

Bide-hautaketa problema, optimizazio problema. Zenbait ebazpide

Olatz Arbelaitz eta Clemente Rodriguez

Konputagailuen Arkitektura eta Teknologia saila
649 p.k. 200 80 DONOSTIA

Laburpena: Lan honen helburua, irakurlea bide-hautaketa problemen garrantziaz ohartarazi eta horiek ebazteko erabil daitezkeen metodo desberdinak aipatzea da. Bide-hautaketa problema zehatz bat deskribatzeaz gain, hau ebazteko bibliografian topa daitezkeen aukerak azaltzen dira bertan.

1. SARRERA

Bizi garen inguruneko enpresa askok egunero egin behar diote aurre pertsona, produktu eta antzekoen garraioari. Horrelako lanak ez dira helburu gisa garraioa duten enpresetan bakarrik egiten, izan ere, edozein enpresak eduki dezake, bai pertsonak eta bai materialak, garraiatzeko beharra, beste toki batzuetara edo beste jatorri batzuetatik. Enpresen garraio beharrak gero eta ugariagoak dira, globalizazio prozesua, eta, gero eta zabalduagoa dagoen espezializazioa direla eta. Horregatik, elementuen garraioaren ondorio den kostua edozein enpresetako atal garrantzitsu bihurtzen da. [Bodin 90]-en eskola autobusak, zabor-biltze zerbitzua eta antzeko adibide zehatzak topa daitezke eta [Bodin et al 83] lanean berriz, mota horretako arazoek deskribapen formala eta zenbait soluzio bide.

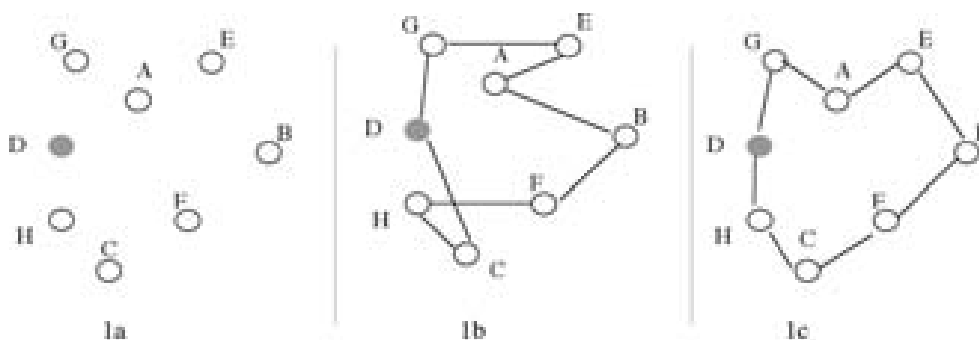
Adibide gisa, [Bodin et al 83]-n aurkeztutako ikerketaren arabera, 1980an EEBBetan banaketa lanetan 400 bilioi dolar gastatzen ziren eta EBn 15 bilioi libera. [Fisher et al 97]-n ziurtatzen dute 1978an garraioari lotutako gastuak nazio-produktu gordinaren %15a suposatzen zuela. Horrelako adibideek nabarmenagoa egiten dute problema horiek modu eraginkorrean ebaztearen garrantzia.

Garraio-problemek enpresa munduan duten eragina dela eta, horiek ebazteko egindako ahalegina handia da; ezin ahaztu baitaiteke, ibilgailuen bide-hautaketa hobetzeak, planifikazio edo antolaketaren (*scheduling*) ho-

bekuntzarekin batera, enpresen ekonomian eragin zuzena izango duen aurrezte izugarria suposatuko duela.

Bide-hautaketa problemarik sinpleena ezaugarri geografikoak besterik ez dituen da, hau da, helburu gisa zenbait punturen arteko bide minimoa topatzea duena. Bide-hautaketa eta antolaketa problemak berriz, bide-hautaketa soila baino zabalagoak dira, bide minimoak topatzea helburu izateaz gain, beste faktore batzuk kontuan hartu behar dituztenak, hauetan, baliabideen erabilera egokia lortu behar izateak eta bestelako murrizketek soluzio optimoa topatzea izugarri zaila egiten dute.

Mota honetako problemak optimizazio konbinatorioko problemen artean sailka daitezke, hau da, soluzio-multzoa finitua baina esponentziala dutenen artean. Hauetako zenbait problema ebazteko badaude algoritmo eraginkorrak. Beste batzuk zehazki ebazteko berriz, denbora esponentziala behar da. Problema horietan guztietan, soluzio posibleen multzoa esponentzialki hazten da bere sarreraren tamaina handitzen denean. Askotan, soluzioak elementuen ordenazio desberdinak eratuz lortzen dira eta konbinazio posibleen kopurua ordena faktorialekoa da. Bide-hautaketakoa soilik eta oinarritzakoa den problema *TSP* (*Traveling Salesperson Problem* edo *Merkatariaren Bidaiariaren problema*) da. *TSP*-an bidaiari batek hiri-multzo bat zeharkatu behar du, hiri guztietatik behin bakarrik paseaz. Helburua, bidaiariak ahalik eta distantzia txikiena ibiltzea da. 1 Irudian *TSP* problemaren adibidetxo bat ageri da. 1a atalak problema erakusten du, 8 hiriko multzoa, abiapuntua C hiria izanik. 1b atalak soluzio posible bat erakusten du baina begibistakoa da ez dela soluziorik laburrena eta azkenik, 1c atalean optimoa izan daitekeen soluzio bat ageri da



1. irudia. *TSP* problemaren adibidea. (1a) hiriak, (1b) soluzio bat (1c) soluzio optimoa.

*TSP*ren n hiriko instantzia batean, azter daitezkeen konbinazio posibleen kopurua $(n - 1)!/2$ -koa da. Ondorioz, 3 hiriko kasurako soluzio posible ba-

karra badago ere, 50 hiriko problema baterako berriz, soluzio posibleen kopurua $3 \cdot 10^{62}$ ingurukoa izango da. Agerikoa da beraz, soluzio posible guztiak banan-banan aztertzea ezinezkoa dela, tamaina batetik aurrerako instantzietan.

