

Orientazio problema eta algoritmo genetikoak

Gorka Kobeaga Urriolabeitia*¹, María Merino Maestre²

¹ BCAM (Internship)
Matematikaren eta Zientzia Esperimentalen Didaktika,
Bilboko Irakasleen Unibertsitate Eskola (UPV/EHU)

² Matematika Aplikatua eta Estatistika
eta Ikerkuntza Operatiboa, Zientzia eta Teknologia Fakultatea (UPV/EHU)

* gorka.kobeaga@ehu.eus

DOI: 10.1387/ekaia.14540

Jasoa: 2015-05-29

Onartua: 2015-09-01

Laburpena: Artikulu hau UPV/EHUko *Modelizazio eta Ikerkuntza Matematika, Estatistika eta Konputazioa* izeneko masterrean Gorka Kobeagak defendatutako master amaierako lanean oinarrituta dago. Orientazio Problemaren (OP) aurkezpena eta ebazpen ebolutiboaren analisisa du helburu. Saltaile Ibiltariaren problema ezagunaren aldaera berezi bat da OP. Optimizazio Konbinatorioko problema honen aplikazioak eta modelizazioak aurkezten dira. OPren konputaziozko konplexutasuna dela eta, *Concorde TSP solver* erabiltzen duen algoritmo genetiko bat diseinatu eta inplementatu da. Gainera, herri kopuruarekiko eta distantzia mugarekiko portaera eta prozedura genetikoen parametroekiko konbergentzia aztertu da. Euskal Autonomia Erkidegoko eta literatura zientifikoko datu-baseak erabili dira.

Hitz-gakoak: Algoritmo Genetikoak, CPLEX, Concorde, EAE, TSP, Optimizazio Konbinatorioa, Orientazio Problema.

Abstract: This article is based in the Master Thesis defended by Gorka Kobeaga on the *Mathematical Modeling and Research, Statistics and Computing* Master from UPV/EHU. Its goal is to present the Orienteering Problem (OP) and its evolutionary solving. OP is a variant of the well known Travelling Salesman Problem. Applications and modeling of that Combinatorial Optimization problem are shown. Because of OP's computational complexity, a genetic algorithm using *Concorde TSP solver* has been designed and implemented. Moreover, its behaviour with respect to the number of towns and distance limit and the convergence analysis depending on the genetic algorithms' parameters are detailed. It has been used data base from Basque Country and scientific literature.

Keywords: Genetic Algorithms, CPLEX, Concorde, Basque Country, TSP, Combinatorial Optimization, Orienteering Problem.

1. SARRERA

Optimizazio Konbinatorioko problema ezagunenetariko bat Saltzaile ibiltariaren problema (*Travelling Salesman Problem, TSP*) da. Herri multzo batetik igarotzen den ibilbidea zehaztean datza, lehenengo herrian hasi eta amaituz, herri bakoitza behin bakarrik bisitatuz eta ibilbidearen kostua minimizatuz. Lan honen ardatza TSPren aldaera den Orientazio Problema (*Orienteering Problem, OP*) izan dugu. Ibilbidearen luzera maximoaren murrizketa errespetatuz, herri bakoitzean jasotako sarien batura maximizatzea helburu duen ibilbide-problema da. [1]-en OPren eta bere aldaeren laburpen bat aurki daiteke.

Pentsa dezagun hurrengo mendi-lasterketan parte hartuko dugula. Bi egunetako denborarekin gailurrak egin behar ditugu Euskal Herriko mendietan. Mendi bakoitzak emango digun puntuazioa metrotan duen garaiera da eta lasterketaren garailea denbora agortu ostean puntu-kopuru handiena lortu duena izango da. Zein ibilbide egingo genuke mendi guztiak igotzeko denborarik ez dugula jakinda? Pirinio inguruetako mendiak igoko genituzke, garaiera handiagokoak baina gailurra egiteko denbora gehiago eskatzen ditutenak? Ala, kostaldeko mendietatik egingo genuke ibilbidea, txikiagoak baina denbora gutxian gailur asko egiteko aukera ematen ditutenak?

Problema hau lehenengo aldiz [2] liburuan agertu zen, kamioi talde batek erregai-banaketa egokia egiteko aukeratu behar dituen ibilbideen problemari lotuta. Hala ere, izena [4] artikulutik hartu zuen: artikulua horretan gonbidatu gaituzten antzerako kirol-lasterketa bat planteatzen da; denbora muga baten barruan ingurune batean banatuta dauden kontrol puntu kopuru handienetik igaro behar den lasterketa bat hain zuzen.

Orientazio problemak izen desberdinak hartu izan ditu: *bilduma handienaren problema* eta *banku lapurraren problema*, besteak beste. Antzerako problema batzuk modelizatzeko ere erabili izan da OP; adibidez, *ibilgailuen bideraketa problema* (*vehicle routing problem, VRP*) [2], produkzioaren antolaketa [5] eta turista bidaien antolaketa [6], [7], [8] eta [9].

Lan hau honako moduan dago antolatua: 2. atalean Orientazio Problemaren aplikazioak, zailtasuna eta modelizazio batzuk agertzen dira. 3. atalean, CER izeneko algoritmo genetikoa azaltzen da eta haren konputaziozko esperientzia 4. atalean erakusten da. Azkenik, ondorioak eta etorkizunerako proposamenak agertzen dira.

2. ORIENTAZIO PROBLEMA

2.1. Aplikazioak

Azken urteetan, ugaritu egin dira OPren aplikazioak, teknologia berriek eskainitako aukerekin batera; batez ere, mugikorretan interneten erabilera

zabalduta eta paketeen banaketarako eta inguruaren monitorizaziorako robotikako elementuen erabilera aztertzen hasi direnetik.

- Herri guztiak bisitatzeko astia ez duen saltzaile bidaiariarena. Saltzaileak badaki herri bakoitzean egin dezakeen salmenta kopurua eta, egun (edo aste) bateko denbora mugarekin, salmenta kopuru totala maximizatu nahi du; ikusi [10].
- Etxeetako erregaiaren banaketaren problema: multzo handi bateko bezeroen artean kamioi talde batek erregaia modu eraginkorren banatzeko bezeroen azpi-multzoa aukeratzean datza, bezeroen erreserba-aurreikuspenen arabera lehentasunak definituz; ikusi [2].
- Sakeleko telefonodun turistaren gida. Herri bat bisitatzera doan turista batentzat ia ezinezkoa da leku interesgarri guztiak bisitatzera eta gogokenak izango diren ikustekoak aukeratu beharko ditu. Antolaketa-problema horri, egonaldiaren iraupena kontuan izanik, *Tourist Trip Design Problems* (TTDP) deitzen zaio; ikusi [1]. Segundu gutxiren barruan kalitate handiko soluzioak eskatzen ditu.
- Espedizioaren problema. Baliabide mugatutun (bateria, erregai, janari) robot, ibilgailu edo pertsona baten espedizioa antolatzea. TTDP problemaren antzekoa da. Batu beharreko paketeak edo egin beharreko argazkiak maximizatzea izan daiteke helburua; ikusi [11].

2.2. Konputaziozko konplexutasuna

J. Edmonds matematikariak algoritmo ona-ren definizioa proposatu zuen; ikusi [3]. Teoria garatzeko, erabaki-problema, bai edo ez erantzuna dutenak, aztertu zituen bakarrik. Izan ere, edozein optimizazio-problema erabaki-problema baten moduan berridatzi daiteke beti. Kasu horretan esaten da, algoritmo bat ona dela baldin eta denbora polinomialean ebatzi badaiteke; hau da, exekuzio ordena $O(n^k)$ bada, n sarrerako tamaina eta k konstante independente bat izanik. Mota honetako algoritmoak dituzten erabakitze-problema P motako problema moduan sailkatzen dira. Bestetik, denbora polinomialean era ez deterministikoan ebazten diren problema NP multzoan sartzen dira; hau da, paraleloan exekuzioak egin ditzakeen Turingen makina batean denbora polinomikoan ebazgarriak diren problema. NP multzoko problema bat, X , NP-*osoa* motako problema bat dela esaten da, baldin eta NPko edozein problema X -ra murriztu baldin badaiteke denbora polinomial batean. Problema bat, V , NP-*gogorra* motako problema bat dela esaten da baldin eta NPko edozein problema V -ra denbora polinomikoan murriztu badaiteke.

OP, TSPren aldaera bat izanik, NP-gogorra klaseko problema bat da. Hori ikusteko, har dezagun TSP bere erabakitze-bertsioan, hau da, herrien multzo bat eta d_{max} zenbaki osoa emanik, hurrengo galderari erantzuten diona: existituko ote al da herri guztietatik behin pasatzen den eta d_{max}

baino luzera laburragoa duen ibilbiderik? Ezaguna da TSP erabakitze bertsioan NP-osoak klaseko problema dela. Moldatu dezagun erabakitze-problema hau OPra, eman diezaiogun 1 balioko saria herri bakoitzari eta aukeratu ditzagun hasierako eta amaierako herriak. Orduan, OPren emaitza herri kopururaren berdina bada, izan bedi TSP erabakitze bertsioaren erantzuna bai eta bestela ez. TSP erabakitze-bertsioan NP-osoak klaseko problema bat denez eta OPra denbora polinomialean moldatzea lortu dugunez, NP-ko edozein problema OP-ra denbora polinomialean moldatu daiteke eta, beraz, OP NP-gogorra multzokoa da; ikusi [2].

