



INIGO IBÁÑEZ

JOSU DONCEL VICENTE

(Getxo, 1985). Matematika eta Elektronikako Ingeniaritza Teknikoa ikasi zuen EHU, baita Matematika Aplikatua eta Estatistika masterra ere. Okzitaniako Tolosako Unibertsitateko doktoretza-titulua atera zuen 2015ean. Azken urteotan, LAAS-CNRS (Frantzia), Inria (Frantzia) eta BCAM-Basque Center for Applied Mathematics (Bilbo) ikerketa-zentroetako ikertzailea izan da, eta irakasle aritu da ENSIMAG (Frantzia), INSA-Toulouse (Frantzia) eta IUT-Blagnac (Frantzia) unibertsitateetan. Gaur egun, EHUko Matematika Aplikatua eta Estatistika saileko irakaslea da.

Sareen arteko komunikazioak bizkortzea: bide optimoa aurkitzen duten algoritmoak

Telekomunikazio-sare modernoetan, hainbat motatako gailuek (ordenagailuak, telefonoak eta tabletak, esate baterako) datu-paketeak bidaltzen dizkiote elkarri; hori egiten dutenean, komunikatu egiten direla esaten dugu. Sare horien erabiltzaileek kalitatezko zerbitzua eskatzen dute (Youtube-ko bideo bat etenik gabe ikustea edo mezu elektronikoak ahalik eta lasterren ja-

sotzea, adibidez), eta kalitatezko zerbitzu hori beteko bada, garrantzitsua da paketeak sareko bide arinenetik, hau da, optimotik, bidaltzea. Bide optimoa aurkitzeko, hala ere, sare osoaren unean uneko egoera ezagutu behar da, eta hori, zoritxarrez, ezinezkoa da, ezin baitugu aurreikusi zenbat erabiltzaile izango diren sarean, ez eta zer zerbitzu eskatuko duten ere.



1. irudia. Esperimenturako aukeratutako 19 ordenagailuen kokalekua. Esperimentuaren emaitzaren arabera, ordenagailuen arteko komunikazioak ongi asko hobetu daitezke. JOSU DONCEL

Artikulu honetan, Bide Optimoaren Aurkikuntza (BOA) deritzon eredu matematikoaren gainean hitz egingo dugu; izan ere, haren bidez, ordenagailu-multzoen arteko komunikazioak bizkortu daitezke. Funtsean, algoritmo optimoa aurkitzean datza, BOA problema modu optimoan ebazten duen algoritmoa aurkitzean, alegia. Algoritmo batek BOA problema modu optimoan ebazten du, baldin eta, sareari buruzko informazio minimoa eskatuz, bi ordenagailuren arteko bide optimoa aurkitzen badu.

BOA problema modu optimoan ebazten duten algoritmoak aztertuko ditugu. Ikusiko dugu bada, la optimotik gertu dagoen algoritmo bat, eta, halaber, algoritmo hori Google, Facebook edo Amazonen gisako enpresetako ordenagailu-multzoetan erabil daitekeela.

PAKETE-BIDALKETAK BIZKORTZEAREN ORDENAGAILU-MULTZOETAN

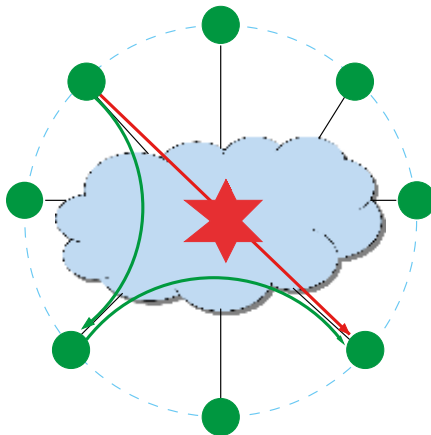
BOA problemaren garrantzia erakusten duen esperimentu bat azalduko dugu jarraian. Munduan zehar banatuta dauden 19 ordenagailu aukeratu ditugu (ikus 1. irudia). Ordenagailu bakoitzetik gainerako ordenagailuetara pakete bat bidali dugu, eta pakete-bidalketa bakoitzaren latentzia neurtu dugu, hau da, zenbat denbora behar izan duen paketeak sorrera-ordenagailutik xede-ordenagailura heltzeko. Paketea bi minututik bi minutura bidali dugu astebez.