Bide-hautaketa problemen artean, ibilgailuen bide-hautaketarena (*VRP* edo *Vehicle Routing Problem*) oso ezaguna da. Kasu horretan, helburua ibilgailuentzat kostu minimoa duten ibilbide-multzoa diseinatzea da, ibilbide guztiak biltegi batean hasi eta amaitzen direlarik, eta bezeroaren eskari-multzo osoa zerbitzatzeko erabiliko den ibilgailu-multzoa mugatua izanik. Bezero bakoitzak eskari bat egiten du, ibilgailu bat soilik erabiliz zerbitzatu behar dena eta bestalde, ibilgailu batek ezin du garraiatu bere kapazitatea gainditzen duen kargarik. Bezeroa ezin denean edozein unetan zerbitzatu, hau da, denborazko baldintzak (*TW Time Windows*) agertzen direnean, ebatzi beharreko problema, denborazko baldintzak bete behar dituen ibilgailuen bide-hautaketako problema bihurtzen da (*VRPTW Vehicle Routing Problem with Time Windows*).

Guk egindako azterlan nagusia azken problema horren ingurukoa izan da; enpresa bateko eguneroko erregai banaketaren planifikazioa egiteko gai den sistema eraiki dugu hain zuzen ere, eta hori dela eta, lan honetan azalduko diren zenbait erreferentzia harekin lotuta egongo dira. Aurrerago argitaratuko den artikuluko batean aurkeztuko ditugu problema erreala bera, haren ebazpena, eta lortutako emaitzak.

Optimizazio konbinatorioko problemek kalkulu ahalegin handia eskatzen dute eta bestalde, denbora mugatuan lortu behar da soluzioa. Hau dela eta, soluzio-bide egokiak aurkitzea garrantzitsua da.

Lan honen helburua, mota honetako problemak ebazteko erabiltzen diren metodoen aipamen laburra egitea da. Helburu horrekin, jarraian datozen atalek gai desberdinak jorratuko dituzte: 2. atalean, optimizazio problema bat zer den formalki deskribatuko da. 3. atalak, optimizazio-problema ebazteko erabil daitezkeen metodoak laburki azaltzen ditu, metodo zehatzak eta heuristikokoak tartean direlarik. Jarraian, 4. atalean, *VRPTW* problema ebazten duten ikerlariak beren emaitzak ebaluatzeko erabiltzen dituzten datu-multzoa eta emaitza motak deskribatzen dira. Azkenik, 5. atala lan honetatik atera daitezkeen ondorio labur batzuk aipatzeko erabiltzen da.

2. OPTIMIZAZIO PROBLEMEN DEFINIZIOA

Optimizazio problema bat honela defini daiteke:

$f(\underline{x})$ funtzio jakin bat minimizatu, soluzio posibleen multzoa $\underline{x} \in A$ mugatzen duen murrizketa-multzoa kontuan izanik.

f -k helburu-funtzio edo kostu-funtzio izena hartzen du eta minimizatu beharreko kostu izeneko kantitatea neurtzen du. A soluzio espazioak, problemaren murrizketak betetzen dituzten erabaki-aldagaien, \underline{x} , konbinazio guztiak ditu. \underline{x} erabaki-aldagaien instantziario jakin bat A -n badago posiblea dela esaten da.

Optimizazio-problema oro modu desberdinetan adieraz daiteke. Adierazpen guztiei dagokien soluzio optimoa bera izan arren, adierazpen desberdinei dagozkien ezaugarri matematikoak eta ebazteko zailtasuna ere aldatu egiten dira.

3. SOLUZIO-BIDEAK

Soluzio optimoa ezezaguna duen optimizazio konbinatorioko edozein problematarako, soluzioa bilatzeko modurik intuitiboena soluzio guztiak banan-banan sortu eta bakoitzari dagokion kostua kalkulatzeko da, baliorik onena duenarekin geratuz.

S soluzio-espazioaren tamaina dela eta, prozedura hori erabiltzeak denbora gehiegi eskatuko luke kasu gehienetan. Bilaketa-espazioa jarraitua den bestalde, ezinezkoa izango da erabiltzea, aztertu beharreko soluzio-kopurua infinitua izango baita. Banan-banako azterketak kasu oso zehatzetan soilik erabil daitezke: tamaina txikikoak izanik, bilaketa-eremu guztia aztertzeko denbora ematen duten problemetan. Beraz, optimizazio konbinatorioko problemak ebazteko modu adimentsuagoak bilatu beharko dira.

3.1. Metodo zehatzak

Multzo honetako metodoek, soluzio bat topatzen dutenean optimoa dela ziurtatzen dute.

Aipatu den moduan, optimizazio-problema erabaki-aldagaiak eta murrizketa-multzoa dituen funtzio baten moduan adierazi ohi dira maiz, eta ekuazio-multzo horretan oinarritzen dira metodo hauek soluzioak bilatzeko.

Helburu-funtzioa eta murrizketak linealak badira, problema hauek Programazio Lineal bidez ebatz daitezke. Aldagaietako bateon bat osoa bada, Programazio Lineala eta osoaren arteko nahasketa bat edukiko dugu; eta, aldagai guztiak osoak badira berriz, Programazio Osoko problema baten aurrean egongo gara. Erabaki-aldagaiak diskretuak direnean, aldiz, optimizazio konbinatorioko problema bat ebatzi beharko dugu. Badago azken mota honetako problemak programazio oso edo linealeko problemetara bihurtzeko aukera, baina bilaketa-espazioaren tamaina handitzen den heinean, hauek ebazteko beharrezko denbora onartezina suertatzen da.

Hori dela eta, soluzio optimoa topatzeko erabiltzen diren metodo zehatzak problemaren edo soluzio-multzoaren zatiketan oinarritzen dira oro har, azken emaitza lortzeko, soluzio partzialak konbinatzen dituztelarik. Gehien erabiltzen diren teknikak Lagrange-ren Erlaxazioa [Fsher 85], Zutabe Sor-tzea [Dantzig et al 74] eta Programazio Dinamikoa dira. Bigarrena da hain zuzen ere ibilgailuen bide-hautaketa problemak ebazteko gehien erabiltzen dena.