2.3. Modelizazioa

Orientazio problema $\langle G, d, s, v_1, v_n, d_0 \rangle$ tuplak definitzen du. $G = \langle V, E \rangle$ adierazpena $V = \{v_1, \dots, v_n\}$ erpinak eta E ertzak dituen grafoa da, $d: V \times V \rightarrow \mathbb{R}^+ \rightarrow \{0, \infty\}$ aplikazioak erpinen arteko distantzia (zeharkatzeko denbora edo kostua) adierazten du, $d(v_i, v_j) = d_{ij}$ delarik, eta $s: V \rightarrow \mathbb{R}^+$ aplikazioak erpin bakoitzeko saria adierazten du, $s(v_i) = s_i$ izanik. Izan bedi *ibilbidea* ertz guztiak desberdinak dituen erpinen segida; zikloa ibilbide itxia, hau da, hasiera eta amaierako erpina bera duen ibilbidea; eta *azpi-ibilbidea* ibilbidearen erpin guztiak ez dituen ibilbidea. OPren soluzioa v_1 -en hasi eta v_n bukatzen den eta luzera totalak d_0 gainditzen ez duen erpinen azpimultzoaren *ibilbide hamiltondar* bat da (hau da, erpin guztietatik behin igarotzen dena). OPren optimoa aurkitzea erpinetako sarien batura maximizatzen duen soluzioa aurkitzean datza.

Erpinetan denbora-galera bat badago, hau da, v_i bakoitza bisitatzeko t_i denbora behar bada, OPren baliokide batera molda dezakegu erpinetako denbora aldeetan txertatuz. v_i erpinaren denboraren erdia gehitu genezaiok lotzen duen erpin bakoitzari. Izan ere, erpin bakoitzerako batean sartu eta beste batetik irtengo garenez, hasierako problemaren baliokidea izango da.

Normalean hasierako eta amaierako erpinak, v_1 eta v_n , desberdinak dira. Berdinak direnean *Bira Orientazio Problema (Orienteering Tour Problem, OTP)* deitu izan da baina oso antzekoak dira bi problemak. Izan ere, beti gehi dezakegu v_1 eta v_n -ren arteko denbora zero balio duen alde lagungarri bat. Orokortasunik galdu gabe, $v_1 = v_n$ hartuko dugu eta k elementudun ziklo nagusi bakarra eta $n - k$ itzuliko V multzoaren permutazioak izango dira bilatuko ditugun ibilbideak.

Defini dezagun OP programazio lineala erabiliz. Horretarako, izan bedi \mathcal{N} , herrien multzoa, $\mathcal{N} = \{1, 2, \dots, n\}$, $|\mathcal{N}| = n$, $\mathcal{N}^- = \mathcal{N} \setminus \{1\}$. Ohar gaitzen V eta \mathcal{N} multzoen artean korrespondentzia bijektibo bat existitzen dela, eta v_i erpina adierazteko i ere erabiliko dugu. Eta aintzat har ditzagun ondoko elementuak:

Parametroak

s_i , i herria betetzeagatik jasotzen den saria, $i \in \mathcal{N}$.

d_{ij} , i herritik j herrirako kostua (distantzia, denbora, dirua), $i, j \in \mathcal{N}$.

d_0 , amaierako ibilbidearen kostu maximoa.

Aldagaiak:

x_{ij} , bitarra, 1 baldin eta i herritik j herrirako bidaia egiten bada, $i, j \in \mathcal{N}$.

x_i , bitarra, 1 baldin eta i herria bisitatzen bada, $i \in \mathcal{N}$.

OPren soluzio optimoa ibilbide osoaren saria maximizatzean datza:

$$\max \sum_{i=1}^n s_i x_i \quad (1a)$$

Erpin bat bi aldiz bisitatzek ez dakarrenez onurarik, erpin bakoitzetik ibilbidea behin bakarrik sartzea eta irtetzea komeni da: (1b) eta (1c) murrizketekin bidez adierazten dira hurrenez hurren,

$$\sum_{i=2}^n x_{i_1} = x_1 = 1, \quad \sum_{\substack{i=1 \\ i \neq j}}^n x_{ij} = x_j \quad 2 \leq j \leq n \quad (1b)$$

$$\sum_{j=2}^n x_{1j} = x_1 = 1, \quad \sum_{\substack{j=1 \\ j \neq i}}^n x_{ij} = x_i \quad 2 \leq i \leq n \quad (1c)$$

Izan ere, lehenengo herritik ateratzen da eta heltzen da behin, $x_1 = 1$. Beste edozein herritatik bi aukera daude: ez pasatzea, $x_i = 0$ edo pasatzea, $x_i = 1$, baina behin helduz eta behin ateraz bertatik.

Bestetik, ibilbidearen gehieneko distantziaren baldintza kontuan hartu behar da.

$$\sum_{i=1}^n \sum_{\substack{i=1 \\ j \neq i}}^n d_{ij} x_{ij} \leq d_0 \quad (1d)$$

(1b) eta (1c) ekuazioek ez dute ziurtatzen azpi-ibilbideak egongo ez direnik eta murrizketa berriak sartu behar dira, (1e) motako murrizketak hain zuzen ere.

$$\text{ibilbideak ziklo bakarra du} \quad (1e)$$

(1a) helburu funtzioak, (1b-1e) murrizketek eta aldagai motak problema lineala definitzen dute, hau da, OPren modelizazio orokorra.

TSPn azpi-ibilbideak ezabatzeko murrizketak OPra moldatzen saiatuko gara. Ibilbide batek OPn azpi-ibilbidea duela esango da baldin eta ibilbidea, $\{v_i \in V : x_i = 1\}$ multzoko erpinetara murriztuz, ibilbide hamiltondarra ez bada. Azpi-ibilbideak ezabatzeko (1e) murrizketan era desberdinetara adierazi daitezke. Jarraian hiru modelizazio ikusiko dira.

2.3.1. OPren DFJ modelizazioa

Azpi-ibilbideak ekiditeko (1e) baldintzak adierazteko aurkeztuko dugun lehenengo modelizazioa *Clique Packing Constraints* izeneko murrizketetan oinarrituta dago (Datzing, Fulkerson eta Johnsonek 1954an argitaratua); ikusi [12]. DFJ deituko ditugun murrizketek OPrako balio digute, (2a)-(2f) modelizazioa osatuz.

$$\max \sum_{i \in \mathcal{N}} s_i x_i \quad (2a)$$

$$\text{h.b.: } \sum_{j \in \mathcal{N} \setminus \{i\}} x_{ij} = x_i, \quad i \in \mathcal{N} \quad (2b)$$

$$\sum_{i \in \mathcal{N} \setminus \{j\}} x_{ij} = x_j, \quad j \in \mathcal{N} \quad (2c)$$

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \setminus \{i\}} d_{ij} x_{ij} \leq d_0, \quad (2d)$$

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} x_{ij} \leq |S| - 1, \quad \emptyset \neq S \subset \mathcal{N}^- \quad (2e)$$

$$x_{ij} \in \{0, 1\}, x_i \in \{0, 1\}, x_1 = 1, \quad i \neq j, i, j \in \mathcal{N} \quad (2f)$$

Ohar gaitezen azpi-ibilbideak badaude murrizketa ez dela betetzen. Izan bedi (i_1, \dots, i_k) azpi-ibilbidea, $1 \notin S = \{i_1, \dots, i_k\}$; beraz, $S \subset \mathcal{N}^-$. Bestetik v_{i_1}, \dots, v_{i_k} ibilbidean daudenez, $\sum_{i \in S} x_i = |S|$. (2b) baldintzak erabiliz:

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} x_{ij} = \sum_{i \in S} \sum_{j \in \mathcal{N}^- \setminus \{i\}} x_{ij} = \sum_{i \in S} x_i = |S|$$

Lehenengo berdintzaren partekotasun bat berehalakoa da. Beste partekotasunerako, ohartu $x_{ij} = 0$ dela $i \in S$ eta $j \in \mathcal{N} \setminus S$ guztietarako.

Orain ibilbide batek azpi-ibilbiderik ez badu, murrizketak betetzen dituela ikusiko dugu. Izan bedi $\emptyset \neq S \subset \mathcal{N}^-$ azpimultzoa. Ohar gaitezen baldin eta existitzen bada $i_0 \in S$ non $x_{i_0} = 0$ den, orduan

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} x_{ij} \leq \sum_{i \in S} \sum_{j \in \mathcal{N}^- \setminus \{i\}} x_{ij} = \sum_{i \in S} x_i = \sum_{i \in S \setminus \{i_0\}} x_i \leq |S| - 1$$

betetzen dela. Beraz, $\forall i \in S \ x_i = 1$ dela suposatuko dugu.