Esperimentuak agerian utzi du askotan kongestio-arazoak izaten direla bi ordenagailuren artean, eta, bidalketak bizkortze aldera, komeni dela paketeak bitarteko ordenagailu batetik igarotzea. Adibidez, esperimentuko neurketen arabera, pakete batek 100 ms behar ditu Irlandatik Kaliforniara joateko, 60 ms Irlandatik São Paulora joateko, eta 20 ms São Paulotik Kaliforniara joateko. Denbora horiei erreparatzen badiegu, erraz ikus dezakegu Irlandatik Kaliforniara doazen paketeak arinago heltzen direla São Paulotik igaroz (80 ms) Kaliforniara zuzenean bidaliz baino (100 ms). Kasu honetan, beraz, komeni da paketeak beste ordenagailu batzuetatik pasatzea, helmugara bizkorrago heltzeko.

BOA PROBLEMAREN DESKRIBAPENA

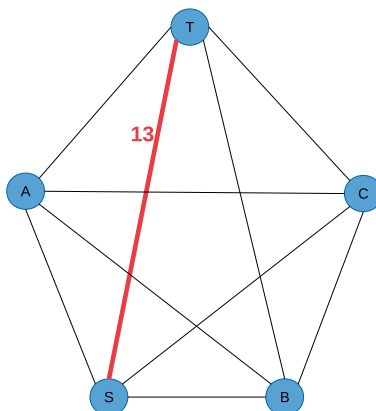
Aurreko ataleko esperimentuaren emaitza aintzat hartuz, argi dago ordenagailu-sistemetako komunikazioak hobetu egin daitezkeela, bide zuzenean kongestiorik izanez gero paketeak bitarteko ordenagailuetatik pasaraziz. Sistema horretan paketeak bide arinetik, hau da, bide optimotik pasatzeko, ordea, ordenagailu guztien

arteko latentzia kalkulatu behar da, eta, aintzat hartuta n ordenagailuko sistema batean egin beharreko neurketa-kopurua n^2 ordenakoa dela, n handia bada, ezinezkoa izan liteke hainbeste latentzia kalkulatzeko. Beraz, bide optimoa zein den jakitea lortu behar dugu, baina sare osoaren latentziak kalkulatu beharrik gabe.



2. irudia. Artikulu honen ideia nagusiaren azalpena. Bi nodoren artean kongestiorik badago, paketeak bitarteko nodo batetik pasatuz, komunikazioa asko bizkortzen da. JOSU DONCEL

Jo dezagun sistemako ordenagailu-multzoa n erpinetako grafo bat dela, zeinaren erpinak ordenagailuak baitira, eta ertzak ordenagailuen arteko bideak (Internet protokoloak ezarritakoak, hain zuzen). Grafo hori osoa da (alegia, edozein erpinetatik edozein erpinetara, beti ertz bat dago), eta ez-orientatua (A eta B erpinen arteko latentzia eta B eta A erpinen artekoa berdina dira). Sarean, ertzen balioak (ordenagailuen arteko latentziak) ezezagunak dira, eta haien balioak



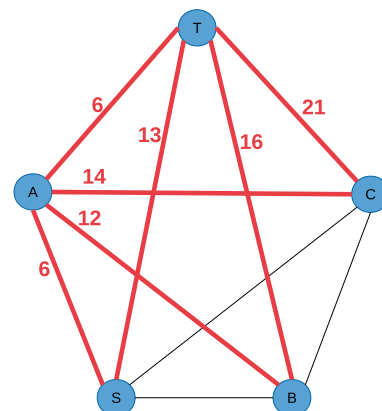
3. irudia. (S, T) ertza besterik kontsultatzen ez duen algoritmoa. JOSU DONCEL

jakitea ezinbestekoa da bide optimoa zein den jakiteko. Ertz baten balioa jakiteko, kontsulta bat egiten da, eta horrek kostua du. Horiek horrela, BOA problema era optimoan ebazten duten algoritmoek batetik bi erpinen arteko bide optimoa aurkitu behar dute, eta, bestetik, ahalik eta kontsulta gutxien egin behar dituzte hori lortzeko, kostua txikia izan dadin.