Metodo horiek erabiliz lortutako emaitzen ezaugarriak honela laburbil ditzakegu:

- a) Kalitate onekoak dira tamaina txikiko problementzat. Baina emaitza bat aurkitzeko behar duten denbora magnitude ordena bat edo bi handiagotzen da sarreraren tamaina birekin biderkatzen denean eta ondorioz, tamaina handiko problemak ebazteko horrelako metodoak erabiltzea pentsaezina da.
- b) Horrelako metodoek duten desabantaila bakarra ez da problemaren tamaina eta denboraren arteko erlazioa. Izan ere, ebatzi beharreko problemaren zehaztapenean egindako edozein aldaketa txikik, sistema problema berrira egokitze ahalegin handia eskatzen du, baita zenbait kasutan jatorrizko sistemaren zenbait atal birdiseinatu behar izatea ere. Egoera hau oso maiz gertatzen da sistema errealean.
- c) Metodo hauek ez dute beti denbora mugatuan soluzio bat aurkitzen. Aztertu diren ikerlanetan ez da azaltzen zer gertatzen den soluzio optimoa topatzen ez denean. Badirudi kasu horietan sistema ez dela denbora mugatuan emaitza posible bat emateko gai. Hori onartezina izango da sistema errealean, behar beharrezkoa baita soluzio bat topatzea, optimoa ez bada ere.

3.2. Metodo heuristikoak

Metodo hauek ez dute ziurtatzen aurkitutakoa problemaren soluzio optimoa denik, baina, oro har, kalitatezko soluzioak bilatzeko gai dira. Irizpide desberdinak erabil daitezke metodo heuristikoak sailkatzeko baina ibilgailuen bide-hautaketa problemak ebazteko erabiltzen diren metodoak aztertzen baditugu, bi multzo nagusitan banatzen direla esan dezakegu: ibilbideen eraikuntza metodoak eta hobekuntza metodoak.

3.2.1. Ibilbideen eraikuntza metodoak

Metodo hauen helburua soluzio bat modu adimentsuan eraikitzea da. Oro har, ibilbideak pausoz pauso osatzen dira bezeroak irizpide edo heuristiko desberdinen arabera txertatuz. Metodo hauek ez dira gaur egun isolatuta erabiltzen, baina bibliografia aztertuz gero, bide-hautaketa problemak

ebazteko horrelako metodoak erabiltzen dituzten adibide mordoa topa daiteke. Gehientsuenak [Solomon 87] artikuluan deskribatzen dira. Desberdintasun nagusiak bi arlotan topa daitezke: bezeroak ibilbideetan txertatzeko aukeratutako irizpideetan eta ibilbideak seriean edo paraleloan eraikitzean. Ibilbideak seriean eraikitzen direla esaten da banan-banan eraikitzen direnean, hau da, ibilbidean txertatu ezin den lehenengo bezeroa agertzen denean ibilbidea amaitutzat eman eta berri bat hasten denean; paraleloan eraikitzen direla esaten da berriz, ibilbidea amaitutzat ematen ez denean prozesu osoa amaitu arte. Ibilbideen eraikuntza metodo aipagarrien artean honako hauek daude: Gertueneko Auzokoaren Heuristika (*Nearest Neighbour Heuristic*), Aurrezte-Heuristika (*Saving Heuristic*), Txertatze-Heuristika (*Insertion Heuristic*) eta Denborazko Ekortze-Heuristika (*Time-oriented Sweep Heuristic*).

Adibide gisa Aurrezte-Heuristika azalduko dugu, [Clarke et al 64] artikuluan deskribatutakoan oinarritzen dena. Algoritmoa *biltegi-i-biltegi* ibilbide posible guztiekin hasten da eta iterazio bakoitzean bi ibilbideren murrerako bezeroen arteko lotura egiten du, hain zuzen ere aurrezki handiena lortzen dutenena. Bi bezero (i, j) elkartzean lortutako aurrezki honako espresioaren bidez kalkulatzen da:

$$\text{aurrezkia}_{ij} = d_{i0} + d_{0j} - d_{ij}, \quad \geq 0 \quad (2.2)$$

d_{i0} eta d_{0j} i bezerotik biltegiara eta biltegitik j bezerora dauden distantziak dira eta d_{ij} bi bezeroen arteko distantzia.

Bi ibilbide elkartzean sortzen den soluzio berria posiblea ote den aztertu behar da, baldintza guztiak betetzen ote dituen hain zuzen ere (ibilgai-luen kapazitatearekin lotutakoak, denborarekin lotutakoa ...). Argi dago, ibilbideak eraikitzeke modu hau metodo paraleloa dela, hau da, aldi berean ibilbide multzo batekin lan egiten dela eta hauetako bakoitza ez dela amaitutzat ematen prozesu guztia bukatzen den arte.

3.2.2. Ibilbideen hobekuntzan oinarritutako heuristikak

Hasierako soluzio batetik abiatu, ausaz sortutakoa edo ibilbideen eraikuntza heuristika bat erabiliz sortutakoa, eta aldaketa txikien bidez hau hobetzen saiatzen diren metodoak dira. Hobekuntzan oinarritutako heuristika ororen oinarria auzoa kontzeptua da. s soluzio baten auzoa, $N(s)$ soluzio-multzoa da; s soluzioan aldaketa bat eginez lortu daitekeena. Auzo baten soluzio guztiak edo multzo bat aztertzea helburu-funtzioaren arabera hobeak diren soluzioak aurkitzeko modua izan daiteke. Ideia hau errepika daiteke topatutako soluzio hoberenetik abiatuz, auzoan soluzioa hobetu ezin den puntura heldu arte. Puntu honetan optimo baten aurrean egongo gara; optimoa lokala izango da gehienetan, baina, baliteke optimo absolutu

baten aurrean egotea ere. Auzo-egitura aldatzeak beraz, bilaketa estrategia aldatzea suposatuko du.

Ohiko bilaketa-metodoak beheranzkoak dira: ausazkoa eta maldarik handienekoa. Hona hemen dagokien deskribapen orokorra eta algoritmoaren itxura (1 Algoritmoa):

- (a) **Ausazkoa:** iterazio bakoitzean bizilagun bat ausaz aukeratzen da bilaketa jarraitzeko. Iterazio bakoitzean soluzio batetik bere auzo-koa den eta kostu-funtzioaren balio hobea duen beste batera pasatzen da. Prozesua auzoko soluzioen artean balio hoberik duenik topatzen ez denean amaitzen da.
- (b) **Maldarik handienekoa:** iterazio bakoitzean auzo osoa aztertzen da eta soluziorik hoberena hautatzen da hurrengo iterazioaren abiapuntu gisa.