Hipotesiz azpi-ibilbiderik existitzen ez denez, orduan existitzen da $i_0 \in \mathcal{N}$, $x_{i_0 1} = x_{i_0} = 1$ dena. Argi dago edozein $S \subset \mathcal{N}^-$ non $i_0 \in S$ den, (2e) murrizketa betetzen dela. Bestalde, $i_0 \notin S$ bada $\mathcal{N}^- \setminus S \subset \mathcal{N}^-$ ez hutsa da eta existitzen dira $i_1 \in S$ eta $i_2 \in \mathcal{N}^- \setminus S$ $x_{i_1 i_2} = 1$ direnak. Beraz,

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} x_{ij} = \sum_{i \in S \setminus \{i\}} \sum_{j \in S \setminus \{i\}} x_{ij} \leq \sum_{i \in S \setminus \{i\}} \sum_{j \in \mathcal{N}^- \setminus \{i\}} x_{ij} = \sum_{i \in S \setminus \{i\}} x_i = |S| - 1$$

2.3.2. OPren FST modelizazioa

Hurrengo (3a)-(3e) modelizazioa Fischetti, Salazar-González eta Toth egileek 1998an argitaratutako OPren modelizazioa da; ikusi [13]. Ohar gaitzen modelizazioa bakarrik dela baliogarria norantza gabeko grafoentzat.

$$\max \sum_{i \in \mathcal{N}} s_i x_i \tag{3a}$$

$$\text{h.b.: } \sum_{j \in \mathcal{N} \setminus \{i\}} x_{ij} = 2x_i, \quad i \in \mathcal{N} \tag{3b}$$

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \setminus \{i\}} d_{ij} x_{ij} \leq d_0, \tag{3c}$$

$$\sum_{i \in S} \sum_{k \in \mathcal{N} \setminus S} x_{ik} \geq 2x_j, \quad 1 \in S \subset \mathcal{N}, j \in \mathcal{N} \setminus S \tag{3d}$$

$$x_{ij} \in \{0, 1\}, x_i \in \{0, 1\}, x_1 = 1, \quad i, j \in \mathcal{N} \tag{3e}$$

Baliokidea da (3d) murrizketa-multzoa eta (3f) sistema:

$$\sum_{k \in S} \sum_{i \in \mathcal{N} \setminus S} x_{ki} \geq 2(x_i + x_j - 1), \quad S \subset \mathcal{N}, 2 \leq |S| \leq n - 2, i \in S, j \in \mathcal{N} \setminus S \tag{3f}$$

Baliokidea da (3d) murrizketa-multzoa eta (3g)-(3h) multzoa

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} x_{ij} \leq \sum_{i \in S} x_i - x_j, \quad S \subset \mathcal{N}, 1 \in S, j \in \mathcal{N} \setminus S \tag{3g}$$

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} x_{ij} \leq \sum_{i \in S} x_i - x_i, \quad S \subset \mathcal{N}, 1 \in \mathcal{N} \setminus S, i \in S \tag{3h}$$

2.3.3. OPren MTZ modelizazioa

Azkenik MTZ deituko ditugun murrizketak moldatuko ditugu problema honetara. Miller, Tucker eta Zemlin-ek 1960an argitaratu zuten murrizketetan oinarrituta dago; ikusi [14]. Izan bitez $2 \leq u_i \leq n$ erpin bakoitzak ibilbidean duen posizioa adierazten duten aldagai jarraituak ($2 \leq i \leq n$). 4. atalean, MTZ modelizazioarekin, (4a)-(4f), izandako konputaziozko esperientzia erakutsiko dugu.

$$\max \sum_{i \in \mathcal{N}} s_i x_i \quad (4a)$$

$$\text{h.b.:} \sum_{j \in \mathcal{N} \setminus \{i\}} x_{ij} = x_i, \quad i \in \mathcal{N} \quad (4b)$$

$$\sum_{i \in \mathcal{N} \setminus \{j\}} x_{ij} = x_j, \quad j \in \mathcal{N} \quad (4c)$$

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \setminus \{i\}} d_{ij} x_{ij} \leq d_0, \quad (4d)$$

$$u_i - u_j + (n-1)x_{ij} \leq n-2, \quad i \neq j, i, j \in \mathcal{N}^- \quad (4e)$$

$$x_{ij} \in \{0, 1\}, x_i \in \{0, 1\}, x_1 = 1, u_i \in \mathbb{R} \quad i \neq j, i, j \in \mathcal{N} \quad (4f)$$

Ikus dezagun MTZ baldintzak OPrako baliogarriak direla. Bi kasu desberdinduko ditugu:

- (i) $x_i = 0$ edo $x_j = 0$

Orokortasunik galdu gabe eman dezagun $x_i = 0$ dela. Orduan $x_{ij} = x_{ji} = 0$ da (4b) baldintza bete behar delako. Beraz, $u_i - u_j + 1 \leq n - 1$ daukagu eta hau beti betetzen da $2 \leq u_i, u_j \leq n$ direlako. Ohartu emaitza berdina dugula $x_j = 0$ bada, (4c) baldintzagatik.

- (ii) $x_i = 1$ eta $x_j = 1$

Bi gauza gertatu daitezke: edo $x_{ij} = 0$ izatea edo $x_{ij} = 1$.

— Baldin eta $x_{ij} = 0$ bada, aurreko moduan $u_i - u_j + 1 \leq n - 1$ bete teko da $2 \leq u_i, u_j \leq n$ direlako.

— Baldin eta $x_{ij} = 1$ bada, $u_i - u_j + 1 \leq 0$. Baina, $x_{ij} = 1$ bada (hau da, j herria i herriaren hurrengoa bada), betetzen da, $u_j - u_i = (t+1) - t = 1$ delako. Demagun k luzerako ziklo bat existitzen dela, eta izan bedi (i_1, i_2, \dots, i_k) azpizikloa non $1 \notin S = \{i_1, i_2, \dots, i_k\} \subset \mathcal{N}$, orduan,

$$\left. \begin{array}{l} x_{i_1 i_2} = 1 \\ x_{i_2 i_3} = 1 \\ \vdots \\ x_{i_{k-1} i_k} = 1 \\ x_{i_k i_1} = 1 \end{array} \right\} \Rightarrow \left. \begin{array}{l} u_{i_1} - u_{i_2} + (n-1) \leq n-2 \\ u_{i_2} - u_{i_3} + (n-1) \leq n-2 \\ \vdots \\ u_{i_{k-1}} - u_{i_k} + (n-1) \leq n-2 \\ u_{i_k} - u_{i_1} + (n-1) \leq n-2 \end{array} \right\} \Rightarrow (n-1)k \leq (n-2)k$$

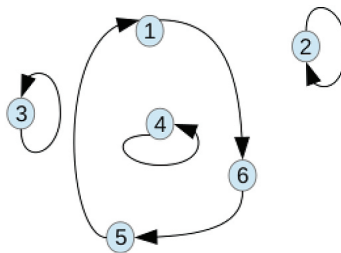
eta hau ezinezkoa da $n, k \geq 2$ direlako.

3. ALGORITMO GENETIKOAK

Atal honetan OPren ebazpenerako garatu dugun CER algoritmo genetikoa aurkeztuko dugu. Algoritmoari CER izena eman diogu Concorde softwarea [15] eta indibiduen gurutzaketarako TSPrako garatutako *Aldeen Berkonbinazioaren (Edge Recombination, ER)* bertsio moldatu bat erabili dugulako. Erabiliko dugun ER gurutzaketak problemaren d distantzia matricea simetrikoa izatea eskatzen du. Bereziki, distantzia euklidearra erabiliko dugu herrien arteko tartea neurtzeko. Concorde erabiltzeak exekuzioaren denboran eragin handia duenez 150-200 erpin dituzten problemetara mugatuko gara.

3.1. Indibiduoak: definizioak eta ezaugarriak

Prozedura zehaztu aurretik, oinarriko kontzeptuak azalduko ditugu. Algoritmo genetikoak *populazioetan* oinarritutako optimizazio-teknikak dira. Kasu honetan populazioko *indibiduoak* nodoen permutazioak dira non ziklo nagusi bakarra dagoen eta gainontzekoak itzuliak diren. Ikusi 1. irudian populazioko indibiduo baten adibidea.



1. irudia. Populazioko indibiduo baten adibidea.

Indibiduo bakoitzaren hurrengo ezaugarriak hartuko ditugu kontuan:

- *Genotipoa*: indibiduoaren errepresentazioa da, n elementuko permutazio baten bidez. Izan ere, genotipoa n -tupla batekin adieraziko dugu, i . posizioan i . herriaren hurrengo herria jarriz. Adibideko indibiduo hurrengo eran adieraziko dugu: (6, 2, 3, 4, 1, 5).
- *Ibilbidea*: honela deituko diogu permutazioko ziklo nagusiari. Adibidean, ibilbidea 1, 5 eta 6 herriek osatzen dute.
- *Asebetetzea (fitnessa)*: ibilbidean dauden herrien sarien batura da. Adibidean asebetetzea $s_1 + s_5 + s_6$ da.
- *Kostu totala*: ibilbide osoaren luzera da. Adibidean kostu totala $d_{16} + d_{65} + d_{51}$ da.
- *Kostu lokala*: ibilbideko erpin bakoitzak bere kostu lokala du; aurreko herritik dagoen distantziaren eta hurrengo herrira dagoenaren batura da, ibilbidean ez dauden herrien kostu lokala zero izanik. Adibidean 1. erpinaren kostu lokala $d_{16} + d_{51}$ da.

3.2. Osagaien eta eragileen deskribapena

Algoritmoan erabiltzen diren osagaiak eta eragileak deskribatuko ditugu. Problemaren definiziotik hurrengo elementuak ditugu: \mathcal{N} , erpin edo herrien multzoa; n , erpinen kopurua, $|\mathcal{N}| = n$; d , erpinen arteko distantzia matrizea; s , erpinetako sarien bektorea eta d_0 , distantzia maximoaren (1d) murrizketako goi-muga, $d_0 = \alpha \cdot v(TSP)$, non $\alpha \in [0, 1]$ proportzioa eta $v(TSP)$ TSP-ren soluzio optimoa diren.