BOA PROBLEMA EBAZTEN DUTEN ALGORITMOAK

BOA problema ebazten duen algoritmoak bide optimoa aurkitu arte kontsultatzen ditu grafoaren ertzen balioak. Demagun, esaterako, bost erpinetako grafo oso batean S eta T erpinen arteko bide arinena aurkitu nahi dugula, ahalik eta kontsulta gutxien eginez (adibide honetan, beraz, bide optimoa bide arinena da). Algoritmoak, lehenik, S eta T erpinen arteko ertzen balioa, hau da, (S, T) ertzen balioa kontsultatzen du. Ertz hori besterik kontsultatu ezean, algoritmoak ez daki benetan hori denentz bide arinena: (S, A) ertzen eta (A, T) ertzen balioak, adibidez, ez ditu kontsultatu, eta baliteke bide arinena A erpinetik igarotzea. Ez du, beraz, BOA problema ebatzi, eta, horregatik, kontsultak egiten jarraitu behar du bide optimoa aurkitzeko.

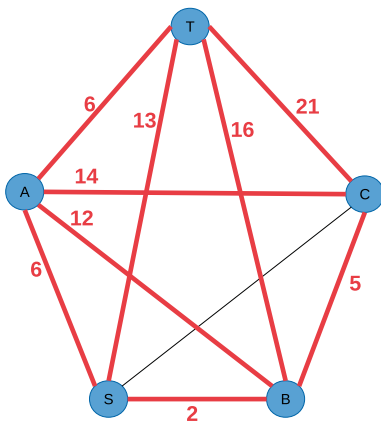
BOA problema ebazten duen algoritmo bat 4. irudian irudikatutik dago. Algoritmo horrek, (S, T) ertza ez ezik, beste 8 ertz ere kontsultatzen ditu, eta, behin bide arinena aurkitzen duenean, ez du kontsulta gehiago egiten. Kontsultatutako ertzen balioak aintzat hartuta, bide arinena (S, A, T) bidea da, hau da, S, A eta T erpinetatik pasatzen dena; izan ere, (S, A) eta (A, T) ertzen balioa 6 denez, bide arinenaren balioa 12 da, eta S erpinetik T erpinera doazen gaine-



4. irudia. Bide optimoa aurkitzeko bederatzi kontsulta egiten dituen algoritmoa. JOSU DONCEL

rako bide guztiek balio handiagoa duen ertzen bat dute; beraz, bide laburrenak baino balio handiagoak dituzte.

Algoritmo hori, hala ere, BOA problema ebazteko baliagarria den arren, ez da, agian, algoritmo optimoa; izan ere, beste algoritmo batek zazpi ertzen balioak soilik kontsultatuz lor dezake bide arinena aurkitzea (5. irudian). Hala bada, BOA problema era optimoan ebazteko, zazpi kontsultako algoritmoa baliatu behar da.



5. irudia. BOA problema era optimoan ebazten duen algoritmoa. Bide laburrena aurkitzeko, zazpi kontsulta eginen ditu. JOSU DONCEL

BOA problema ebazten duen algoritmo bat optimotik gertu dagoen jakiteko, kontsulta-arrazoa erabiltzen da. Algoritmo baten kontsulta-arrazoa zatiketa bat da: algoritmo horrek egin dako kontsulta-kopurua zati algoritmo optimoaren kontsulta-kopurua. Adibidez, 4. irudian agertzen den algoritmoaren kontsulta-arrazoa $9/7$ da, algoritmoak bederatziz ertzen balioak kontsultatzen dituelako BOA problema ebazteko, eta algoritmo optimoak, ordea, zazpi; algoritmo bat ez da optimoa haren kontsulta-arrazoa bat baino handiagoa bada.

ALGORITMO OPTIMOAREN BILA

Ikusi dugu aurreko ataleko adibiderako bada-goela algoritmo optimo bat, hots, kontsulta-kopuru minimoa eginez bide optimoa aurkitzen

duen algoritmo bat, eta horrek bidea eman diezaguke ondorioztatzeko algoritmo hori optimoa dela edozein grafo osorentzat. Gure analisisen arabera, ordea, ondorio hori ustela da. Izan ere, guk frogatu dugu ez dagoela ezein algoritmorik n erpineko grafo oso guztiarentzat optimoa denik. Alegia, algoritmo baten kontsulta-arrazoa, grafo batean, 1 izan daiteke (hau da, era optimoan ebazten du algoritmoak BOA problema) eta beste batean, berriz, haren kontsulta-arrazoa $1 + 0.25 \cdot n - 0.125 \cdot n^2$ da; alegia, bat baino heretik handiagoa da.