Hautatu hasierako soluzioa s_0

Errepika

(a) *Hautatu $s \in N(s_0)$ non $f(s) < f(s_0)$*

(b) *Hautatu $s \in N(s_0)$ non $f(s) = \min\{f(s_i)\}, \forall s_i \in N(s_0)$*

Ordezkatu s_0 s -z

Harik eta $f(s) > f(s_0), \forall s \in N(s)$

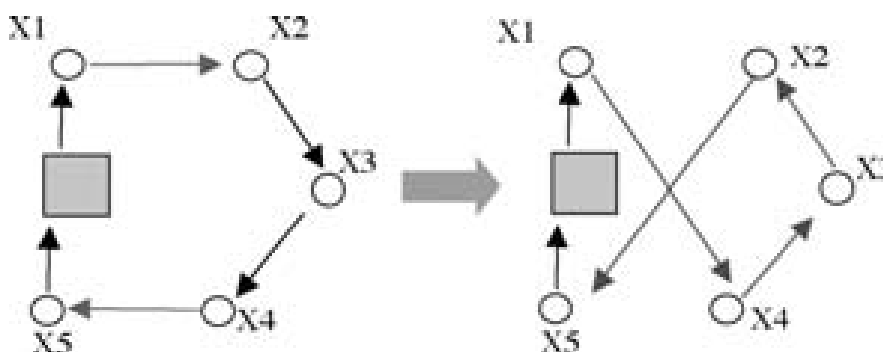
s_0 optimoaren hurbilketa izango da.

1. Algoritmoa. Beheranzko bilaketa-algoritmoaren egitura orokorra. (a) agindua Ausazkoaren kasuan egikaritzen da eta (b) Maldarik handienekoarenean. $N(s)$ auzo-egitura da eta $f(s)$ kostu-funtzioa.

Auzo-egiturek problemak ebazteko teknika askoren atal garrantzitsua osatzen dute, hauen artean ibilgailuen bide-hautaketa problemak. Azken hauetan erabiltzen diren auzo-egitura gehien funtsa bezeroen mugimendua da, ibilbide batean posizio batetik bestera, ibilbide desberdinen artean bezero trukeak eta antzekoak. Asko dira bibliografian aurki daitezkeen metodoak: *r-Opt* [Lin 65], *Or-Opt-1*, azken honen hedapena den *Or-opt* [Or 76], *2-Opt** [Potvin et al 95], azken honen hedapena den gurutzaketa [Taillard et al 95] eta abar.

Mota honetako problemak ebazteko erabiltzen diren auzo-egituren ideia hobeto ulertu ahal izateko, jarraian deskribatuko dugu auzo-egiturarik

erabilienetako bat, r -Opt [Lin 65]. Funtsean, ibilbide batean r arku aukeratu, ezabatu, eta beste r arkurengatik ordezkatzeko dira (Ikus adibidea 1 Irudian). Auzo-egituraren tamaina $O(n^r)$ koa da, eta hori dela eta, r handitzen den heinean, azter daitezkeen soluzio espazioa handiagoa eta anitzagoa izango da, eta, beraz, baita optimo bat topatzeko aukera ere. Bestalde, auzo-egitura osoa aztertzeke behar den denbora ere handiagoa da, eta bali-teke, beharrezko ahalegina gehiegizkoa izatea. Beharrezkoa da kostu konputazionalaren, r -ren menpe dagoena, eta lortutako soluzioen kalitatearen arteko oreka bat lortzea. Arrazoi hau dela eta, eraikitako aplikazioetan r -ren tamainarik handiena 3 izan ohi da.



2. irudia. 2-opt ren adibide bat.

2 Irudiko adibidean bi arku (x_1, x_2) eta (x_4, x_5) beste birekin (x_1, x_4) eta (x_2, x_5) trukatzeko dira. Horren ondorioz beste bi arkuren noranzkoa aldatu behar izan da lortutako emaitza ere soluzio posiblea izan dadin.

3.3. Metaheuristikak

Metaheuristikaren multzoko tekniken ezaugarri nagusia, errendimendua handiagotzeko adimen artifizialean, biologian, matematikan edo fisikan oinarritutako kontzeptuak erabiltzea da. Metodo gehienak auzoan egindako bilaketetan oinarritzen dira eta optimo lokaletatik ihes egiteko teknikak gehitzen dizkiete hauei.

Auzoan egindako bilaketak optimo lokal batean erori eta bertatik irteteko gai ez izatea oso arrunta da. Horregatik, optimo lokaletatik ihes egiteko eta bilaketa espazioaren beste atal batzuk aztertzeke mekanismoak dituzten metaheuristikak beharrezkoak dira. Horien artean aipagarrienak *Simulated Annealing* (SA), Tabu bilaketa, Metodo Ebolutiboak, etab. dira.

3.3.1. Simulated Annealing (SA)

Algoritmo honen oinarri diren ideiak Metrópolis eta Rosembluth-ek argitaratu zituzten 1953an [Metrópolis et al 53]: materialaren hoztea simulatzeko algoritmo bat da hain zuzen ere. 30 urte beranduago [Kirkpatrick et al 83] eta [Cerny 85], bakoitzak bere aldetik, horrelako simulazioak optimizazio problemetan (soluzio optimora heltzeko helburuarekin) soluzio posibleak bilatzeko erabil zitezkeela azaldu zuten. Metodo honi buruzko informazio zabala [Reeves 93, Díaz et al 96, Michalewicz et al00]-en topa daiteke.

Metodo hau, auzoan egindako bilaketa bat dela esan genezake, ausazkoa hain zuzen ere, baina minimo lokaletan erortzeko probabilitatea gutxitzen duena, bilaketa prozesuan okerragoak diren soluzioak onartzen baititu. Metodo honen konbergentzia matematikoki frogatuta dago eta ziurta daiteke kasurik txarrean, soluzio optimoa infinitu pauso ondoren topatuko dela (pauso-kopurua n , handia denean, konbergentzia ziurta daiteke Cantorniren teoremaren arabera).

SA algoritmoaren eskema orokorra, helburua f funtzio baten minimizazioa denean, 2 Algoritmoan aurkezten dena izan daiteke.

Hautatu $t = t_0$ hasierako temperatura, $\alpha(t)$ hozte-funtzioa eta L kate-luzera

iter_kont = 0

Hautatu ausaz s_0 hasierako soluzioa

Errepika

Errepika

Hautatu ausaz $s \in N(s_0)$

$\delta = f(s) - f(s_0)$

baldin $\delta < 0$ orduan $s_0 = s$

bestela, baldin ausazko $[0,1) < e^{-\delta/t}$ orduan $s_0 = s$

iter_kont ++

Harik eta iter_kont%L = 0

t = $\alpha(t)$

Harik eta gelditze-baldintza

2. Algoritmoa. SA algoritmoaren eskema orokorra.

Algoritmoan zehaztu gabeko parametro mordoa dago, problema-mota-
ren arabera finkatu beharko direnak. Hartu beharreko erabakiak bi multzo-
tan bana daitezke:

- Erabaki generikoak, prozesuan parte hartuko duten parametroei da-
gozkienak, hasierako tenperatura t_0 , tenperatura edo katearen luzera,
 L , hozte-funtzioa $\alpha(t)$ eta gelditze-baldintza adibidez.
- Problemaren ezaugarri zehatzekin lotuta dauden erabakiak, oso gar-
rantzitsuak hauek ere, adibidez, bilaketa edo soluzio-espazioaren
definizioa eta auzo-egitura.