Eboluzioa bideratuko duten sarrerako parametro batzuk erabiliko ditugu; haiek doituz algoritmoaren konbergentzia hobetzea bilatuko dugu:

- n_0 : belaunaldi kopurua, hots, algoritmoaren iterazio kopurua.
- n_{pop} : populazioaren tamaina.
- $ncand$: belaunaldi bakoitzean guraso izateko aukeratzen den hautagai kopurua.
- $pmut$: haur batek mutatzeko duen probabilitatea.
- β : kendu eragilean erpinak ordenatzerakoan sariak duen pisua; ikusi 3.2.3. azpiatala.
- $d2d$: kendu eragileen arteko belaunaldi kopurua.

Hemendik aurrera izendapen hauek ere erabiliko ditugu:

Haurra: bi ibilbide gurutzatuz lortzen den hirugarren ibilbidea.

Gurasoak: belaunaldi bakoitzean haurra sortuko duen ibilbide bikotea.

Eragileei dagokienez, batzuk (gehitu, kendu, gurutzaketa) OPrako espreski garatu baditugu ere, beste batzuk (mutazioa, gurasoen aukeraketa) testuinguru orokorrako batetik jaso ditugu. Jarraian banan-banan azalduko ditugu.

3.2.1. Hasierako populazioa

$npop$ indibiduo sortuko ditugu hasieran, eta populazioko indibiduo kopuru hau konstante eutsiko dugu prozesu osoan zehar. Indibiduo bakoitzean herri bakoitza ibilbidean egotea ala ez egotea ekiprobabletzat hartuko dugu. Behin ibilbidean egongo diren herriak aukeratuta tsp eragilea aplikatuko dugu. Hasierako populazioaren pseudo-kodea 1. algoritmoan adierazita dago.

1. algoritmoa **Hasierako populazioa**

```
for  $i = 1$  to  $npop$  do
   $v_1$  herria ibilbidean sartu;
  for  $j = 2$  to  $n$  do
    Ekiprobableki  $v_j$  ibilbidean badagoen edo ez erabaki.
  end for
  TSP eragilea aplikatu;
  Kostu lokala, kostu totala eta ibilbidearen asebetetzea gorde;
end for
```

3.2.2. Gehitu eragilea

Eragile honen zeregina, 2. algoritmoan laburbilduta, ibilbidearen luzeraren (1d) murrizketa betetzen den bitartean ibilbidera erpinak sartzea da. Horretarako, ibilbidean ez dauden erpinetan orden bat definituko dugu: ibilbidean ez dagoen erpin bakoitzarentzat ibilbidean dauden erpinetarako bi distantzia txikienen batura kontuan hartuz. Ez ditugu erpinen sariak kontuan hartuko, lan hori eboluzioaren esku utziko dugu. Adibidez, 1. irudian 2. erpinaren hurbiltasuna $i_2 = d_{21} + d_{26}$ eta 3. erpinarena $i_3 = d_{31} + d_{35}$ izango litzateke, eta $i_2 < i_3$ izango balitz, orduan 2. herria izango litzateke ibilbidean sartzeko hautagaia.

2. algoritmoa **Gehitu eragilea**

```
while not bukatu do
  Ordenatu ibilbidearen kanpoko erpinak hurbiltasun indizearen arabera; Aukeratu hurbilena ibilbidean sartzeko hautagai gisa;
  if hautagaia ibilbidean sartuta ibilbide osoaren luzeraren baldintza betetzen bada
  then
    Sartu hautagaia ibilbidean;
    Kostu lokala, kostu totala eta ibilbidearen asebetetzea eguneratu;
  else
    Bukatu;
  end if
end while
```

Behin ibilbidean ez dauden erpinak ordenatuta, hurbilen dagoena ibilbidean sartzen den ala ez aztertuko dugu, hau da, ea erpin berria ibilbidean sartuta ibilbide osoaren luzeraren baldintza betetzen den. Horrela bada, ibilbidean sartu eta prozesua errepikatuko dugu; bestela, amaituko dugu. Ohar gaitezen bukaeran lortuko dugun ibilbideak ez duela zertan ibilbiderik laburrena izan, baina bestela indibiduo bakoitzerako erpin bat gehituzerakoan TSP ebatzi beharko genuke eta hori konputazionalki garestiegi litzateke.

3.2.3. Kendu eragilea

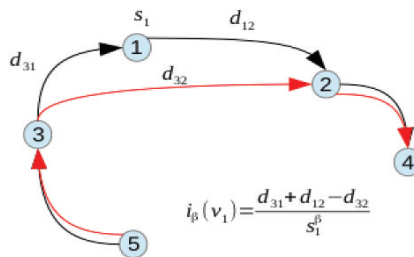
Eragile honen helburua distantziaren murrizketa betetzen ez duten indibiduoak moldatzea da; ikusi 3. algoritmoa. Baldin eta ibilbide baten luzera d_0 baino handiagoa bada, erpinak kentzeari ekingo diogu ibilbidearen luzera osoaren (1d) baldintza bete arte. Erpinak kentzeko ordena bat definituko dugu haien gainean. Bi ezaugarri izango ditugu kontuan ordenatzerakoan: batetik, erpina ibilbidean izatearen kostua, eta, bestetik, erpinaren saria. Erpina ibilbidean izatearen kostua intuizioz definituko dugu: ibilbidea erpin horretatik pasatuko ez balitz, erpin horren aurreko eta hurrengo erpinak ondoz ondokoak izanik, zenbatean laburtuko zen ibilbide osoaren luzera. Erpina ibilbidean izatearen kostua txikia eta erpinaren saria handia izatea bilatuko dugu eta bi irizpide horien pisua β parametroarekin doitu dugu. Hori kontuan izanda, honela defini dezakegu indizea $i_\beta : V \rightarrow \mathbb{R}^+$,

$$i_\beta(v_i) = \frac{d_{hi} + d_{ij} - d_{hj}}{s_i^\beta},$$

non v_h eta v_j herriak v_i herriaren aurrekoa eta hurrengoa diren hurrenez hurren. Ikusi 2. irudian i_β indizearen adibidea.

3. algoritmoa **Kendu eragilea**

while not ibilbidearen luzera murrizketa bete **do**
 Ordenatu erpinak eta azkena kendu ibilbidetik;
 Kostu lokala, kostu totala eta ibilbidearen asebetetzea eguneratu;
end while

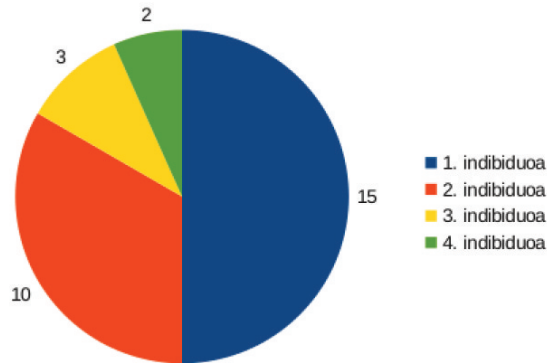


2. irudia. Kendu eragileko ordena indizea.

3.2.4. Gurasoen aukeraketa

Gurasoen aukeraketa egiterakoan bi helburu ditugu: hurrengo belau-naldiak hobetzea eta konbergentzia arinegiak eta maximo lokaletan jauzteko ekiditea. Maximo lokalak ekiditeko populazio osotik $ncand$ indibiduo aukeratu ditugu uniformeki gurasogaitzat. Gero, asebetetzearen arabera probabilitatean hautagai horietatik bi indibiduo aukeratu ditugu gurasotzat. Gurasoen aukeraketa honi erruletaren bidezko aukeraketa ere deitzen zaio eta algoritmo genetikoetan erabiltzen den teknika orokor bat da. Eragile honen pseudo-kodea 4. algoritmoan ikus daiteke. Hurrengo balioak kalkulatu behar dira: \bar{s}_i indibiduo bakoitzaren asebetetzea, hau da, $\bar{s}_i = \sum_{j \in S_i} s_j$, non $S_i \subset \mathcal{N}$, $i = 1, \dots, ncand$; m minimoa, $m = \min_{i=1 \dots ncand}(\bar{s}_i)$; m_0 konstantea, $m_0 = \min_{i=1 \dots n} (s_i)/ncand$; r_i pisuak, $r_i = \bar{s}_i - m + m_0$ eta p_i probabilitateak $p_i = r_i / \sum r_i$.

3. irudian lau gurasogaientzako erruleta bidezko aukeraketaren adibide bat dugu. Eman dezagun, $ncand = 4$, $\bar{s}_1 = 100$, $\bar{s}_2 = 101$, $\bar{s}_3 = 108$, $\bar{s}_4 = 113$ eta $m_0 = 2$. Gurasogai hautatua izateko probabilitateak $\frac{100}{422}$, $\frac{101}{422}$, $\frac{108}{422}$ eta $\frac{113}{422}$ hartu beharrean, guztiak $1/4$ ingurukoak baitira, p_i probabilitateak erabiliko ditugu. $m = \min(\bar{s}_i) = 100$ enez, $r_1 = 100 - 100 + 2 = 2$, $r_2 = 101 - 100 + 2 = 3$, $r_3 = 108 - 100 + 2 = 10$ eta $r_4 = 113 - 100 + 2 = 15$, irudian ikusten den bezala. Beraz, $p_1 = 2/30$, $p_2 = 3/30$, $p_3 = 10/30$ eta $p_4 = 15/30$.



3. irudia. Erruleta bidezko aukeraketa.

