

Zorizko instantzia uniformeak sortzen al dira optimizazio konbinatorioan?¹

(Do we generate instances uniformly at random in combinatorial optimization?)

Josu Ceberio¹, Borja Calvo¹, Alexander Mendiburu², Jose Antonio Lozano¹

Intelligent Systems Group (ISG)

¹ Konputazio Zientziak eta Adimen Artifiziala Saila

² Konputagailuen Arkitektura eta Teknologia Saila, Informatika Fakultatea (UPV/EHU)

josu.ceberio@ehu.eus

DOI: 10.1387/ekaia.18877

Jasoa: 2017/12/15

Onartua: 2018/03/05

Laburpena: Konputazio ebolutiboan, algoritmoek optimizazio-problemen gainean duten errendimendua ebaluatzeko, ohikoa izaten da problema horien hainbat instantzia erabiltzea. Batzuetan, problema errealean instantziak eskuragarri daude, eta beraz, esperimentaziorako instantzien multzoa hortik osatzen da. Tamalez, orokorrean, ez da hori gertatzen: instantziak eskuratzeko zailtasunak direla tarteko, ikerlariak instantzia artifizialak sortu behar izaten dituzte. Lan honetan, instantzia artifizialak uniformeki zoriz sortzearen inguruko aspektu batzuk izango ditugu aztergai. Zehazki, bibliografian horrenbestetan onetsi den ideia bati erreparatuko diogu: Instantzien parametroen espazioan zein helburu-funtzioen espazioan uniformeki zoriz lagintzea baliokideak dira. Exekutatu ditugun esperimentuen arabera, baliokidetasuna kasu batzuetan ez dela betetzen frogatuko dugu, eta beraz, sortzen diren instantziek espero diren ezaugarriak ez dituztela erakutsiko dugu.

Hitz gakoak: Optimizazio konbinatorioa, problema, ranking, instantzia, parametroa.

Abstract: In evolutionary computation, it is common practice to use sets of instances as test-beds for evaluating and comparing the performance of new optimisation algorithms. In some cases, real-world instances are available, and, thus, they are used to constitute the experimental benchmark. Unfortunately, this is not the general case. Due to the difficulties for obtaining real-world instances, or because the optimisation prob-

¹ Artikulu hau [1] lanean dago oinarrituta, eta aurrekoa publikatu ondoren garatutako ikerketa jasotzen du.

lems defined in the literature are not exactly as those defined in the industry, practitioners are forced to create artificial instances. In this paper, we study some aspects related to the random generation of artificial instances. Particularly, we elaborate on the assumption that states that sampling uniformly at random in the space of parameters is equivalent to sampling uniformly at random in the space of functions. Illustrated with some experiments, we prove that for some type of algorithms this assumption does not hold.

Keywords: Combinatorial optimization, problem, ranking, instance, parameter.

1. SARRERA

Optimizazio heuristikoaren arloan, komunitateak ehunka teknika proposatu izan ditu planteatu izan diren optimizazio konbinatorioko problemak ebazteko. Batzuetan, problema horiek errealak dira, eta euren jatorria anitza izan ohi da. Industria [2], logistika [3], hezkuntza [4] edo osasuna [5] dira aniztasun horren adibide. Aldiz, gehienetan, komunitatearen beharrak direla eta, optimizazio-teknika aurreratuak garatzeko batak bat, jatorrizko problema horiek moldatu egiten dira. Batzuetan zailtasun maila handitzeko, eta beste batzuetan aldiz, errealitateak inposatzen dituen murriztapen partikularrak sinplifikatzeko.

Zoritxarez, hainbat arrazoi direla medio, problema errealak eskuratzeko zaila izaten da. Kasu batzuetan, ezagutzen diren adibideak zaharrak dira (1980ko hamarkadakoak), eta gaur egungo konputazio-baliabideak kontuan hartuta, optimizazioa berehalakoa da. Beste kasu batzuetan, konfidentzialtasun-arrazoiak direla eta, ia ezinezkoa da instantzia errealak lortzea.