Auzo-egitura egokia aukeratzea oso garrantzitsua da: edozein soluzio
lor badaiteke beste batetik abiatuta mugimendu-kopuru mugatuan, algorit-
moaren konbergentzia hasierako soluzioaren s_0 -ren independentea dela
ziurta baitaiteke.

Parametro hauen guztien aukeraketa egokiak baldintzatuko du *Simula-
ted Annealing* algoritmoaren konbergentzia-abiadura.

Esperientziak esaten du, hartu beharreko erabaki gehienak, bai algorit-
moarekin lotutakoak eta baita zehatzak ere, problema zehatzaren menpe-
koak izango direla eta parametroak modu egokian finkatzeko modua espe-
rimentazio luzea izango dela.

Bibliografian topa daitezke bide-hautaketa problemak ebazteko SA erabil-
tzen duten zenbait autore, adibidez [Janssens et al 94] eta [Chiang et al 96].
Oro har, 3.2.2. atalean aipatutako auzo egituren antzekoak erabiltzen dira so-
luzio berriak aztertzen joateko.

3.3.2. *Tabu bilaketa*

Algoritmo metaheuristiko asko motelak dira problemen tamaina oso
handia ez denean ere. Fred Glover-ek [Glover 90]-en aurkeztutako Tabu bi-
laketa, bilaketa-prozesua azkartzen saiatzen da, soluzio espazioaren azter-
keta «adimentsua» eginez. Tabu bilaketaren deskribapen zabalak honako
idazkietan topa daitezke [Reeves 93, Díaz et al 96, Michalewicz et al 00].

Teknika hau erabiltzen duten autoreek modu desberdinetan deskriba-
tzen dute. [Chiang et al 97] idazkian adibidez, esaten da Tabu bilaketa
«bilaketa lokaleko edozein metodotan txerta daitekeen metaestrategia
da, bilaketa optimo lokaletatik at eraman eta zikloak ekiditeko». Tabu
bilaketa maldarik handieneko beheranzko metodoarekin pareka liteke;
pauso bakoitzean mugimendu posibleen artean onena hautatzen du, bai-
na desberdintasuna da, bilaketa lokalak beti optimo lokalean amaitzen
duela, eta Tabu bilaketak ordea, uneko soluzioa baino kalitate kaxkarr-
gokoa den inguruneke soluzio batera mugitzeko aukera ematen duela,
era honetan optimo lokalak utzi ahal izateko. Metodo honek ausa oso

gutxi erabiltzen du bilaketa adimentsua modu sistematiko eta zuzenduan egin behar dela argudiatuz. Tabu bilaketaren inplementazio gehienak ia deterministak dira.

Batzutan, bilaketa on batek beharrezkoa izango du aurrez aztertutako soluzioak berriro aztertzea, bilaketaren helburu nagusia ez baita zikloak ekiditea, kalitate oneko soluzioak topatzea baizik. Hori dela eta Tabu metodoen baldintzak bete gabe utzi ahal izango dira zenbait kasutan, Tabu den mugimendu batek beste edozeinek baino emaitza hobek ematen dituen hain zuzen ere.

Metodoaren ideia nagusia, bilaketa-prozesua kontrolatzeko, memoria malgua erabiltzea da. Memoria honen funtzionamendua honela uler daiteke: demagun Tabu bilaketak, bilaketa prozesuan gertatutako egoeretako zenbait gauzen historia gordetzen duela, H , eta ohiko auzo-egitura, $N(s)$, aldatutako auzo-egituragatik ordezkatzeko duela, $N(H, s)$. Hau da, historiak parte hartzen du jatorrizko soluziotik mugimendu baten bidez lor daitezkeen mugimenduak zein diren finkatzerakoan. Horretaz gain, historiak, irigarriak diren soluzioen ebaluazio aldatuak lortzeko ere balio du, hau da, $f(s)$ helburu-funtzioa ordezkaturik izango da historia, H , kontuan hartzen duen beste batengatik $f(H, s)$. 3 Algoritmoan aurkezten da Tabu bilaketari dagokion algoritmo orokorra.

Hasieratu $s_{ora} S, H(hutsa), s_{hobe} = s_{ora}, kostu_{hobe} = f(s_{hobe})$

Errepika

Definitu $N_kandidato(s_{ora}) N(H, s)$ -ren azpimultzoa.

Hautatu s_{hur} $N_kandidato(s_{ora})$ non $f(H, s)$ minimizatzen den

Eguneratu historia H

$s_{ora} = s_{hur}$

Harik eta gelditze-baldintza.

3. Algoritmoa. Tabu bilaketari dagokion eskema.

Bibliografian bide-hautaketa problemak Tabu bilaketa erabiliz ebazten dituzten autore ugari topa daitezke, adibidez [Rochat eta al 95, Potvin et al 96a, Chiang et al 97, Schulze et al 99]. Horiek ere, oro har 3.2.2. atalean deskribatutako auzo egiturei aldaerak gehitzen dizkiete, horretarako historia kontuan izanik.

3.3.3. Metodo ebolutiboak

Metodo ebolutiboen familia Algoritmo Genetikoek, Programazio Ebolutiboak, Estrategia Ebolutiboek eta Programazio Genetikoak osatzen dute. Algoritmo Genetikoak eta Estrategia Ebolutiboak metaheuristikak dira, eta, hori dela eta, biek osatzen duten multzoari Metaheuristika Ebolutiboen multzoa ere esaten zaio.

Metodo hauek guztiak eboluzioarekin lotutako kontzeptu biologikoetan oinarritzen dira eta haien artean ageri diren desberdintasun nagusiak eboluzioan parte hartzen duten elementuetan eta eboluzioan zehar erabiltzen diren eragileetan oinarritzen dira. Deskribapenak [Schwefel 95, Michalewicz et al 00]-n topa ditzakegu.