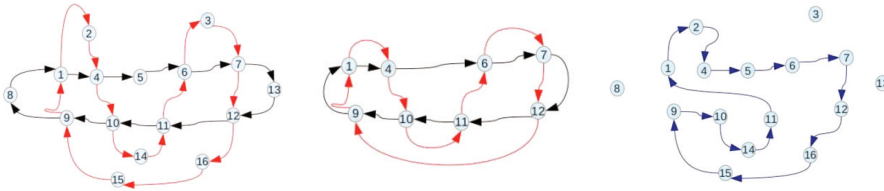
4. algoritmoa **Gurasoen aukeraketa**

Aukeratu populaziotik $ncand$ gurasogai;
 Gurasogaiei asebetetze txikienari deitu m , $m = \min(\bar{s}_i)$;
 Gurasogaiei esleitu hurrengo balioa, $r_i = \bar{s}_i - m + m_0$;
 Gurasogaiei esleitu guraso izateko probabilitatea, $p_i = r_i / \sum r_i$;
 Aukeratu bi guraso erruleta bi aldiz jaurtiz;

3.2.5. ER gurutzaketa-eragilea

Gurasoak diren bi indibiduotatik eratorritako haur bat sortzea da eragile honen helburua. Hau da, bi ibilbide gurutzatu eta hirugarren ibilbide bat sortzea; ikusi pseudo-kodea 5. algoritmoan. Eragile hau definitziteko [16] liburuan agertzen den TSPrentzat garatutako *Edge Recombination* gurutzaketa hartu dugu eredutzat. TSP simetrikorako gurutzaketa hau [17] argitalpenean aurkeztu zuten eta OPra moldatzeko aukera ematen du. Moldaketa egiterakoan jarraitu dugun irizpidea bi ibilbideetan dauden erpinak mantentzea izan da. Horregatik, gurutzaketa-eragile honetan, ez dira norabideak kontuan hartzen eta guraso batean bakarrik dauden erpinak probabilitistoki sartzen dira haurran.

OPra ER eragilea moldatzeko, lehenengoz bi ibilbideetako herri komunak hartu behar dira aintzat; ikusi 5. irudian ibilbide bikotea 4. irudiko gurasoetatik aterata.



4. irudia. Gurasoak.

5. irudia. Herri komunetako bi ibilbideak.

6. irudia. Haurra.

Bigarrenik, *ondoko herrien mapa* eraikiko dugu, herri bakoitzaren alboko herriak adieraziz. Hirugarrenik, ertz bakoitzean lotutako *tarteko herriak* gordeko ditugu. Ikusi 6. irudian.

5. algoritmoko ER gurutzaketa-eragilea aplikatu ondoren sortutako haurra.

5. algoritmoa **ER gurutzaketa-eragilea**

Aukeratu hasierako herria, gure kasuan 1, eta *momentu herria* izendatu;

while bisitatu gabeko herriak gelditzen dira **do**

Kendu *momentuko herria* herrien maparen eskuineko aldetik;

if Momentuko herriari *ondoko herrien mapan* herriak gelditzen bazaizkio **then**

Aurkitu momentuko herriaren ondoko herrietatik zeinek dituen herri gutxien bere ondoko herrien listan. Baldin eta berdintza bat badago aukeratu herri bat ausaz.

Deitu *hurrengo herria*;

Momentuko herriaren eta hurrengo herriaren tarteko herri multzoetatik aukeratu bat ausaz eta ibilbidera sartu;

Hurrengo herriari momentuko herria deitu;

else

if bisitatu gabeko herriak gelditzen dira **then**

Aukeratu ausaz bisitatu gabeko herri bat eta deitu *hurrengo herria*;

Momentuko herriaren eta hurrengo herriaren tarteko herri multzoetatik aukeratu bat ausaz eta ibilbidera sartu;

Hurrengo herriari momentuko herria deitu;

end if

end if

end while

3.2.6. Mutazioa

Soluzioen espazioa esploratzea da mutazioaren helburua, horregatik haurrek mutazioak jasan dezakete probabilitate batekin. Erabili dugun mutazio eragilean mutatzeko probabilitate berdina dute erpin guztiek; ikusi 6. algoritmoan mutazio-eragilearen pseudo-kodea.

6. algoritmoa **Mutazio-eragilea**

pmut probabilitatearekin mutazioa gertatuko den edo ez erabaki;

if Mutazioa gertatuko da **then**

{2, ..., *n*} multzoan aukeratu mutazioa jasango duen erpina;

if erpina ibilbidea badago **then**

kendu erpina ibilbidetik;

else

sartu erpina ibilbidera;

end if

end if

3.2.7. TSP eragilea

OP oso lotuta dago TSPrekin eta eragile honek funtzio hori betetzen du, aukeratutako herrietatik pasatzen den ibilbidea itzultzen du hain zuzen. Ibilbide hau distantzia minimoa egiten duena izan daiteke, baina ez du zertan; baliteke problema oso handietan heuristikak erabili behar izana. Horregatik, ibilbidearen kalkulua eragile moduan definitu dugu. Lan honetan nagusiki *Concorde* erabiliko dugu.

3.3. CER algoritmo genetikoaren eskema orokorra

Algoritmoa ideia simple batean oinarritzen da: hasieran ausazko indibiduoekin osatutako populazio bat eraikiko dugu eta belaunaldi berriak sortuko ditugu aurreko belaunaldien informazioa erabiliz asebetetzea hobetzen joateko. Indibiduen asebetetzeek, baliozkoak izan daitezten, ibilbide osoaren luzeraren (1d) baldintza errespetatu behar dute, baina murrizketa hori belaunaldi guztietan bete beharrean tarte bat utziko dugu eboluzioak bere lana egin dezan eta gehitu/kendu eragileen esku hartzea gutxitzeko. Izan ere, eragile hauek heuristikak direnez, maximo lokaletan erortzeko arriskua dago. CER algoritmoaren pseudo-kodea ikusi daiteke 7. algoritmoan.

7. algoritmoa CER algoritmoa

```
Ausazko populazio bat sortu (1. algoritmoa);
Kendu eragilea aplikatu populazioko indibiduo guztiei (3. algoritmoa);
Gehitu eragilea aplikatu populazioko indibiduo guztiei (2. algoritmoa);
tsp eragilea aplikatu populazioko indibiduo guztiei;
for i = 0 to n_0 do
  if mod(i, d2d) = 0 then
    Kendu eragilea aplikatu (3. algoritmoa);
    Gehitu eragilea aplikatu (2. algoritmoa);
    tsp eragilea aplikatu;
  else
    Populaziotik gurasoak aukeratu (4. algoritmoa);
    Guraso horren haurra sortu (5. algoritmoa);
    Haurra mutatu (6. algoritmoa);
    tsp eragilea aplikatu;
    Populazioa berritu haurra sartuz eta indibiduo txarrena kenduz;
  end if
end for
Kendu eragilea aplikatu populazioko indibiduo guztiei (3. algoritmoa);
Gehitu eragilea aplikatu populazioko indibiduo guztiei (2. algoritmoa);
tsp eragilea aplikatu populazioko indibiduo guztiei;
```

4. KONPUTAZIOZKO ESPERIENTZIA

4.1. Erabilitako *softwarea* eta *hardwarea*

OP ebazteko CER algoritmoaren portaera aztertuko dugu parametro eta datu multzo desberdinekin. Rn inplementatu dugu (ikusi [18]) prototipoak eta esperimentuak egiteko aproposa izateaz gain, azterketa estatistikorako tresnak, SQL datu baseekin konexioak eta C-n idatzitako programei deiak egiteko aukera eskaintzen duelako. Hala ere, programazio lengoia interpretatu bat den heinean, konpilatu behar direnak baino geldoagoa da eta literaturako C-n algoritmoen konputaziozko saiakerekin konparaketak egitea ez da egokia.

Guztira bi konputagailu erabili ditugu saiaketetarako: bata, UPV/EHU-ko SGI/IZO-SGIkeren ARINA konputagailu klusterra (ikusi [19]) 2.3 Ghzko 8 Xeon *core* eta 35 Gb RAM dituen Red Hat bat, eta bestea, UPV/EHUko Ekonomia Kuantitatiboko Laborategiak duen LEC konputagailua, Debian bat, 2 Xeon 5.355 prozesagailua (2×4 *core*koa), 2.664 Ghz eta 31 Gb RAM dituen. Metodo zehatzak eta metaheuristikoak konparatzeko LEC makinan egin ditugu exekuzioak eta ezaugarri desberdinak dituen konputagailu batean MTZ modelizazioaren portaera konparatzeko egin ditugu ARINAn ere exekuzioak.

R-ren 3.1.0 bertsioa erabili dugu eta oinarrizko liburutegiez gain, PostgreSQL datu basearekin konektatzeko RODB 1.3.10, Rtik *concorde* deitu eta TSP liburutegiaren 1.0-8. bertsioaren moldaketa bat; problema-*ren* erlaxazio lineala ebazteko, GLPK optimizazio *software*ari deiak egiteko *glpkAPI* 1.2.10 eta gehitu eta kendu eragileen bertsio paralelorako *snowfall* 0.3. Eskuragarri dagoen *Concorde*-ren azken bertsioa erabili dugu, hau da, 03.12.19 (ikusi [15]) eta CPLEX V12.5 optimizazio *softwarea* (ikusi [20]).

