretician's guide to the experimental analysis of algorithms. s. 215–250 teoksessa Goldwasser, Johnson & McGeoch (toim.), *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Challenges*. AMS, 2002. Löytyy David S. Johnsonin kotisivun kautta.

Ks. myös luku 11 (Experimentation) teoksessa J. Zobel: *Writing for Computer Science, Second Edition*. Springer-Verlag, 2004.

Miksi kokeellista algoritmitutkimusta?

Kokeilla on vaikea tuottaa yleisiä tuloksia, ja mahdoton todistaa niitä varmuudella. (Vrt. testaaminen) Toisaalta ...

- ⑥ teoreettiset tarkastelut eivät anna täyttä kuvaa algoritmin tehokkuudesta
- ⑥ kokeet voivat vahvistaa teoreettisia tuloksia ja herättää uusia kysymyksiä
- ⑥ kokeilu tärkeä silta teoriasta käytännön sovelluksiin

↪ Viime vuosina noussut laatulehtiä ja -kokouksia, joiden nimissä ja aihepiirinä *Experimentation*, *Experimental Algorithms* ja *Algorithm Engineering*

Kokeellisen algoritmitutkimuksen julkaisut

- ⑥ **sovellusartikkeli** algoritmin käytöstä jonkin sovellusalueen ongelman ratkaisuun
- ⑥ “**ravikilpa-artikkeli**” (horse race paper) algoritmin (toteutuksen) paremmuudesta suhteessa aiempiin
- ⑥ **kokeellinen analyysi**, tavoitteena *ymmärtää* kiinnostavien algoritmisten ideoitten toiminta, heikkoudet sekä vahvuudet käytännössä
- ⑥ **kokeellinen keskimääräisanalyysi**, tavoitteena tuottaa oletuksia keskimääräisestä käyttäytymisestä

Artikkelit usein näiden tyyppien yhdistelmiä

Hyvän kokeellisen työn periaatteet

1. Uutuusarvoiset kokeet
2. Kytkeä aiempaan tutkimukseen
3. Yleisiä johtopäätöksiä tukevat koejärjestelyt
4. Tehokkaat ja tulokselliset koejärjestelyt
5. Kohtuullisen tehokkaat toteutukset
6. Toistettavuus ja vertailtavuus
7. Kattava raportointi
8. Perusteltujen johtopäätösten teko ja selittäminen
9. Tulosten informatiivinen esittäminen

Periaatteiden tausta-ajatus

Tavoite: Tuottaa

- ⑥ uutta ja kiinnostavaa/hyödyllistä tietoa sellaisilla tavoilla että
- ⑥ tutkija vakuuttuu oikeellisuudesta siten esitettynä, että
- ⑥ lukija voi varmistua oikeellisuudesta

1. Uutuusarvoiset kokeet

Julkaistavien tulosten tulisi olla uusia ja kiinnostavia

- ⑥ tarkasteltavat menetelmät (ainakin mahdollisesti) käytännössä hyödyllisiä

Entä jos menetelmä osoitautuu “dominoiduksi”, s.o. kaikissa tilanteissa tehottomammaksi kuin tunnettu vertailumenetelmä?

Tyypillinen virhe: Puutteellinen taustatutkimus: ei perehdytty aiempaan tutkimukseen. *Ei hyvä!*

Negatiiviset koetulokset

Positiiviset tulokset halutumpia

Tulokset dominoidustakin algoritmista mahdollisesti, jos

- ⑥ se on käytössä ja tehokkaammat vertailukohtat epäkäytännöllisiä, tai tulos yllättävä
- ⑥ se on metodisesti kiinnostava. (Esim. uusi yleinen näkökulma sovellettaviin tieto- tai kontrollirakenteisiin)
- ⑥ sen huonoa käyttäytymistä ei aiemmin todistettu. (Ehkä osaksi laajempaa tarkastelua)

Negatiivinen tulos perusteltava huolellisesti ja/tai selitettävä

Huom: Kaksinkertainenkaan aikaero ei välttämättä merkittävä!

Riski: Väärien kysymysten testaaminen

- ⑥ Älä rajoitu yhteen/pariin testitapaukseen — Testaa mieluummin systemaattisesti (erityyppisten) tapausten valikoimalla
 - △ Testitapaukset \sim otos äärettömästä perusjoukosta (vrt. esim. opiskelijoiden keskipituuden arviointi)
- ⑥ Älä suorita kattavia testejä *liian varhain*
 - △ Tutkiskele (explore) ennen toteutuksen viilausta ja testien suunnittelua (**Mitä mitaan? Miten? Millä aineistoilla?**)

“Think before you compute”

Johnsonin lähestymistapa

1. Noin puolet varatusta ajasta generoiden runsaasti testidataa ja etsien säännönmukaisuuksia ja poikkeavuuksia
2. Kiinnitä kysymykset; Viimeistele toteutus; Suunnittele ja suorita kattavat kokeilut
3. Analysoi data; Palaa edelliseen vaiheeseen, jos vastauksia ei saatu tai syntyi uusia kysymyksiä

Kuitenkin: Lopeta kokeet joskus!