Gure helburua da n erpineko grafo oso orotan BOA problema modurik onenean ebazten duen algoritmo bat aurkitzea. Gure ikerketan, BOA problema ebazten duen algoritmo bat asmatu dugu. Guk frogatu dugu algoritmo horrek gehien jota $2n - 1$ kontsulta behar duela BOA problema ebazteko, eta algoritmo optimoak, aldiz, n baino ez. Horrek esan nahi du haren kontsulta-arrazoa $2 - 1/n$ dela grafo guztietan, hau da, 2tik behekoa, grafo teoriarik ezagunak diren beste algoritmo batzuen baino txikiagoa.

Algoritmoaren deskribapena:

Biz $D = \{s, t\}$, $u_1 = s$ eta $u_2 = t$

Bide optimoa aurkitu ezean,

Kontsultatu (u_1, v) eta (v, u_2) , non v ez dagoen D multzoan.

Eguneratu u_1 : D-ko erpin guztietatik s erpinera bide arinena duena.

Eguneratu u_2 : D-ko erpin guztietatik t erpinera bide arinena duena.

Gehitu u_1 eta u_2 erpinak D multzora.

Saiatu bide optimoa aurkitzen.

Alegia, grafo-teoriako algoritmo batzuk ere aztertutitu n erpineko edozein grafo osotan, eta, algoritmo horien kontsulta-arrazoa, gure analisisen arabera, n ordenakoa izatera hel daiteke: muturreko kasu batzuetan, grafo-teoriako algoritmoak ertz guztien balioa kontsultatu behar dute, hau da, n^2 ertz. Algoritmo optimoak, aldiz, n ertz kontsultatzen du bide optimoa aurkitzeko. Beraz, haien kontsulta-arrazoa n ordenakoa da, alegia, gure optimotik gertu dagoen gure algoritmoarena baino askoz handiagoa, zeina, esan bezala, 2tik beherakoa baita.

EMAITZAK INPLEMENTAZEN: GOOGLE, FACEBOOK ETA AMAZON

Gure ikerketaren emaitzak, gure ustez, telekomunikazio-sare modernoan komunikazioak hobetzeko erabil daitezke. Google, Facebook, Amazon eta haien gisako enpresek ordenagailu-multzo handiak erabiltzen dituzte bezeroei zerbitzua emateko (adibidez, Googlek, ustez, milioi bat zerbitzari baino gehiago ditu, hainbat herrialdetan banatuak).

Enpresa horien bezeroek kalitatezko zerbitzua eskatzen dute; horregatik, enpresa horientzat oso garrantzitsua da komunikazio bizkorak gauzatea. Hori lortzeko, ezinbestekoa dute sarearen bide optimoa zein den jakitea. Gure ustez, aurkeztu dugun algoritmoa erabiliz, aukera izango dute sarearen egoera modu arrazoizko batean ezagutzeko. Izan ere, algoritmo hori tar-teko dela, kasurik txarrean ere, optimotik gertu dago, sarearen egoera jakiteko egin behar den ahalegina.

BIBLIOGRAFIA

- [1] ABRAHAM, I.; DELLING, D.; FIAT, A.; GOLDBERG, A.V.; WERNECK, R. E. "Highway dimension and provably efficient shortest path algorithms". *Journal of the ACM*, 63(5) Dec. 2016, 41:1–41:26.
- [2] GIBBONS, A.: *Algorithmic graph theory*. Cambridge University Press, 1985.
- [3] HARCHOL-BALTER, M.: *Performance Modeling and Design of Computer Systems*. Cambridge University Press, 2013.
- [4] HARRISON, P. G., PATEL, N. M.: *Performance Modelling of Communication Networks and Computer Architectures*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1992.
- [5] SCHAEFFER, S. E. et al: *Algorithms for nonuniform networks*. Helsinki University of Technology, 2006.
- [6] SZEPESVÁRI, C.: "Shortest path discovery problems: A framework, algorithms and experimental results". In *AAAI*, (2004), pages 550–555.
- [7] VAN STEEN, M.: *Graph theory and complex networks: An introduction*. Amsterdam, 2010.