Računanje z DNA

Iztok Pižorn

14. avgust 2007

1 Problem Hamiltonovih usmerjenih poti

Problem Hamiltonovih usmerjenih poti (Hamiltonian Path Problem: HPP) spada med algoritmično najtežje – NP-polne[1] probleme.

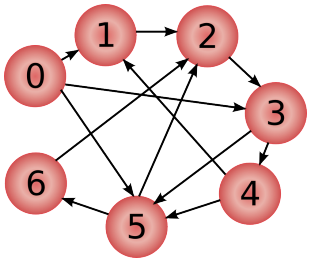
Vzemimo *usmerjen* graf z $n+1$ točkami (verteksi) in izberimo začetno (0) in končno točko (n). Problem se glasi: *ali obstaja pot iz* (0) *v* (n)*, ki vsako točko* (i) *obide natančno enkrat?*

Problem HPP iz matematične teorije grafov je v teoretičnem računalništvu znan tudi kot problem trgovskega potnika (*Traveling Salesman Problem*). Če nam kdo prišepne morebitno rešitev, lahko zlahka preverimo, če je le-ta prava; vendar število eventuelnih rešitev eksponentno narašča s številom mest in razen preverjanja (z zrnom soli) vseh mogočih kombinacij ni druge možnosti reševanja. Obstaja tudi naslednja nedeterministična metoda reševanja:

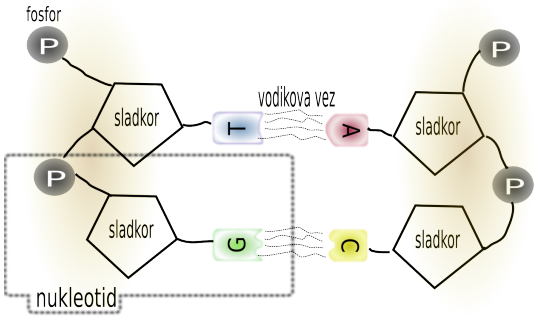
1. generiraj slučajne poti po grafu
2. obdrži le poti, ki se začnejo v (0) in končajo v (n)
3. obdrži le poti, ki imajo $n+1$ točk
4. obdrži le poti, ki vsako točko vsebujejo vsaj enkrat
5. če kakšna pot ostane kljub zgornjemu izločanju, je to rešitev problema in odgovori z 'da', sicer rešitve verjetno[2] ni in odgovori z 'ne'.

[1] V teoretičnem računalništvu so NP-polni problemi najtežji, saj ni algoritma, ki bi zanje poiskal rešitev v času, ki največ polinomsko narašča z dimenzijo problema.

[2] Nedeterministična metoda lahko gotovo odgovori le z 'da', z 'ne' pa le z določeno verjetnostjo, ki se s številom poskusov bliža k 1.



Slika 1: Usmerjen graf s 7 točkami, kakršnega je obravnaval L. M. Adleman. Rešitev je 'da', in sicer: $0 \to 1 \to 2 \to 3 \to 4 \to 5 \to 6$.



Slika 2: Nukleotidi se med sabo razlikujejo le v bazi. Med sabo se povezujejo v vlakna, ki hibridizirajo s komplementarnimi vlakni.

Problem HPP so leta 1994 rešili tudi Leonard M. Adleman in sodelavci, in sicer s to posebnostjo, da je reševanje potekalo na molekularnem nivoju z DNA. Čeprav se prvi hip zdi težko verjetno, da bi molekule reševale računalniške probleme, predstava kmalu postane jasnejša, če si veliko število molekul DNA predstavljamo kot realizacijo vseh mogočih itinerarijev, pri čemer ekvivalente računalniških bitov predstavljajo štiri vrste nukleotidov. Sedaj je potrebno le še odstraniti neustrezne in preveriti, če kakšna molekula preživi vse teste – potem je rešitev 'DA'.

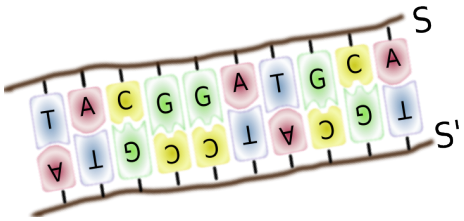
2 Adlemanov eksperiment

Po uvodni razlagi koncepta računanja DNA, si sedaj podrobneje oglejmo vse njegove postopke in ključne konceptualne elemente.