4.2. Datu multzoak

Algoritmoen portaera aztertzerakoan eragin handia du zein datu multzo aukeratzen dugun. Datu multzoak aukeratzeko orduan bi irizpide hartu ditugu kontuan. Lehenengoa, hurbiltasunagatik motibagarria izatea, adibidez, turismoan edo mendi-lasterketetan aplikagarriak. Datu multzo horri honako baldintzak betetzea eskatu diogu: (i) problema interpretatzeko modukoa izatea; (ii) tamaina handitzeko aukera ematea; (iii) erpinetako sariak zentsu bat izatea, ez ausazkoa. Hori kontuan izanda, erpinak Euskal Herriko herriak izatea eta erpinen sariak herri bakoitzaren izenaren letra kopurua izatea erabaki dugu. Interneten bidez EAeko 140 herriren longitudea eta latitudea agertzen den taula hartu dugu; ikusi [21]. *d* distantzia matrizea zehazteko, herrien arteko distantzia euclideanra kalkulatu dugu.

Bigarren irizpidean, OPren emaitzak erakusten dituen artikulu baten datu-basea erabili nahi izan dugu. Literaturan OPrekin lotuta, datu multzo batzuk aurki daitezke eskuragarri, adibidez, Werktuigkunde unibertsitatearen web orrialdean argitaratuta daudenak; ikusi [22].

Hiru datu multzoen artean, [4]ren 66 herriko problema erabaki dugu aukeratzea, herri kopuruaren egokitasunagatik. [10] artikuluan agertzen direnak baztertu ditugu, tamainaren aldetik txikiegiak baitira. [13]-ko datu multzoa baztertu egin dugu, herrien sariak ausaz kalkulatuta daudelako eta, beraz, ez errepikagarriak direlako.

4.3. MTZ modelizazioaren CPLEXen emaitzak

Saiakerak egin aurretik interesgarria izango da problemaren modelizazioaren araberako ezaugarriak ezagutzea. EAEko datu-basearekin 5 kasu begiratu ditugu. Datu-baseko lehenengo 10, 20, 40, 80 eta 140 herri hartuz OP10, OP20, OP40, OP80 eta OP140 problemak zehaztu ditugu, hurrenez hurren.

1. taulan (4a)-(4f) modelizazioaren dimentsioak agertzen dira. Zutabeek hurrengo ezaugarriak adierazten dituzte: n , herri kopurua; na , aldagai kopurua, $na = 2 \cdot n + n^2$; nb , aldagai bitarren kopurua, $nb = n + n^2$; nu , aldagai jarriekin kopurua, $nu = n$; m , murrizketa kopurua, $m = n^2 - n + 1$; nen , elementu ez nuluen kopurua; den , murrizketen matrize-dentsitatea ehunekotan adierazita, $den = 100 \cdot \frac{nen}{m \cdot na}$.

1. taula. EAEko OPko kasuen dimentsioak.

Kasua	n	Aldagaiak			Murrizketak		
		na	nb	nu	m	nen	den
OP10	10	120	110	10	93	506	4,53
OP20	20	440	420	20	383	2.206	1,31
OP40	40	1.680	1.640	40	1.563	9.206	0,35
OP80	80	6.560	6.480	80	6.323	37.606	0,09
OP140	140	19.880	19.740	140	19.463	116.206	0,03

2. taulan LEC eta ARINA makinetan EAEko problemen (4a)-(4f) modelizazioaren CPLEXen portaera adierazi dugu. Zutabeen esanahiak ondokoak dira: OPT , CPLEXekin lortutako soluzioa, OG , *optimality gap*, CPLEXekin lortutako soluzio faktiblearen eta lortutako azken kotaren arteko distantzia erlatiboa ehunekotan, $OG = 100 \cdot \frac{OPT - OPT}{OPT}$ eta T , exekuzio denbora segundutan. Ikusten dugunez, ordu betean (4a)-(4f)rekin modeli-

zatutako 10 eta 20 herriko problemak ebazteko gai izan gara CPLEXekin eta beste problemetan, ordea, OG gero eta handiagoekin bukatzen du. Hor-taz, algoritmo aproposagoak behar ditugu, baina zaila izan daiteke metodo zehatzetan oinarritutako algoritmo apropos bat aurkitzea. Hori ikusita, al-goritmo genetiko bat garatu dugu eta haren portaera eta egokitasuna aztertu ditugu.

2. taula. CPLEXeko esperientzia n ren arabera.

Kasua	α	n	LEC			ARINA		
			<i>OPT</i>	<i>OG</i>	<i>T</i>	<i>OPT</i>	<i>OG</i>	<i>T</i>
OP10	0,5	10	53	*	0,14	53	*	0,48
OP20	0,5	20	101	*	41,57	101	*	43,13
OP40	0,5	40	193	6,22	—	193	5,7	—
OP80	0,5	80	429	14,28	—	429	13,67	—
OP140	0,5	140	655	33,75	—	671	30,26	—

*: optimoa.

—: denboraren muga gaindituta (3.600 seg).

4.4. n eta α ren araberako CER algoritmoa vs CPLEX

Atal honetan CER algoritmoaren CPLEXekiko portaera konparatu nahi dugu. Hemendik aurrera besterik ez badugu esaten parametroei $\alpha = 0.5$, $d2d = 100$, $ncandit = 10$, $\beta = 0.2$, $pmut = 0.05$, $npop = 100$ eta $n_0 = 2000$ balioak emango dizkiegu.

Bi egoera diferente aztertuko ditugu: lehenengo, herri kopuruaren era-gina ikusiko dugu; horretarako aipatutako 5 problemak erabiliko dira. Bi-garren, OP140 problema handian oinarrituta, 5 kasu sortu ditugu (4e) mu-rrizketan *rhs* aldatuz. Konparaketak egiteko problemaren kasu bakoitzeko bost saiakera egin ditugu.

Hurrengo taulen zutabe berrien esanahia honakoa da: *ABF absolute best fitness*, bost exekuzioren ostean lortutako asebetetze onena; *MBF (DEF) mean best fitness*, exekuzio bakoitzeko asebetetze onenen batzbestekoa (desbiderapen estandarra); *GG goodness gap*, CER algoritmoaren emaitza-ren kalitatea neurtzeko erabiliko dugu, CPLEXekin lortutako emaitzarekiko distantzia erlatiboa ehunekotan adieraziz, hau da, $GG = 100 \cdot \frac{ABF - OPT}{OPT}$.

3. taulan herri kopuruaren araberako konparaketak egin ditugu. Horre-tarako, CPLEXen exekuzio denboraren limitea 5 orduan jarri dugu. $n = 40$ deneko kasuan maximo berdina lortu dugu bi metodoentzat baina CER al-goritmoak 3 minutu t'erdi behar izan dituen bitartean CPLEXek 5 ordu

behar izan ditu; gainera, CER algoritmoaren exekuzio guztietan maximo berdinerira iritsi gara. $n = 80$ deneko kasuan maximo oso antzekoa lortu dute bi metodoek, baina MBF begiraturaz algoritmo genetiko honek 8 minutuan metodo zehatzak 5 orduan lortuko duen maximoa baino handiagoa lortuko duela espero dezakegu. $n = 140$ den kasuan aldea are nabarmenagoa da. Algoritmo genetikoaren bukatzeko behar den denbora herri kopuruarekin hasten dela ikusten da.

3. taula. n ren arabera CER eta CPLEXen emaitzak.

Kasua	α	n	CER			CPLEX			GG
			ABF	MBF(DEF)	T(DE)	OPT	OG	T	
OP40	0,5	40	198	198(0)	206,13(1,33)	198	0,78	—	0
OP80	0,5	80	435	433,4(1,14)	490,11(12,36)	432	11,51	—	0,69
OP140	0,5	140	751	744(8,51)	947,11(35,37)	671 ¹	30,07	12960	11,62

—: denboraren muga gaindituta (18.000 seg).

¹: memoriaren muga gaindituta (31 GB).

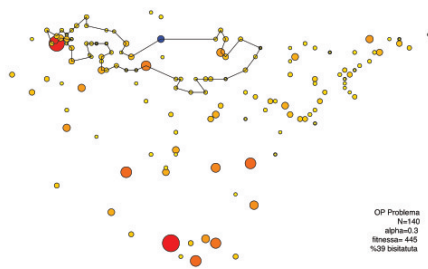
4. taulan α desberdinekin bi metodoek nolako portaera duten ikusi dezakegu. α parametroa 0 edo 1-era hurbildu ahala GG txikitzen dela ikusten dugu. Bestetik, α handitu ahala algoritmo genetikoaren exekuzio-denbora handitzen dela ikusten dugu. Ikus dezakegunez, GG beti positiboa da, hots, CER algoritmoaren emaitzak beti hobetoak dira.

4. taula. α ren arabera CER eta CPLEXen emaitzak.

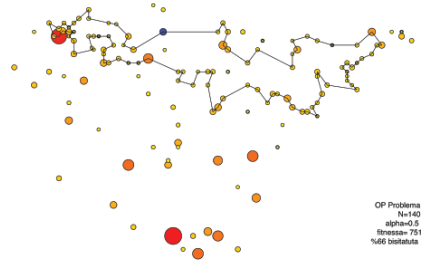
Kasua	α	n	CER			CPLEX			GG
			ABF	MBF(DEF)	T(DE)	OPT	OG	T	
OP140	0,3	140	445	443,2(4,02)	463,86(17,51)	409	44,78	—	8,80
OP140	0,5	140	751	744(8,51)	947,11(35,37)	655	33,75	—	14,66
OP140	0,7	140	951	943,2(6,723)	1212,2(95,53)	797	31,24	—	19,32
OP140	0,9	140	1.155	1152,34(2,73)	2015,27(55,89)	983	19,65	—	17,50

—: denboraren muga gaindituta (3.600 seg).