Argudioak argudio, sarritan, proposatutako algoritmoen errendimendua ebaluatzeko beharrezkoak diren problemen instantziak urriak dira. Arazo horri aurre egiteko eta azterketa esperimentalak gauzatzeko, ikertzaileek instantzia artifizialak sortu ohi dituzte, ahal den neurrian, problema errealak dituzten ezaugarriak kontuan izanik. Ezinezkoa denean aldiz, instantziak uniformeki zoriz sortu ohi dira.

Nola edo hala, instantzia artifizialak erabiltzea algoritmoen konparazio esperimentalak egiteko erabat onartuta dagoen prozedura bat da. Izan ere, bibliografian ehunka adibide aurkitu ditzakegu. Batzuk aipatzearen, eta permutazio-problemetan [6] zentratuta, Gent et al. [7]-ek saltzaile ibiltariaren problemaren (*Travelling Salesman Problem*) ausazko instantzien transizio-fasea izan zuten aztergai. Instantziak sortzeko, n hiri uniformeki ausaz kokatzen zituzten espazio kartesiar batean, eta distantziak bertatik eratorri.

[8] artikuluan, instantzia artifizial multzo bat proposatu zen *permutation flowshop scheduling problem (PFSP)* problemarako. Autorearen ara-

bera: ahalik eta instantzia zailenak sortzeko asmotan, ausazko instantzia-multzo handi bat sortu zuen ([9] artikuluan proposatutako zorizko zenbaki sortzailea erabiliz). Jarraian, tabu-bilaketarako algoritmo bat aplikatu zuen hainbat aldiz instantzia bakoitzaren gainean. Lortutako emaitzak aztertu ostean, zailenak ziruditen 120 instantzia aukeratu zituen, zailenak ziruditenak.

[10] artikuluan, autoreek 188 instantzia bildu zituzten, historian zehar esleipen-problema koadratikoaren (*Quadratic Assignment Problem (QAP)*) inguruan eginiko ikerketetan proposatuak, eta eskuragarri jarri zituzten guztiak *QAP Library* izeneko on-line biltegi batean. Haietatik, bakarrik 42 ziren jatorri errealeko instantziak; gainerako guztiak prozedura artifizial bati jarraituz sortu ziren. [11] artikuluan ordea, hainbat algoritmo metaheuristiko konparatu ostean, *QAP Library* biltegiko instantziak errazegiak direla ondorioztatu zen. Aukeratutako algoritmoak, guztiak ziren optimoetara heltzeko gai. Hori dela eta, 100 instantzia zailez osaturiko multzo berri bat proposatu zuten.

Ordenazio linealaren problemari (*Linear Ordering Problem (LOP)*) dagokionez, artikuluetan gehien erreferentziatzen den liburutegia LOLIB da [12]. Liburutegi horretan, ekonomiaren arlotik eratorritako 49 instantzia erreal biltzen dira. [13] artikuluan ordea, autoreek, berriro ere, instantzien zailtasun-maila apala nabarmenduta, instantzia-multzo berri bat sortu zuten, instantzia errealean ezaugarriak mantenduz, baina zailtasuna areagotuz; instantzia berriok, LOLIB-eko instantzien parametroak uniformeki ausaz laginduz lortu zituzten. Aurrekoez gain, [14, 15] artikuluan, beste instantzia-multzo bat proposatu zen. Kasu horretan, autoreek [0,100] tartean uniformeki ausaz laginduz lortu zituzten parametroak.

Aipatu ditugun lanetan ikusi daitekeen bezala, instantzia artifizialak sortzeko, ohikoa da horiek osatzen dituzten parametroak uniformeki ausaz lagintzea. Helburua da instantzien espazioaren eredugarria den instantzialagin uniforme bat lortzea da. Baina prozedura hori zuzena izateko, parametroen espazioan uniformeki ausaz lagintzea eta instantzien/helburu-funtzioen² espazioan uniformeki ausaz lagintzea baliokideak izan behar dira.