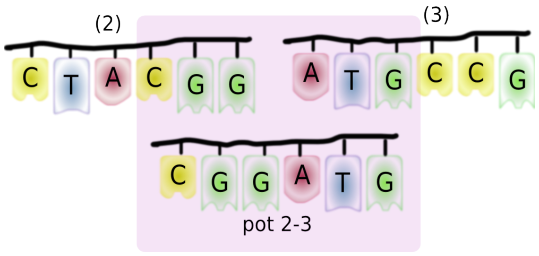
Najprej si oglejmo bistvene sestavine DNA. Nukleotid je organska struktura s tremi komponentami (Slika 2): fosfatno skupino, saharidno skupino in dušikovo skupino. Slednjo imenujemo *baza*, saj se nukleotidi med sabo razlikujejo le po tej. Baza nastopa v štirih različicah: adenin (A), timin (T), citozin (C) in guanin (G), ki predstavljajo štiri različne molekule s približno 15 atomi (ogljik, dušik, kisik in vodik). Celoten nukleotid je sestavljen iz približno 50 atomov. Nukleotidi se med sabo povezujejo v vlakna, ki hibridizirajo s sebi komplementarnimi vlakni. Komplementarna vlakna so takšna, kjer je vsak nukleotid zamenjan z njegovim partnerjem, torej T namesto A, A namesto T, C namesto G in G namesto C.

2.1 Korak 1: Kodiranje usmerjenega grafa

Prof. Adleman je pri kodiranju podatkov namesto računalniških bitov uporabil različne tipe nukleotidov, tako da je verteksu priredil izbrani niz 20 nukleotidov, povezanih v vlakno (kratka vlakna nukleotidov imenujemo *oligonukleotidi*). Sinteza oligonukleotidov dandanes ne predstavlja tehnološko zahtevnega postopka; obstajajo že komercialni sintetizatorji nukleotidov. Sinteza poteka v trdnem stanju (ne v raztopini). Pričnemo z enim nukleotidom, vezanim na steklo, na katerega polijemo raztopino drugega nukleotida, ki se veže na prvega. Preostanek raztopine speremo in ostane nam 2-mer. Postopek ponavljamo do želene



Slika 3: Nizu nukleotidov S, ki tvori vlakno, ustreza komplementarni niz S'. Obe vlakni hibridizirata in z zvijanjem tvorita dvojno vijačnico.



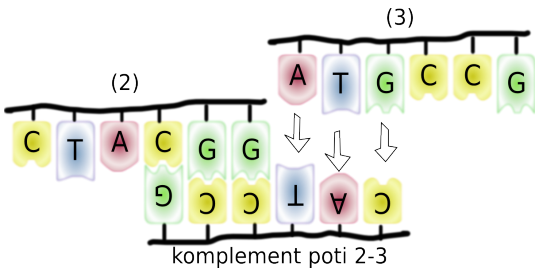
Slika 4: Shematični prikaz kodiranja verteksov in poti z nizi nukleotidov. Zaradi preglednosti so oligonukleotidi le dolžine 6 namesto 20.

dolžine.

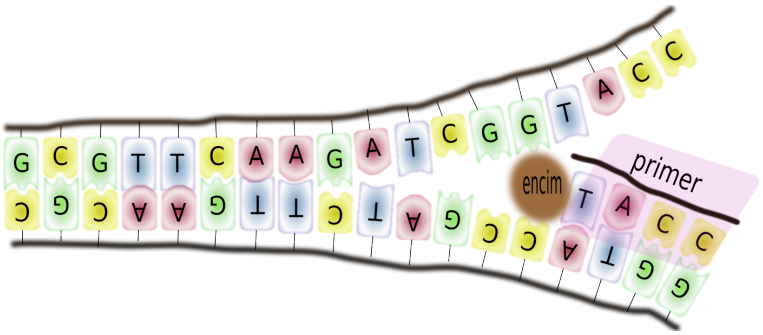
Zamisel, kako predstaviti poti med mesti, je značilna prav za DNA, saj izkorišča lastnost, da vsakemu zaporedju nukleotidov ustreza le eno komplementarno zaporedje. Ko vlakno z zaporedjem S pride v stik z vlaknom s komplementarnim zaporedjem S', vlakni hibridizirata (Slika 3) in z uvijanjem tvorita dvojno vijačnico. Če raztopino segrejemo, se vijačnica spet razvije in razklopi. Rešitev implementacije usmerjene poti med dvema verteksoma se torej ponuja na dlani. Vzemimo zadnjih deset nukleotidov prvega oligonukleotida in prvih deset nukleotidov drugega oligonukleotida in ju zaporedno združimo; rezultat enolično določa usmerjeno pot med tema verteksoma (Slika 4). [3]

Po zgoraj opisanem načinu generiramo vlakna nizov nukleotidov za vsa mesta in vse poti, ki jih povezujejo, pri čemer pripravimo veliko količino vsake vrste

[3]Tehnična podrobnost je, da za pot 0-1 vzamemo (ne le drugo polovico, temveč) celoten niz (0) in prvo polovico (1); podobno storimo za pot med $(n-1)$ in (n).