Teknikarik ezagunena eta erabiliena Algoritmo Genetikoena da. Hori dela eta deskribapen zabalagoa aurkeztuko dugu.

3.3.3.1. Algoritmo Genetikoak (AG)

Algoritmo Genetiko izena kromosoma baten egitura genetikoa eta bektore baten bidez egitura konplexu bat adieraztearen arteko parekotasunetik dator. Animalia eta landareen selekzio naturalean adibidez, ezaugarri desiragarriren bat dutenek irauten dute, ezaugarri horiek, guraso-kromosomak konbinatzeko modua dela eta lortzen dira. Antzeko moduan, [Holland 75]-ek lehenengo aldiz azaldu zituen eta [Goldberg 89, Reeves 93, Díaz et al 96] ongi deskribatuta dauden AGetan, problema konplexuentzat soluzio hobek bilatzeko aurrez existitzen diren soluzioen atalak konbinatzea proposatzen da. AGak beraz, genetika eta selekzio naturalaren mekanikan oinarritutako teknika moduan defini genitzake. Ezaugarri nagusiak malgutasuna eta problema-multzo zabalean erabilgarri izatea dira. Ikerkuntza operatiboaren arloan, algoritmo genetikoaren ideia «*ausazko bilaketaren esplotazio adimentsu*» gisa uler daiteke [Reeves 93].

Beste teknika metaheuristikoen desberdina da, iterazio bakoitzean soluzio bakarra sortu ordez, soluzio-populazio batekin lan egiten duelako, eta eboluzio biologikoak memoriarik ez duenez, ez du testuinguruko informazioa erabiltzen soluzio berriak sortzeko.

Prozesu ebolutibo osoan zehar kromosoma populazio bat mantentzen da. Bi eragiketa nagusi erabiltzen dira soluzio berriak sortzeko: gurutzaketa eta mutazioa. Kromosoma bakoitzari balio bat dagokio, *fitness* izena ematen zaiona eta sistemaren soluzioaren kalitateari buruzko informazioa ematen du. Balio hau, optimizatu nahi den funtzioaren balioa edo beste balioren bat izan daiteke. Oinarritzko AG baten eskema 4 Algoritmoan aurkezten da.

Hasierako populazioa sortu

Populazioko elementu bakoitzarentzat fitness-a kalkulatu

Errepika

Gurutzaketarako elementu bikoteak hautatu

Probabilitate batez gurutzatu

Soluzio berriak mutatu probabilitate batez

Elementu berri bakoitzarentzat kalkulatu fitness balioa

Sartu elementu berriak belaunaldi berriko populazioan

Harik eta populazioaren konbergentzia.

4. Algoritmoa. Oinarrizko AG bati dagokion eskema.

Bide-hautaketa problemak ebazteko erabiltzen diren AG-ak ere topa daitezke bibliografian; adibidez, [Tangiah 95, Potvin et al 96b, Zhu 00].

3.3.3.2. Beste metodo ebolutibo batzuk

Badaude algoritmo genetikoaz gain beste zenbait metodo ebolutibo ere:

Programazio Genetikoak, [Kozs 92]-n aurkezten denak, Algoritmo Genetikoaren paradigma zabaltzen du linealak ez diren egituretara. Helburua, problema jakin bat ebazteko gai den programa topatzea da eta, bilaketa, programa posibleen espazioan egiten da; populazioko elementuak programak izango dira hain zuzen.

Programazio Ebolutiboa [Fogel et al 66]-k aurkeztutako ideietan oinarritzen da optimizazio konbinatorioko problemak ebazten saiatzeko. Algoritmo horien egitura Algoritmo Genetikoaren egituraren parekoa da. Diferentzia nagusia, elementu berriak sortzeko mutazioa soilik erabiltzen dela da, hau da, ez daudela gurutzaketa eragilerik. Hau dela eta, Programazio Ebolutiboa sexurik gabeko Algoritmo Genetiko bat dela esan daiteke.

Azkenik, Estrategia Ebolutiboak, zenbakizko optimizazioko problemak ebazteko erabili izan dira [Schwefel 81]. Ez da gurutzaketa erabiltzen, ebolutzioa kopia eta mutazio bidez lortzen da. AG-ekin dituzten diferentzia nagusien oinarria problemaren adierazpidea eta bilaketa egiteko erabilitako eragileak dira. Oro har, Estrategia Ebolutiboek ez dituzte elementuak kodetzen eta problemaren soluzioen gainean egiten dute bilaketa. [Hombberger et al 99]k metodo hau erabiltzen duten bi sistema aurkezten dituzte eta [Bräysy et al 01]-ek beste bat.

4. OPTIMIZAZIO KONBINATORIOKO PROBLEMEN ADIBIDE BAT, DATU-MULTZOA ETA ZENBAIT EMAITZA

Sarreran aipatu den moduan, guk egindako azterlan nagusia *VRPTW* problemaren ingurukoa izan da. Hain zuzen ere, problema-mota hau ebazteko gai den sistema paraleloa eraiki dugu oinarri gisa *Simulated Annealing* metaheuristika eta ibilbideen eraikuntza metodo baten konbinaketa erabiliz [Arbelaitz et al 01a, Arbelaitz et al 01b, Arbelaitz et al 01c, Arbelaitz et al 01d] lanetan ikus daitekeen moduan. Mota honetako sistemen eraginkortasuna neurtzeko erabilitako oinarri esperimentalak, hau da, sistema batzuen emaitza beste batzuenarekin konparatzeko erabilitako datu-multzoa Solomon-ek 1983an proposatutakoa da.

Solomon-ek aurkezten duena, denborazko baldintzak bete behar dituen ibilgailuen bide-hautaketako problema-multzoa da. Multzo honetako problemek 100 eskaera dituzte, bakoitza bezero desberdin bati dagokiona, eta 25 ibilgailuko multzoa eskaera hauei aurre egiteko. Problema guztietan bezero-bikote guztien artean bidea dagoela suposatuko da, bi bezeroren arteko distantzia eta batetik bestera joateko bidea berdina izango direlarik.