Riski: Turha viilailu: kokeiden tarkoitus vain tutkittavien ilmiöiden havaitseminen ja mittaaminen

Muita uutuusarvon tekijöitä

Tulosten yleisyys, merkittävyys ja uskottavuus

- ⑥ Miksi tulokset ovat sitä mitä ovat?
 - △ Esim. keskeisten operaatioiden suorituskertatilastot, koodin profilointi
- ⑥ Muodostuuko laajempaa näkemystä esim. asympotoottisesta käyttäytymistä tai likimääräismenetelmän tarkkuudesta?

Toteutus vs. algoritmi

Kokeet antavat suoraa tietoa vain **toteutuksista**

Kysymyksiä yleisempien tulosten johtamiseksi
(algoritmeista):

1. Toteutuksen yksityiskohtien, parametrien, heuristiikkojen ja tietorakennevalintojen vaikutus suoritusaikaan?
2. Skaalautuvuus tapauksen koon suhteen?
Riippuvuus tapauksen tyypistä?
3. Suoritusaikoja selittävät operaatioiden lukumäärät?
Pullonkaulat? Poikkeamat pahimman tapauksen analyysistä?

Toteutus vs. algoritmi (2)

4. Arkkitehtuurin vaikutus? Suoritusaikojen ennustettavuus?
5. Suoritusaikojen suhde (tehokkaimpaan) vertailualgoritmiin? E.m. tekijöiden vaikutus suhteeseen?
6. Uusien tapausluokkien vaikutus tunnettuihin algoritmeihin?

Riski: Ajautuminen *algoritmin* tutkimisesta tietyillä tapauksilla näiden *tapausten* ominaisuuksien tutkiskeluun

2. Kytkeä aiempaan tutkimukseen

Perehdy ensin olemassaolevaan tutkimuskirjallisuuteen

- turhien kokeiden välttäminen
- avoimet kysymykset
- vertailumenetelmät

Pyri hankkimaan vertailualgoritmien toteutukset, tai laadi *vertailukelpoiset* toteutukset itse

- ⑥ Vertailukohta uutta ongelmaa ratkaisevalle algoritmille?
Esim. *järkevä*lle tietojenkäsittelijälle ensimmäisenä mieleen tuleva menetelmä

Vertailtavuudesta lisää myöhemmin

3. Yleisiä johtopäätöksiä tukevat koejärjestelyt

Kahdenlaisia testitapauksia: todellisia/realistisia ja satunnaisia

- ⑥ Huom: satunnaisuus kontrolloitua, realistisia tapauksia vastaavaa

Satunnaistapaukset mahdollistavat skaalautuvuuden arviointia

Todellisista tapauksista johtopäätöksiä soveltuvuudesta eri kaltaisiin aineistoihin

Tyyppien yhdistelmät suositeltavia

Riski: Liian pienillä tapauksilla testaaminen

- ⑥ Alle 0,1 s suoritusaikojen merkittävyys ja luotettavuus?

4. Tehokkaat ja tulokselliset koejärjestelyt

Minimoi satunnaisvaihtelua. Esim. sovelta vertailualgoritmeja *samoihin* satunnaistapauksiin (vs. omat satunnaistapaukset kullekin erikseen)

Helpota tulosdatan hallinnointia

- ⑥ Organisoiki hakemistoiksi (README-tiedostot!)
- ⑥ Generoiki tulostiedostoihin relevantit tiedot: mittaustulokset, algoritmin nimi ja versio, laitteisto, päiväys, testitapaus, parametrit, ...
 - △ Nimeä tulostiedostojen kentät

Dataa voi tarvita vuosien kuluttua, jolloin sen tulkinta vaikeaa ja uudelleen generointi mahdotonta → Säilytä tulosdata! (Huom: Yliopiston työntekijällä ei tekijänoikeutta tulosdataan)

5. Kohtuullisen tehokkaat toteutukset

Jotta toteutuksien suoritusaikoihin perustuvat johtopäätökset olisivat luotettavia

“Algoritmimme olisi verrokkaa tehokkaampi, jos sitä vain optimoitsiin vastaavasti” ei vakuuta

Huom: *Kohtuullisen* tehokkaat — viimeisten 10 % irtiotto ei tarpeen

6. Toistettavuus ja vertailtavuus

Koetulosten oikeellisuus pitää pystyä tarkastamaan toistamalla kokeet

- ⑥ Samoja *mittaustuloksia* ei yleensä saa, mutta *johtopäätösten* oltava testattavia
- ⑥ Tutkijan vakuututtava johtopäätösten oikeellisuudesta ja riippumattomuudesta koejärjestelyn yksityiskohdista
- ⑥ Koejärjestelyn merkittävät tekijät kuvattava: algoritmit, testitapaukset, suoritusympäristö (prosessori ja sen nopeus, keskusmuistin koko, KJ, toteutuskieli ja kääntäjä; *versiot*, optimointi)
 - △ tuloksiin vaikuttavien parametriasetusten vaikutus!