Slika 5: S hibridizacijo med komplementi poti in točkami se tvorijo itinerariji.



Slika 6: Encim polimeraza omogoči kopiranje DNA molekul. Pri ustrezni temperaturi se vijačnica razvije, encim pa vlakno dopolni s komplementarnim delom, začenši s kalupom (*primer*).

vlakna. Vse pripravljene molekule nato zmešamo skupaj v eno epruveto, skupaj z DNA ligazinom, soljo in dodamo vodo. S hibridizacijo (ki jo dosežemo z ohlajanjem) se tvorijo ne le povezave med dvema verteksoma, temveč dolge molekule DNA, ki predstavljajo itinerarije različnih dolžin med dvema slučajnima točkama. Postopek tvorbe itinerarijev kemijsko poteka s pomočjo DNA ligase in adenazin trifosfata (ATP). Pri dovolj velikem številu molekul pričakujemo, da bo med množico dolgih molekul DNA tudi pravilna rešitev problema, če le-ta obstaja. Za občutek velikosti navedimo, da je bilo v Adlemanovem eksperimentu te raztopine vseh mogočih DNA molekul vsega za petdesetino čajne žličke.

Hibridizacija, torej tvorba itinerarijev, poteka vzporedno v času, kar je poglavitna prednost računanja na molekularnem nivoju. Pri klasičnem računanju paralelizacija namreč kvečjemu zveča skupni procesorski čas, pri računanju z DNA pa je le-ta intrinzična lastnost molekularnih sistemov.

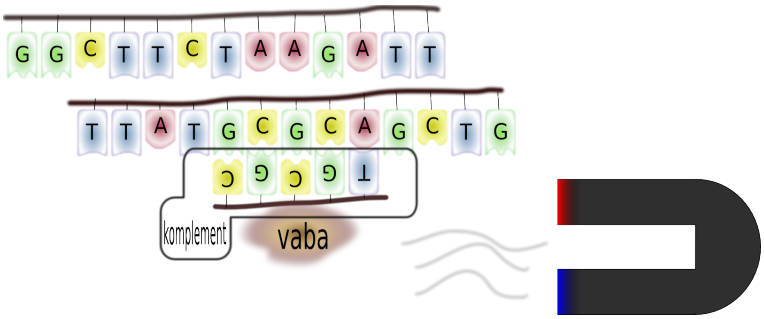
Pri tvorbi itinerarijev lahko nastajajo napake. Včasih DNA encimi pretrgajo vlakno ali vstavijo napačen nukleotid; zaenkrat nas tovrstne težave ne bodo zanimale. Omenimo pa, da dvojna vijačnica predstavlja redundančni zapis informacije. Če v enem izmed vlaken v vijačnici pride do napake, lahko encimi popravijo prizadeti del, saj je njegova konfiguracija razvidna iz njemu ustreznega dela v komplementarnem vlaknu.

	DNA računalniki	Silicijevi računalniki
Pomnilnik	nukleinske kisline	polprevodniki
Operatorji	biokemijske reakcije	logična vrata
Delovanje	paralelno	sekvenčno
Način delovanja	stohastičen	deterministčen

2.2 Korak 2: Ločevanje itinerarijev s pravilnima končnima točkama

Po končanem generiranju itinerarijev se v epruveti nahajajo DNA molekule, ki predstavljajo vse mogoče poti; med temi je večina napačnih: nekatere se začnejo ali v napačnih točkah, druge so napačnih dolžin, tretje ne povezujejo vseh točk.

Prvi korak v selekciji je ta, da obdržimo le tiste, ki se pričnejo in končajo v pravilnih točkah. Pri tem nam pomaga poseben encim znan kot polimeraza.



Slika 7: Ločevanje vlaken DNA, ki vsebujejo izbrano zaporedje nukleotidov.

Ta encim selektivno kopira DNA molekule, začenši s kalupom (angl. *primer*, to je komplement nekega izbranega oligonukleotida). Molekule DNA najprej segrejemo, da se dvojne vijačnice razvijejo v dve enojni vlakni. Encim DNA polimeraza, ki ima za kalup začetni verteks, dopolni eno izmed vijačnic razvite DNA molekule, tako da začne pri začetnem mestu in v točno določeni smeri nadaljuje po celotni molekuli (Slika 6); polimeraza s komplementarnim kalupom pa dopolni komplementarno vlakno. Tako iz ene molekule DNA, ki se prične s pravilnim oligonukleotidom, dobimo dve enaki, medtem ko preostale ostanejo nepodvojene.