Solomon-ek aurkeztutako problema-multzoa 6 taldetan banatutako 56 problemaz osatutako multzoa da: C1(9), C2(8), R1(12), R2(11), RC1(8) eta RC2(8) (parentesi artean ageri den zenbakiak talde bakoitzeko problema-kopurua adierazten du). Problemen ezaugarriak aldatu egiten dira multzoka. Aldatzen diren ezaugarriak, datu geografikoak, ibilgailu bakoitzak zerbitzatzen duen bezero-kopurua, denborazko baldintzak eta abar izanik. Banaketa geografikoari dagokionean, C1 eta C2 multzoetako problemetan bezeroak 10 zonalde desberdinetan bilduta daude. R1 eta R2 taldeetako problemetan bezeroak ausaz sakabanatuta daude eta azkenik, RC1 eta RC2 multzoko problemak aurrekoen nahasketa dira, bezero batzuk eskualdeka bilduta daude eta beste batzuk ausaz banatuta. Bezeroen kokaguneak berdinak dira mota jakin bateko problema guztientzat, hau da, berdinak dira R-ko problemetzat, RC-koentzat eta C-koentzat.

Ibilgailu bakoitzak zerbitzatzen duen bezero-kopuruari dagokionean, C1, RC1 eta R1 taldeetako problemek malgutasun gutxi dute eta ibilbide bakoitzean bezero gutxi zerbitza daitezke (5 eta 10 artean); C2, RC2 eta R2 taldekoek aldiz, malgutasun gehiago dute eta 30 bezero arte zerbitza daitezke ibilgailu batekin.

Talde bakoitzaren barruan problemen arteko desberdintasun nagusiak denborazko baldintzetan oinarritzen dira.

Banaketa eta baldintza desberdinak direla eta, talde bakoitza ebazteko zailtasuna eta metodorik egokiena aldatu egiten dira. Azpimarratzekoa da, lortzen diren emaitza-motagatik, lehenengo multzoko problemak mundu

errealean agertzen diren banaketa-problemen antzekoagoak direla eta, iker-tzaile gehienek beren lana multzo hauetara mugatu dute.

Jarraian aurkeztuko ditugu problema hauentzat artikulu honetan zehar aipatutako metodo metaheuristikoak erabiliz lortu diren emaitzarik onenak. Autore gehienek problema bakoitzerako topatu duten baliorik onena beste-rik ez dute ematen, kasu askotan, emaitza horiek lortzeko behar izan duten denbora eta saiakera kopuruaren aipamenik ere egiten gabe. Hori dela eta, 1 Taulan metodo bakoitzarentzat autore desberdinek lortutako soluzio one-nen batezbestekoa aurkezten dugu C1, RC1 eta R1 taldeetarako.

1. taula. Metahueristikak erabiliz lortu diren emaitza onenen laburpena.

Metodoa	Egileak	C1		RC1		R1	
		Ibilgailu	Distantzia	Ibilgailu	Distantzia	Ibilgailu	Distantzia
Tabu bilaketa	[Rochat et al 95]	10,00	828,45	12,38	1369,48	12,58	1197,42
	[Potvin et al 96a]	10,0	856,6	12,6	1465,0	12,6	1294,7
	[Chiang et al 97]	10	828,38	11,875	1397,44	12,167	1204,19
	[Schulze et al 99]	10,00	828,94	11,75	1409,25	12,25	1239,15
Metodo ebolutiboak	[Tangiah 95]	10	892,11	12,5	1474,13	12,75	1300,25
	[Potvin et al 96b]	10	838,8	12,1	1446,2	12,6	1296,8
	[Zhu 00]		867,36		1543,77		1352,77
	[Homberger et al 99] ES1	10	828,38	11,63	1392,57	11,92	1228,06
	[Homberger et al 99] ES2	10	828,38	11,50	1406,58	12,00	1226,36
	[Bräysy et al 01]	10	828,38	12,13	1372,20	12,42	1213,86
SA	[Chiang et al 96]	10,0	909,8	12,38	1473,9	12,5	1308,82

5. ONDORIOAK

Nabarmena da optimizazio konbinatorioko problemak ebazteko, eta ze-hazkiago, bide-hautaketakoak, metodo ugari erabil daitezkeela. Kasu bakoitzerako metodorik egokiena problemaren egituraren menpe egon arren, esan daiteke problema errealek ebazteko, metodo metaheuristikoak direla egokienak: lortzen duten emaitza optimoa dela ziurtatu ezin den arren, kalitate oneko emaitzak lortzen dituzte; erabiltzen duten denbora onargarria da, eta oro har, metodo zehatzak baino malguagoak dira sistema errealean aldaketak islatzeko garaian.

Azken emaitzatatik ondoriozta daiteke metodo metaheuristiko guztiak direla antzeko mailako emaitzak lortzeko gai, parametroak eta auzo-egitura modu egokian hautatzen badira behintzat.

Azterketa hau oinarri izanik, erregai banaketarako ibilbide egokiak bi-latzen dituen sistema erreala eraikitzerakoan, metodo metaheuristikoetan oinarritu gara, metodo zehatzak hasieratik baztertuz.

6. BIBLIOGRAFIA

- [Arbelaitz et al 01a] ARBELAITZ, O., RODRÍGUEZ, C., ZAMAKOLA, I., «Analysis of the Construction of a New Fast System to Solve VRPTW Problems, Based on Simulated Annealing». *Fourth International ICSC Symposium on Soft Computing and Intelligent Systems for Industry (ISFI'2001)*, Paisley.
- [Arbelaitz et al 01b] ARBELAITZ, O., RODRÍGUEZ, C., ZAMAKOLA, I., «Study of the Robustness of a New Algorithm Based on Simulated Annealing to Solve VRPTW Problems». *Second European Conference on Intelligent Management Systems in Operations*, Salford, Ingalterra, 2001, pp. 124-133.
- [Arbelaitz et al 01c] ARBELAITZ, O., RODRÍGUEZ, C., ZAMAKOLA, I., «Comparison of Systems Based on Evolutionary Search and Simulated Annealing to Solve the VRPTW Problem». *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation CIMCA'2001*, Las Vegas, USA, 2001, pp. 237-246.
- [Arbelaitz et al 01d] ARBELAITZ, O., RODRÍGUEZ, C., ZAMAKOLA, I., «Low Cost Parallel Solutions for the VRPTW Optimization Problem». *International Conference on Parallel Processing Workshop on High Performance Scientific and Engineering Computing with Applications*, Valencia, 2001, pp. 176-181.
- [Bodin et al 83] BODIN, L., GOLDEN, B., ASSAD, A., «Routing and Scheduling of Vehicles and Crews. The State of the Art». *Computers and Operations Research*, Vol 70, No 2, pp. 63-211.
- [Bodin 90] BODIN, L.D., «Twenty Years of Routing and Scheduling». *Operations Research*, July-August 1990, Vol 38, n.º 4.
- [Bräysy 01] BRÄYSY, O., «Genetic Algorithms for the Vehicle Routing Problem with Time Windows». *Arpakannus 1/2001*. Special issue on Bioinformatics and Genetic Algorithms, pp. 33-38.
- [Cerny 85] CERNY, V., «Thermodynamical Approach to the Traveling Salesman Problem: an Effective Simulation Algorithm». *Journal of Optimization Theory and Applications* 45, pp. 41-51.
- [Chiang et al 96] CHIANG, W.C., RUSSEL, R.A., «Simulated Annealing Metaheuristics for the Vehicle Routing Problem with Time Windows». *Annals of Operations Research* 63, 1996, pp. 3-27.
- [Chiang et al 97] CHIANG, W.C., RUSSEL, R.A., «A Reactive Tabu Search Metaheuristic for the Vehicle Routing Problem with Time Windows». *INFORMS Journal on computing*, fall 1997, Vol 9, n.º 4, pp. 417-430.
- [Clarke et al 64] CLARKE, G., WRIGHT, W., «Scheduling of Vehicles from a Central Depot to a Number of Delivery Ppoints». *Operations Research* 12, 1964, pp. 568-581.
- [Dantzig et al 74] DANTZIG, G.B., EAVES, B.C., *Studies in Optimization*. The Mathematical Association of America (incorporated) Library of Congress Catalog Card Number 74-21481.