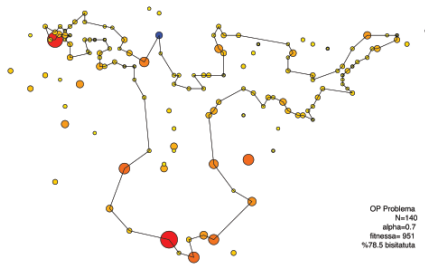
7-11 irudietan, α parametroaren balio desberdinetarako CER algoritmoarekin lortutako ibilbideak ikusi ditzakegu. Ohar gaituzen erpinen tamaina eta kolorea herri bakoitzaren sariari lotuta dagoela, eta 1. herria, urdinez dagoena, Gernika-Lumo dela.



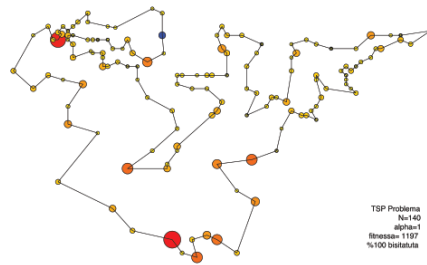
7. irudia. OPren emaitza ($\alpha = 0, 3$)



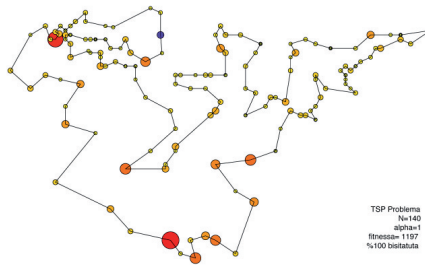
8. irudia. OPren emaitza ($\alpha = 0, 5$).



9. irudia. OPren emaitza ($\alpha = 0, 7$).



10. irudia. OPren emaitza ($\alpha = 0, 9$).



11. irudia. TSPren emaitza ($\alpha = 1$).

Irudi hauek ikusiz, OPeK α parametroarekiko sensibilitate handia duela ikusten da. Batetik, herri bat aukeratu dezakegu eta α -ren arabera ibilbidean dagoen edo ez ikusi. Adibidez, Ondarroa hartzen badugu, $\alpha = 0,3$ denean ibilbidean dagoela ikusten da, $\alpha = 0,5$ denean ez dago ibilbidean eta $\alpha = 0,7$ denean berriz dago ibilbidean. Bestetik, irudi guztietan dauden hiru herri aukera ditzakegu eta haien arteko posizioa ikusi. Adibidez, Bilbo, Amorebieta-Etxano eta Gernika-Lumo aukeratzen baditugu, ordena aldatzen da α -ren arabera.

4.5. CER algoritmoaren analisi parametrikoa

Atal honetan CER algoritmoaren $npop$, $d2d$, β , $ncand$ eta $pmut$ parametrokiko konbergentzia-analisiaren emaitza batzuk erakutsiko ditugu. Parametro bakoitzaren 5 kasu aztertu ditugu eta kasu bakoitzaren bost exekuzio.

4.5.1. Populazioaren tamaina

$npop$ parametroak populazioaren tamaina adierazten du. 5. taulan $npop$ 50, 75, 100, 125 eta 150 deneko kasuen exekuzioak ikusi daitezke, $npop = 100$ oinarritzko balioa izanik.

5. taula. Populazioaren tamainaren arabera CERen emaitzak.

Kasua	$npop$	n	α	ABF	MBF(DEF)	T(DE)
OP140	50	140	0,5	750	732,4(15,2)	871,79(78,06)
OP140	75	140	0,5	749	739,4(10,26)	924,87(68,69)
OP140	100	140	0,5	751	744(8,51)	947,11(35,37)
OP140	125	140	0,5	755	746(6,44)	948,83(35,6)
OP140	150	140	0,5	748	739,2(5,8)	952,13(35,84)

Populazio txikiekin konbergentzia arinagoak lortzen dira, populazioaren batez besteko asebetetzea arinago hasten da, baina maximo lokaletan jausteko arriskua dago. MBF zutabea begiratzuz, ematen du hori gertatu dela 50 eta 75 indibiduoko kasuekin. Populazio handiekin konbergentzia geldoagoa da, populazioaren batez besteko asebetzea astiroago hasten da, baina ABF handiagoak lortzeko aukera handitzen da. Populazioaren tamaina egokiena 125 indibidukoa dela ematen du. Izan ere, ABF eta MBF handienak eta DEF txikienak lortzen dira. Populazioaren tamaina hasi ahala, beste parametroei berdinean eutsiz, algoritmoak behar duen denbora handitzen doala ondoriozta daiteke.

4.5.2. β parametroa

β parametroak sarien pisua adierazten du kendu eragilean erpinak ordenatzerakoan. 6. taulan 0, 0,1, 0,2, 0,5 eta 1 deneko kasuen exekuzioak agerzen dira, oinarritzko $\beta = 0, 2$ izanik.

MBF zutabea begiratzuz ematen du ez dela gomendagarria gehiegizko pisua izatea. ABF eta MBF zutabeei begiratzuz ikusten da β parametroa erabiltzea lagungarria dela, eta bereziki saiakera hauek ikusi ostean $\beta = 0$ edo

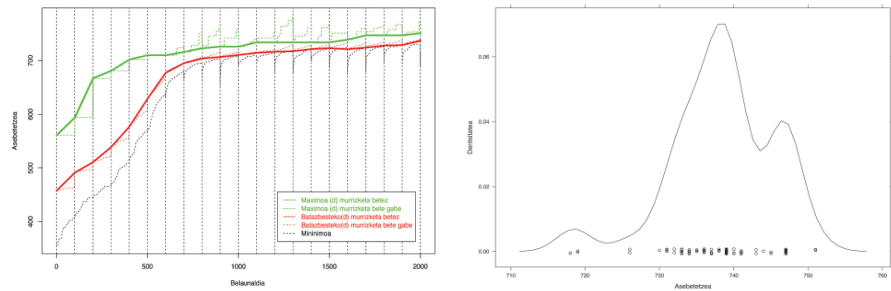
6. taula. β parametroaren araberako CERen emaitzak.

Kasua	β	n	α	ABF	MBF(DEF)	T(DE)
OP140	0	140	0,5	751	745(5,8)	941,12(40,52)
OP140	0,1	140	0,5	753	744,55(9,67)	934,68(41,88)
OP140	0,2	140	0,5	751	744(8,51)	947,11(35,37)
OP140	0,5	140	0,5	754	743,2(7,41)	920,30(37,17)
OP140	1	140	0,5	752	739,55(14,15)	873,67(60,96)

$\beta = 0.5$ aukeratzea izan daiteke onena, MBF eta DEF onenak lortzen direlako. Hala ere, $\beta = 0.5$ denean azterketa parametrikoko osoko ABF handiena lortu da. $\beta = 1$ denean beste kasuetan baino denbora gutxiago behar du. Izan ere, β berretzaile bat da eta operazio gutxiago egin behar ditu.

4.5.3. Konbergentzia eta asebetetze-banaketa

Ezkerraldeko 12. irudian exekuzio baten asebetetzearen konbergentzia ikusi dezakegu; berdez, asebetetze maximoaren balioa, gorritz, batzazbeste-



12. irudia. CER algoritmoaren eboluzioa eta asebetetze-dentsitatea (OP140, $\alpha = 0.5$).

koa eta, beltzez, balio minimoa. Horrez gain, ondoz ondoko $d2d$ desberdinetako balioak lerro jarraiaz batu ditugu, hauek baitira OPren soluzio egin-garriak, eta lerro etenez jarri ditugu belaunaldiz belaunaldi asebetetzearen balioak.

Eskuinaldeko 12. irudian, algoritmoaren amaieran, populazioko indibiduen asebetetzearen banaketa eta dentsitate-funtzioa ikusi daitezke. Irudi honetan asebetetzearen hiru akumulazio antzematen dira: asebetetze txiki-dun zenbait indibiduok osatzen dutenak, asebetetze handien beste talde batek osatutakoa eta azkenik gehiengo batek osatutakoa.

4.6. CERen portaera Chaoren datu-basearekiko

CER algoritmoa beste datu multzo batzuetan nola portatzen den ikusteko [4] artikuluko 5 problema hartu ditugu. 7. taulan 5.l, 5.n, 5.p, 5.r, 5.t, 5.v eta 5.x kasuen emaitzak agertzen dira. Exekuzioetarako erabili ditugun parametroak $d2d = 150$, $ncand = 10$, $\beta = 0,5$, $pmut = 0,05$, $npop = 100$ eta $n_0 = 4000$ dira.

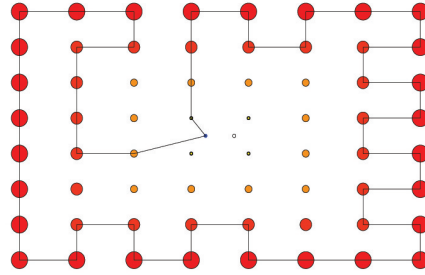
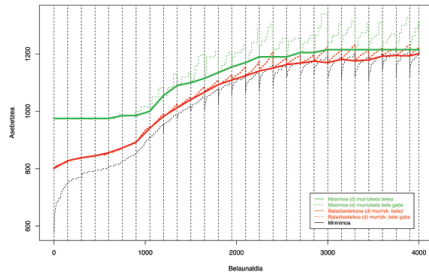
Esan daiteke, erabili ditugun parametroekin, artikuluko horretan proposatutako heuristika aproposagoa dela datu multzo honetan α txikietarako. Eta α handiagoekin, berriz, CER algoritmoak emaitza hobekiago lortzen dituela, exekuzio denbora handiagoekin eta emaitzak txarragoak direla soilik bi kasutan eta diferentzia erlatiboa oso txikia dela.