Princip uporabimo pri postopku računanja, tako da uporabimo polimerazo s kalupoma (0) in komplementom $(n)'$. Pri tem se bodo podvojile natanko tiste vijačnice DNA, ki se pričnejo in končajo s pravilnima točkama, torej z (0) in (n), ostale pa bodo ostale bodisi razvite v vlakna bodisi bo dopolnjeno le eno vlakno. Po večkrat ponovljenem postopku (nekaj desetkrat) bo število molekul s pravima koncema naraslo eksponentno. Postopek je odkril Mullis in se imenuje PCR (*Polymerase Chain Reaction*).

2.3 Korak 3: Ločevanje molekul želenih dolžin

Po predhodni reakciji s polimerazo se v epruveti nahajo le itinerariji s pravilnima končnima točkama, vendar z nedorečenim notranjim delom poti. Tretji korak pri reševanju problema je izmed vseh preostalih poti obdržati tiste, ki vsebujejo natanko $n + 1$ točk.

Način ločevanja molekul želenih dolžin je konceptualno preprost: molekule potopimo v *agarose* gel in vklopimo električno polje, pod vplivom katerega se molekule pričnejo gibati. Velike molekule se teže umikajo oviram v gelu, zato potujejo počasneje. Čez nekaj časa torej dobimo pasove molekul DNA istih dolžin oziroma mas. Sedaj le še poiščemo pas (to nalogo opravi senzor s pripadajočo elektroniko), ki ustreza $n + 1$ točkam, in ga ločimo od ostalih.

2.4 Korak 4: Ali pot vsebuje vsa mesta?

Do končne rešitve preostane še selektivni test: itinerariji pravilnih dolžin s pravilnima koncema so rešitve HPP, če vsebujejo vsa mesta v grafu. Metoda, s katero ločimo vlakna DNA, ki vsebujejo izbrano zaporedje nukleotidov, od preostalih, se imenuje postopek separacije (*affinity separation*).

Na biotin-avidin magnetne vabe vežemo komplement izbranega zaporedja nukleotidov. V raztopini se nanj vežejo takšna vlakna DNA, ki lahko hibridizirajo z vabo, torej prav tista, ki vsebujejo izbrano zaporedje; ostala vlakna ostanejo nevezana. Čez čas vklopimo magnetno polje, pod vplivom katerega se začnejo magnetne vabe gibati, z njimi pa tudi nanje vezana vlakna DNA. Na ta način iz množice DNA molekul "izvlečemo" (Slika 7) natančno tiste, ki vsebujejo zahtevano zaporedje, preostanek pa odplaknemo.

Postopek opravimo za vsak verteks v grafu posebej, tako da na vabo vežemo njegovo komplementarno zaporedje. Ta računski korak, kljub konceptualni enostavnosti, zahteva največ eksperimentalnega časa za izvedbo. Kar ostane na koncu, če kaj, je rešitev HPP; v nasprotnem primeru pa rešitve verjetno ni.

2.5 Obravnava časovne zahtevnosti

Za oceno časovne zahtevnosti metode si oglejmo vsak posamezni korak posebej.

Prvi korak, molekularno kodiranje usmerjenega grafa, zahteva pripravo $n+1$ nukleotidnih nizov, ki predstavljajo vertekse. Časovna zahtevnost torej narašča linearno z dimenzijo n, potrebna sredstva (število DNA molekul) pa eksponentno z n.

Drugi in tretji korak nista eksplicitno odvisna od števila verteksov v grafu. Implicitna časovna odvisnost izhaja najprej iz števila molekul v epruveti, ki podaljša čas poteka kemijskih reakcij; točna odvisnost še ni dobro raziskana. Drugi korak zahteva tudi določeno število iteracij, da se število molekul s pravilnima koncema dovolj poveča v primerjavi z ostalimi. Število iteracij tako s številom verteksov narašča polinomsko.

Četrti korak je spet eksplicitno odvisen od števila verteksov, tako da čas izvedbe koraka linearno narašča z dimenzijo n.

Povzamemo, da čas izvedbe narašča le linearno s številom verteksov, kar je velika prednost v primerjavi s sekvenčnim računalnikom, saj DNA računanje intenzivno izkorišča paralelizacijo pri računanju. Seveda pa to zahteva eksponentno naraščanje *sredstev*, potrebnih za izvedbo eksperimenta, to je število molekul DNA. Skupna časovna zahtevnost je torej še vedno eksponentna v številu verteksov, kar samo po sebi ne predstavlja izboljšane *algoritemske* zahtevnosti. Za razloček, kvantni računalniki obetajo prav to, čeprav zaenkrat le v teoriji.