- [Díaz et al 96] DÍAZ, A., GLOVER, F., GHAZIRI, H.M., *Optimización Heurística Redes Neuronales*. Editorial Praninfo, 1996. ISBN 84-283-2269-4.
- [Fisher et al 97] FISHER, M.L., JÖRNSTEN, K.O., MADSEN. «Vehicle Routing with Time Windows: two Optimization Algorithms». *Operations Research*, May-June 1997, Vol 45, n.º 3.
- [Fisher 85] FISHER, M.L., «An Application Oriented Guide to Lagrangean Relaxation». *Interfaces*, Vol 15, pp. 10-21.
- [Fogel et al 66] FOGEL, I.J., OWENS, A.J., WALSH, M.J., *Artificial Intelligence through Simulated Evolution*. Wiley, New York
- [Glover 90] GLOVER, F., «Tabu Search: Part II». *ORSA Journal on computing* 2, pp. 4-32.
- [Goldbreg 89] GOLDBERG, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [Janssens et al 94] JANSSENS, G.K., BREEDAM, A.V., «A Simulated Annealing Postprocessor for the Vehicle Routing Problem». *Applications of Modern Heuristic Methods*, Chapter 11, pp. 175-191.
- [Holland 75] HOLLAND, J., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor 1975.
- [Hombberger et al 99] HOMBERGER, J., GEHRING, H. «Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows». *INFORMS Journal on Computing*, 1999, 37 (3), pp. 297-318.
- [Kirkpatrick et al 83] KIRKPATRICK, S., GELLAT, C.D., VECCHI, M. «Optimization by Simulated Annealing». *Science*, 220, pp. 671-680.
- [Koza 92] KOZA, J.R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.
- [Lin 65] LIN, S., «Computer Solutions of the Traveling Salesman Problem». *Bell systems, technical journal* 44, 1965, pp. 2245-2269.
- [Metropolis et al 53] METRÓPOLIS, N., ROSEMBLUTH, A.W., «Equation of State Calculation by fast Computing Machines». *Journal of Chemical Physics* Vol 21, n.º 6, pp. 1087-1091.
- [Michalewicz et al 00] MICHALEWICZ, Z., FOGEL, D.B., *How to solve it: Modern Heuristics*. Springer Verlag, Berlin Heidelberg 2000.
- [Or 76] OR, I., *Traveling Salesman-Type Combinatorial Problems and their Relation to the Logistics of Blood Banking*. Tesis doctoral, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston.
- [Potvin et al 95] POTVIN, J.Y., ROUSSEAU, J.M., «An Exchange Heuristic for Routing Problems with Time Windows». *Journal of the Operational Research Society*, 46, 1995, pp. 1433-1446.
- [Potvin et al 96a] POTVIN, J.Y., KERVAHUT, T., ROUSSEAU, J.M. «The Vehicle Routing Problem with Time Windows Part I: Tabu Search». *INFORMS Journal on Computing*, spring 1996, Vol 8, n.º 2, pp. 158-164.
- [Potvin et al 96b] POTVIN, J.Y., BENGID, S., «Vehicle routing problem with time windows. PartII: Genetic Search». *INFORMS journal on Computing*, Spring 1996, Vol 8, n.º 2.
- [Reeves 93] REEVES, C. R., *Modern Heuristic Techniques for Combinatorial problems*. Halsted Press, John Wiley & Sons, ISBN 0-470-22079-1.
- [Rochat et al 95] ROCHAT, Y., TAILLARD, É.D., «Probabilistic Diversification and intensification in local search for vehicle routing». *Journal of Heuristics* 1, 1995, pp. 147-167.

- [Schulze et al 99] SCHULZE, J., FAHLE, T., «A Parallel Algorithm for the Vehicle Routing Problem With Time Window Constraints». *Annals of Operations Research*, 1999, n.º 86, pp. 585-607.
- [Schwefel 81] SCHWEFEL, H.P., *Numerical Optimization for Computer Models*. Willey, Chichester, UK.
- [Schwefel 95] SCHWEFEL, H.P., *Evolution and Optimum Seeking*. John Wiley and Sons, New York.
- [Solomon 83] SOLOMON, M.M., *Solomon's benchmark*. Página web de Solomon M.M, http://www.cba.neu.edu/_msolomon/.
- [Solomon 87] SOLOMON, M.M. «Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints». *Operations Research*, March-April 1987, Vol 35, No 2, pp. 254-265.
- [Taillard et al 95] TAILLARD, E., BADEAU, P., GENDREAU, M., GUERTIN, F., POTVIN, J.Y., *A new Neighborhood Structure for the Vehicle Routing Problem with Time Windows*. Technical report Université de Montreal. CRT-95-66.
- [Thangiah 95] THANGIAH, S.R., «Vehicle Routing with Time Windows using Genetic Algorithms». *Application Handbook of Genetic Algorithms: New Frontiers*. Volume II, Lance Chambers (ed), CRC press 253-277.
- [Zhu 00] ZHU, K.Q. «A New Genetic Algorithm for VRPTW». *Proceedings of IC-AI 2000*, Las Vegas, USA.