Lan honetan, baliokidetasun hori izango dugu eztabaidagai. Egindako ikerketaren arabera, algoritmoak optimizazio-prozesuan zehar soluzioei dagozkien helburu-balioak ezplizituki erabiltzen baldin baditu, orduan helburu-funtzioen espazioa infinitua da, eta beraz, baliokidetasuna betetzen da. Aldiz, optimizazio-algoritmoak soluzioen helburu-funtzioen balioak konparatu besterik ez badu egiten (hobeagoa, txarragoa...), orduan algoritmoak bereiz dezakeen instantzia/funtzio kopurua finitua da. Egindako esperi-

² Irakurketa errazteko, hemendik aurrera 'funtzioa' terminoa erabliko dugu 'helburu-funtzio' terminoari erreferentzia egiteko.

mentuen arabera, kasuan horretan baliokidetasuna ez dela betetzen ikusiko dugu.

Gainerako atalak ondorengo erara antolatzen dira: hurrengo atalean, lanaren muina ulertzeko behar den oinarri teorikoa azaltzen da; 3. atalean, instantzien zorizko laginketaren inguruko eztabaida eta ondorioak aurkezten dira; azkenik, 4. atalean, ondorioak eta etorkizuneko ikerketa-lerro batzuk proposatzen dira.

2. PROBLEMAK, INSTANTZIAK ETA PARAMETROAK

Atal honetan, eztabaida aurrera eramateko beharrezkoak diren terminoak definitzen eta ilustratzen dira.

Optimizazio kombinatorioko problema bat $\mathbf{P} = (\Omega, f)$ tuplak definitzen du, non Ω bilaketa-espazio finitu bat den eta $f: \Omega \rightarrow \mathbb{R}$ helburu-funtzioa den. Lan honetan, permutazio-problemetara mugatuko garen arren, proposatzen dugun azterketa edozein kombinatorio-problematara da aplikagarria. Kasu zehatz honetan, Ω izango da n tamainako permutazio guztien taldea izango da (n problemaren tamaina izanik). Problema f maximizatuko (edo minimizatuko) duen $\sigma^* \in \Omega$ bilatzean datza.

Aurreko elementuez gain, problema baten kasu partikular bat definitzeko sarrera batzuk behar ditugu; sarrera horiei, *parametro* deituko diegu, eta kasu partikular hori definitzen duen parametro-multzoari *instantzia* deituko diogu. Hiru termino horiek —*problema*, *parametro* eta *instantzia*— ilustratzeko, ordenazio linealaren problema edo LOP erabiliko dugu (permutazio-problema bat).

Linear Ordering Problem (LOP). *Izan bedi $B = [b_{ij}]_{n \times n}$ matrizea, non $b_{ij} \in \mathbb{R}$. LOPa-ren helburua B ko errenkadak eta zutabeak berrordenatzea da, diagonal nagusiaren gainean geratzen diren elementuen batura maximizatzeke (edo, diagonal azpiko sarreraren batura minimizatzeke); berrordenazio hori permutazio baten bidez adieraziko dugu. Jarraian dagoen ekuazioak LOParen helburu-funtzioa formalizatzen du:*

$$f(\sigma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n b_{\sigma_i, \sigma_j} \quad (1)$$

non σ_i baita σ -ren i -garren posizioari esleitutako lerroaren (eta zutabearen) indizea den.

LOPa, lehenengo aldiz, 1958an eman zen ezagutzera. Garai hartan, ekonomiako zenbait sektoreren arteko erlazioak modelatzeko erabili zen [16]. Gaur egun aldiz, aplikazioak ugariak dira eta oso arlo desberdine-

Zorizko instantzia uniformeak sortzen al dira optimizazio konbinatorioan?

tan aplikagarria da, esaterako, arkeologian, grafoen teorian edo psikologia matematikoan [17].

LOParen matrizeko *bij* elementuei *parametro* deritzegu, eta *B* matrizea osatzen duten parametro guztien balio-esleipen bati, *instantzia*; $n(n - 1)$ parametroz dago osatuta (helburu-funtzioari erreparaturaz, argi dago diagonal nagusiko parametroak ez direla kontuan izaten). Ikus 1. irudia.

3. AZTERKETA

Sarreran aipatu dugun bezala, parametroak uniformeki zoriz lagintzearen motibazioa instantzia-lagin uniformeak lortzea da, problemaren zailtasunaren eredugarri dena, alegia. Baina, sortzen den proba-kasuen lagina uniformea al da ohiko prozedurari jarraituz? Hori horrela izateko, instantziak parametroen espazioan uniformeki ausaz lagintzea eta instantziak funtzioen espazioan uniformeki ausaz lagintzea baliokideak izan behar dira.