Prednost DNA računanja je v tem, da so eksponentno naraščajoča sredstva relativno mnogo "cenejša" (dosegljivejša) kot pri običajnih računalnikih. Prvič, DNA molekule omočajo izjemno visoko gostoto zapisa informacij, saj za zapis 1 bita potrebujemo le 1 nm^3 molekule DNA, kar je vsaj za faktor 10^9 gosteje kot pri silicijevih čipih. Drugič, DNA računanje je energijsko mnogo manj potratno kot običajni računalniki. Tudi s slednjimi bi v principu lahko dosegli linearno časovno odvisnost, če bi število paralelno povezanih računalnikov eksponentno naraščalo. Toda kmalu bi poraba električne energije presegla dane okvire. En joule energije namreč zadošča za 10^{10} operacij na običajnem računalniku, a kar za 10^{19} operacij DNA-računalnika. Torej, eksponentno naraščajočo zahtevnost DNA-računalnikov si laže privoščimo. Meje seveda obstajajo, za rešitev problema HPP z 200 verteksi bi potrebovali več molekul DNA, kot tehta Zemlja.

Poleg same zahtevnosti računanja se pri velikih sistemih pojavijo tudi napake, saj je računanje z DNA stohastično. Z večanjem števila molekul se povečuje tudi verjetnost napak pri računanju in le-ta kmalu preseže verjetnost,

da je končni rezultat pravilen (kot že omenjeno, rezultat DNA računanja je vedno pravilen le z določeno verjetnostjo). Najbolj nevaren korak z vidika napak je četrti korak, ekstrakcija, kjer izločimo le tiste molekule, ki vsebujejo dano zaporedje. Če se pri tem ne ujamemo molekul s pravilnimi potmi, bo rezultat računanja navidezen 'NE', če pa slučajno izločimo tudi kakšno molekulo z napačno potjo, pa bo rezultat navidezen 'DA'. Slednji primer je sicer manj nevaren, saj lahko rezultat še enkrat preverimo.

Tudi za teoretično reprezentacijo verteksov in poti porabimo veliko časa, saj moramo izbrati takšno reprezentacijo, da med računanjem ne bi prišlo do neželenih hibridizacij (npr. hibridizacija dveh med sabo zamaknjenih vlaken). Zahtevnost priprave sicer narašča vsaj kvadratično v n, saj moramo za reprezentacijo i-tega verteksa upoštevati $i - 1$ prejšnjih. Tudi čas izvedbe celotnega eksperimenta ne sme potekati predolgo, saj sicer molekule DNA razpadejo.

3 Zaključek

Računanje na molekularnem nivoju s pomočjo molekul DNA obeta možnost uporabe bioloških procesov v tehnološke namene. Metodo so zaenkrat preizkusili le v demonstrativne namene, kljub navedenim omejitvam pa lahko upamo, da jo bodo uporabili tudi v uporabne namene. Pri tem je ključnega pomena izrabiti intrinsično prednost računanja DNA, to je paralelizacija. Za sekvenčne tipe nalog DNA računanje nikoli ne bo nadomestilo običajnih računalnikov, uporabo pa smemo pričakovati pri kodiranju informacij in pri problemih, ki se prevedejo na HPP. Za uspešno implementacijo je potrebna še avtomatizacija vseh korakov računanja, saj trenutno računanje temelji na človekovi asistenci.

Literatura

- Leonard M. Adleman, *Molecular computation of solutions to combinatorial problems*, Science **266**, 1021-1024 (1994).
- Leonard M. Adleman, *On constructing a molecular computer*, Proceedings of DIMACS Workshop, Princeton, 1-22 (1995).
- Leonard M. Adleman, *Computing with DNA*, Scientific American, 54-61 (1998).
- L. Kari *et al.*, *A computer scientist's guide to molecular biology*, Soft Computing **5**, 95-101 (2001), glej tudi predstavitev, dostopno na `http://bi.snu.ac.kr/Courses/g-ai02/materials/DNAC_MB.ppt`.
- Jan Prokaj, *DNA Computing*, predstavitev, dostopna na `http://www.cs.ucf.edu/courses/cot4810/spr2005/item/ppt16.ppt`
- Will Ryu, *DNA Computing: A primer*, `http://arstechnica.com/reviews/2q00/dna/dna-1.html`