Hyvä saattaa toteutukset ja aineistot saataville (Webiin)

Vertailtavuus

Jatkotutkimukset melko varmasti eri suoritusympäristössä
Miten suhteuttaa suoritusajoja (jos ei voi toistaa aiempia kokeita)?

- ⑥ Prosessorin nopeus epäluotettava vertailukohta
- ⑥ koestinohjelmien (benchmark programs) avulla
 - △ koodi, joka käsittelee ja suorittaa algoritmeille keskeisiä tietorakenteita ja operaatioita
 - △ → vertailtavuus likimain tarkkuudella $0,5 \times \dots 2 \times$ (*vain!*)

7. Kattava raportointi

Tuloksia on tulkittava ja yksinkertaistettava, *muttei liikaa*
Johtopäätöksien takana olevat mittaustulokset esitettävä

- ⑥ Edustavat näytteet samankaltaisista luokista (muista maininta)
- ⑥ (Keskiarvo)taulukot, ainakin liitteiksi
 - △ Monenko tapauksen/suorituksen keskiarvo?
 - △ Tulosten jakauma? Normaalijakaumasta keskihajonta, erikoisemmasta graafinen esitys

Riski: keskiarvojen esittäminen mittaustarkkuutta suuremmalla tarkkuudella (↔ väärät johtopäätökset)

Kattava raportointi (2)



Poikkeavuudet ja (toistettavat) omituisuudet mainittava

- ⑥ ja selitettävä (tai todettava selittämättömiksi)
- ⑥ Mainitsematta jätetty selittämättöm poikkeavuus mahdollinen ladonta- tai toteutusvirhe

Kerro myös täydet todelliset suoritusajat

- menetelmän käytännöllisyys?
- tehokkuuden merkittävyys? (myös kokonaissuorituksen osana)
- kokeiden uusijan resurssitarve

8. Perusteltujen johtopäätösten teko ja selittäminen

Kokeilujen pitäisi tuottaa ja tukea johtopäätöksiä

Tyypillinen virhe: Tulkitsematon data

- ⑥ esittele havaittavat säännönmukaisuudet
- ⑥ pyri yleisempiin (datan tukemiin) konjektuureihin

Tyypillinen virhe: Perusteettomat johtopäätökset

- ⑥ Esim: asymptoottiset johtopäätökset pienistä tapauksista

Profilointi (tai koodin manuaalinen instrumentointi)
hyödyllinen apuväline

9. Tulosten informatiivinen esittäminen



Ks. esim. luku 6 (Graphs, Figures, and Tables) Zobelin kirjassa

Esityksen tuettava johtopäätöksiä

- ⑥ visualisointi hyödyllistä sekä havainnollistamiseen että havainnointiin

Korosta datasta havaittavia trendejä, eroavuuksia jne.

Tyypillinen virhe: Taulukot ilman graafista esitystä (yhteenvedo, johtopäätökset?)

- ⑥ ja huonosti jäsenneilyt ja/tai puutteelliset taulukot
 - △ järjestys, johdetut tiedot (esim. suhdeluvut)
 - △ alkioden merkitys ja mittayksiköt

Tulosten informatiivinen esittäminen

(2)

Tyypillinen virhe: Graafinen esitys ilman taulukoita
(yksityiskohtaiset tulokset?)

Tyypillinen virhe: Tehottomat kuvat

Esim. allaolevien suoritusaikojen esittäminen

	100	316	1,000	3,162	10,000	31,623	100,000	316,227	10^6
A	0.00	0.02	0.08	0.29	1.05	5.46	23.0	89.6	377
B	0.00	0.03	0.11	0.35	1.38	6.50	30.6	173.3	669
C	0.01	0.06	0.21	0.71	2.79	10.98	42.7	329.5	1253
D	0.02	0.09	0.43	1.64	6.98	37.51	192.4	789.7	5465
E	0.03	0.14	0.57	2.14	10.42	55.36	369.4	5775.0	33414

(gnuplot-demonstraatio)

Suorituskokeiluesimerkki

(Alustavaa tutkiskelua (exploration))

- ⑥ Oppikirja: “Lisäyslajittelu (vaikkakin $\Theta(n^2)$) on *pienillä* taulukoilla pikalajittelua tehokkaampi”
→ Kuinka pienillä? Alle 10, 100, 10000 alkiota?
- ⑥ Ongelma: pienten suoritusaikojen epätarkkuus
- ⑥ Kyse *tapauksen koon* vaikutuksesta *suhteelliseen* suoritusaikaan → Voitiin mitata vertailualgoritmien suorituksia (+ samoja alustusoperaatioita) “riittävän usein” (100,000 krt) toistettuina