	1	2	3	4	5
1	0	16	11	15	7
2	21	0	14	15	9
3	26	23	0	26	12
4	22	22	11	0	13
5	30	28	25	24	0

1. irudia. $n = 5$ tamainako LOParen instantzia baten adibidea. Grisean nabarmendua dauden elementuak dira helburu-funtzioan batzen direnak.

Instantziei dagokienez, *uniformeki ausaz* esaten denean, optimizazioaren ikuspegitik bi egoera bereiz ditzakegu, soluzioa bilatzeko erabiltzen den algoritmo-motaren arabera. Alde batetik, soluzio bakoitzari dagokion helburu-funtzioaren balioa modu esplizituan kontuan hartzen duten algoritmoak ditugu; esate baterako, *Simulated Annealing* [18] algoritmoa edo eruleta-hautestepena edota Boltzmann-en hautespena erabiltzen duten

edozein algoritmo ebolutibo [19]. Beste aldetik, soluzioei dagokien helburu-funtzioaren balioen konparazioa bakarrik erabiltzen duten algoritmoak daude (A soluzioa B soluzioa baino txarragoa denetz, alegia). Multzo honetan, tabu-bilaketa, *Variable Neighbourhood Search* algoritmoa edota ranking-hautespena erabiltzen duen edozein algoritmo dago, besteak beste. Lehenengo taldeari dagokionez, edozein problema kontsideratuz gero, bereiz daitezkeen helburu-funtzioen kopurua infinitua da; hau da, instantzia osatzen duten parametroei balioak aldatuz lortzen ditugun instantzia guztiak ezberdinak dira. Aldiz, bigarren taldeari dagokionez, funtzio kopurua finitua da; izan ere, talde horretako algoritmoek funtzioak bilaketa-espazioko soluzio guztien rankingak bailiran ikusten dituzte. Beraz, n tamainako edozein problemarentzat, sor daitezkeen ranking kopurua (eta, hortaz, helburu-funtzioen kopurua) $|\Omega|!$ da.

Ondorengo parrafoetan, permutazio-problema batzuk adibide gisa hartuta, parametroen espazioan uniformeki ausaz lagintzea eta rankingen espazioan uniformeki ausaz lagintzea berdina ez dela ikusiko dugu. Zehazki, LOP, PFSP eta QAP problemak aztertuko ditugu. LOPa adibide moduan hartuta egindako esperimenterazioa deskribatuko dugu eta, ondoren, esperimenterazioa beste bi problemetara hedatuko dugu. Azkenik, problemaren zailtasunari begira³, baliokidetasuna ez betetzeak dituen ondorioak azalduko ditugu.

Arrazoi teknikoak direla tarteko, optimizatu beharreko helburu-funtzioa, f , injektiboa dela suposatuko dugu (hau da, bilaketa-espazioko soluzio bakoitzak helburu-funtzio balio desberdina duela)⁴. Hori horrela, Ω ko soluzio guztiak ranking bat bezala ikus ditzakegu, non soluziorik onena lehenengo posizioan dagoen, bigarren soluziorik onena bigarren posizioan, eta abar. Artikulu honetan permutazio-problemekin dihardugunez, n delarik problemaren tamaina, Ω ko soluzio kopurua (bilaketa espazioaren tamaina, alegia) $n!$ da (n tamainako permutazio guztiak). Kontuan izan behar da ranking berbera sortzen duten instantzien kopurua infinitua dela, eta beraz, bigarren taldeko algoritmoei dagokienez, optimizatu beharreko