7. taula. CER eta Chaoren emaitzak datu base berriarekiko.

Kasua	n	α	CER			CHAO		GG
			ABF	MBF(DEF)	T(DE)	OPT	T	
5.l	66	0,47	890	830(60,1)	206,82(37,07)	915	84,95	-0.03
5.n	66	0,54	1070	1036(22,7)	267,75(21,06)	1070	119,00	0
5.p	66	0,62	1215	1182(15,24)	360,95(16,55)	1215	108,93	0
5.r	66	0,7	1330	1310(21,51)	524,75(32,31)	1340	502,41	-0.02
5.t	66	0,78	1465	1465(0)	672,61(6,92)	1435	128,56	0.02
5.v	66	0,85	1560	1560(0)	883,1(10,9)	1550	469,94	0.01
5.x	66	0,93	1635	1635(0)	1078,05(11,3)	1635	357,98	0

13. irudian ikusteko dugu hasierako ausazko populaziotik lortutako asebetetze onena gainditzeko ia 1.000 belaunaldi behar dituela. Azterketa parametrikotik egin beharko genuke algoritmoa problemara doitzeko. Baita horren arrazoiak problema honetarako populazio handiegia hartu dugula izatea; gainera, tamaina hori txikituz konbergentzia azkarragoak izatea gertatu daiteke; Chao-ren exekuzio denborekin duen aldea gutxiagotuko luke horrela.

Azkenik, 14. taulan 5.t kasuaren OPren soluzio bat aurkezten dugu. Aurrekoan bezala, erpinen sariak tamaina eta kolorearen bidez adierazita daude. Ikusten dugunez, problema hauen erpinen posizioa eta saria oso berezia da. Izan ere, duten karratu itxuragatik *squared shaped* izena du Chao-ren problema multzo honek.



13. irudia. CER algoritmoaren eboluzioa (5.p).

14. irudia. OPren emaitza (5.t).

5. ONDORIOAK

Lan honetan zehar Orientazio Problema aurkeztu dugu, TSPren aldaera bat dena. OPren aplikazio batzuk, konputaziozko konplexutasuna eta modelizazioak ere landu ditugu. Problema honen ebazpenerako baliogarria izan daitekeen CER algoritmo genetikoa diseinatu da, berrikuntza azpimarragarrienak direlarik *Concorde* eta ER eragile moldatuaren erabilera. 500 lerro inguruko kodea Rn inplementatu da eta konputaziozko esperientzia burutzeko 60 ordu baino gehiago erabili dira. Horretarako, EAEko datu multzo batean eta literatura matematikoan ezaguna den beste batean egin ditugu saiakerak. Zenbait kasutan prozedurak ondo funtzionatzen duen arren, algoritmo genetiko guztietan bezala kasu bakoitzeko azterketa parametrikoa egitea eskatzen du. Gainera, garatutako kodeak aukera ematen du ibilbideak kalkulatzeko *Concorde* zehatza erabili beharrean *Lin-Kernighan* heuristika erabiltzeko eta, horrela, herrien kopuru handiagorekin lan egiteko.

Gaur egungo aplikazio batzuetan, emaitza arinak behar direnez analisi parametrikoa egiteko denborarik ez dago. Esate baterako, internet bidez egindako *turismo bidaia pertsonalizatuek* segundu gutxian soluzio «onargarriak» ematen dituzten algoritmoak eskatzen dituzte. Horrez gain, denbora-leihoak eta egun bat baino gehiago (ibilgailu bat baino gehiago) edukitzen dituzte problema hauek. OPren orokorpena den problema honi *Time Dependent Team Orienteering Problem with Time Windows* (TDTOPTW) deitu diote batzuek; ikusi [7]. Hau bezalako problemak etorkizunerako ikerketa ildo interesgarriak direla uste dugu. Horretarako, VNS eta GRASP metaheuristikak (ikusi [23]) zein *Branch-and-Cut* eta *Branch-and-Price* metodologiak aztertzeke interesa dugu; ikusi [13].

6. ESKER ONAK

Eskerrak eman nahi dizkiegu Euskal Herriko Unibertsitateko BETS 2011 Prestakuntza eta Ikerketa Unitateari, Eusko Jaurlaritzako IT-567-13 Ikerketa Taldeari, Ekonomia eta Lehiakortasun Espainiako Ministerioko MTM2012-31514 proiektuari eta Zientzia eta Teknologia Garatzeko Iberoamerikako Programaren P711RT0278 proiektuari. Eskerrak eman nahi dizkiogu orobat konputaziozko esperientzia burutzeko LEC makinara sarbidea emateagatik UPV/EHUko Ekonomia Kuantitatiboa Saileko Fernando Tusell katedradunari. Aldi berean, eskertzekoa da IZO-SGI, SGIKer (UPV/EHU, MICINN, GVIEJ, ER DF eta ESF)ren ARINA konputagailu clusterra erabiltzeko aukera eduki izana. Era berean egileok begirale tekniko eta hizkuntza-aholkulariei eskertzen diegu egindako lana.

7. BIBLIOGRAFIA

- [1] VANSTEENWEGEN, P.; SOUFFRIAUX, W. eta OUDHEUSDEN, D.V. 2011. «The orienteering problem: A survey», *European Journal of Operational Research*, 209, 1-10.
- [2] GOLDEN, B.L.; LEVY L. eta VOHRA, R. 1987. «The orienteering problem», *Naval*, 34, 307-318.
- [3] EDMONDS, J. 1965. «Paths, Trees, and Flowers», 449-467, 17, 449-467.
- [4] CHAO, I.; GOLDEN, B.L. egta WASIL E.A. 1996. «A fast and effective heuristic for the orienteering problem», *European Journal of Operational Research*, 88, 475-489.
- [5] BALAS, E. 1989. «The prize collecting traveling salesman problem», *Networks*, 19, 621-636.
- [6] GARCIA, A.; VANSTEENWEGEN, P.; ARBELAITZ, O.; SOUFFRIAUX, W. eta LINAZA M.T. 2013. «Integrating public transportation in personalised electronic tourist guides», *Computers & OR*, 40, 758-774.
- [7] GARCIA, A.; ARBELAITZ, O.; LINAZA, M.T.; VANSTEENWEGEN, P. eta SOUFFRIAUX, W. 2010. «Personalized tourist route generation», in *ICWE Workshops* (F. Daniel and F.M. Facca, eds.), 6385 of Lecture Notes in Computer Science, 486-497, Springer.
- [8] SOUFFRIAUX, W.; VANSTEENWEGEN, P.; VERTOMMEN, J.; BERGHE, G.V. eta OUDHEUSDEN D.V. 2008. «A personalized tourist trip design algorithm for mobile tourist guides», *Applied Artificial Intelligence archive*, 22, 964-985.
- [9] VANSTEENWEGEN, P. 2009. «Planning in tourism and public transportation», *4OR*, 7, 293-296.
- [10] TSILIGIRIDES, T. 1984. «Heuristic methods applied to orienteering», *Journal of the Operational Research Society*, 35, 797-809.

- [11] YU, J.; SCHWAGER, M. etga RUS, D. 2014. «Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks», *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago.
- [12] DANTZIG, G.B.; FULKERSON, D.R. eta JOHNSON, S.M. 1954. «Solution of a large-scale traveling-salesman problem», *Operations Research*, 2, 393-410.
- [13] FISCHETTI, M.; SALAZAR GONZALEZ, J.J. eta TOTH, P. 1998. «Solving the orienteering problem through branch-and-cut», *INFORMS Journal on Computing*, 10, 133-148.
- [14] MILLER, C.E.; TUCKER, A.W. eta ZEMLIN, R.A. 1960. «Integer programming formulation of traveling salesman problems», *J. ACM*, 7, 326-329.
- [15] Concorde, «Concorde tsp solver.» <http://www.math.uwaterloo.ca/tsp/concorde/index.html>.
- [16] LARRAÑAGA, P.; KUIJPERS, C.; MURGA, R.; INZA, I. eta DIZDAREVIC, S. 1999. «Genetic algorithms for the travelling salesman problem: A review of representations and operators», *Artificial Intelligence Review*, 13, 129-170.
- [17] WHITLEY, D.L.; STARKWEATHER, T. eta FUQUAY, D. 1989. «Scheduling problems and travelling salesman: The genetic edge recombination operator», in *Proc. of the Third Int. Conf. on Genetic Algorithms* (J.D. Schaffer, ed.), (San Mateo, CA), 133-140, Morgan Kaufmann.
- [18] R, «R project for statistical computing». <http://www.r-project.org/>.
- [19] ARINA, «Cluster IZO-SGI, SGIker (UPV/EHU).» <http://www.ehu.es/sgi/recursos/cluster-arina-2>.
- [20] ILOG CPLEX OPTIMIZER. <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>.
- [21] TimeGenie: Latitud y Longitud País Vasco, http://spa.timegenie.com/latitude_longitude/state/espv.
- [22] WERKTUIGKUNDE, D. «The orienteering problem: Test instances.» <http://www.mech.kuleuven.be/en/cib/op>.
- [23] BLUM, C. eta ROLI, A. 2003. «Metaheuristics in combinatorial optimization: Overview and conceptual comparison», *ACM Computing Surveys*, 35, 268-308.