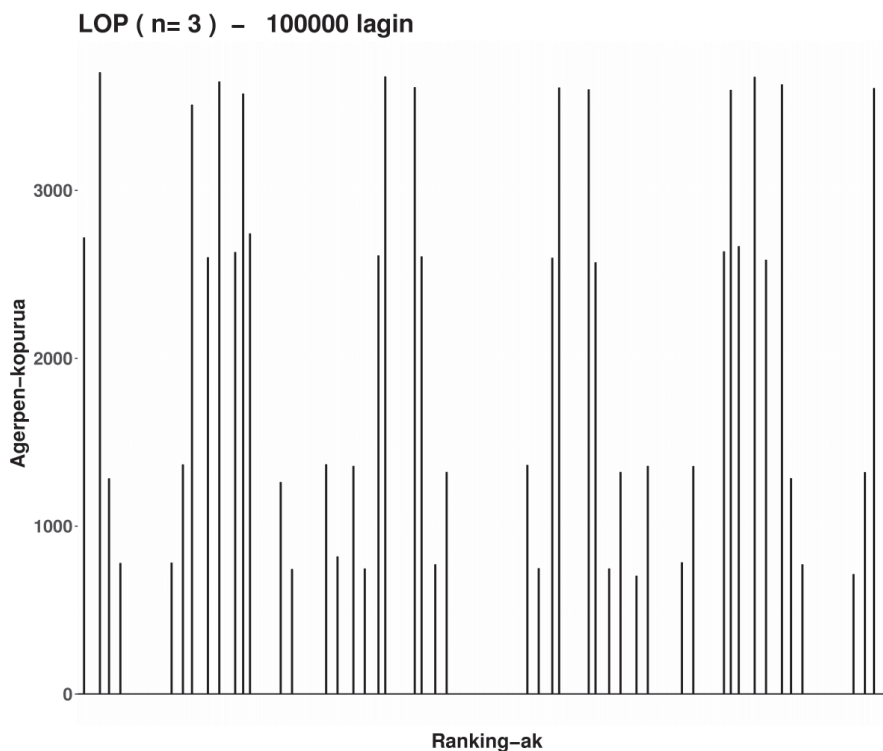
³ Optimizazio konbinatorioan, ez da existitzen zailtasunaren definizio formalik; hori dela eta, sarritan termino ambigua suerta dakiguke. Izan ere, instantzia baten zailtasuna, hainbat faktoreren artean, optimizatzeko erabili nahi den algoritmoari dago erabat lotuta. Hori dela eta, lan honetan, zailtasuna definitzeko, bilaketa lokaleko algoritmoetara mugatuko gara, eta optimo lokalen kopuruarekin lotuko dugu. Zenbat eta optimo lokal gehiago, orduan eta instantzia zailagoa.

⁴ Onarpen hori ez da oso murrizketa gogorra. Funtzioa injektiboa ez izateko, parametroen bi azpimultzo independente egon behar dira, zeinen batura berdina den. Esperimenterazioan parametroak \mathbb{R} multzoan uniformeki lagintzen dira eta, berez, edozein azpimultzoaren (finituaren) batura ere \mathbb{R} multzoan egongo da eta edozein balio izango da posible. Hortaz, bi azpimultzo independenteen batura berdina izateko infinitesimalki txikia izango da.

problema berdina dela. Hori dela eta, nahiz eta helburu-funtzioen kopurua infinitua izan, funtzio horiek sor dezaketen ranking kopurua $(n)!$ da gehienez.

3.1. LOP

Aurrekoak kontuan izanik, jarraian esperimentu labur bat egingo dugu. LOP problema hartuta, $n = 3$ tamainako 100.000 instantzia sortuko ditugu, horiek osatzen dituzten $b_{k,l}$ parametroak $[0, 100]$ tartean uniformeki laginduz. Instantzia bakoitzari dagokion soluzio-rankinga kalkulatu dugu, eta azkenik, ranking bakoitza zenbat alditan errepikatuta agertzen den zenbatu dugu. Tamaina $n = 3$ izanik, sortu daitekeen ranking kopuru maximoa $(3)! = 6! = 720$ da. Emaitzak 2. irudian erakusten dira, non x ardatzean rankingak ordena lexikografikoan adierazten diren.



2. irudia. 10^5 instantzia dagozkien parametroak $[0, 100]$ tartean uniformeki auzaz lagindu ostean sortutako rankingen agerpen kopurua. Kontuan izan behar da x ardatzean sor daitezkeen $(n)!$ rankingak kontsideratzen direla, hau da, $n = 3$ tamainarako 720.

2. irudian ikusten diren emaitzak kontuan izanik, hainbat ondorio interesgarri atera ditzakegu. Lehenik eta behin, begi-bistakoa da 720 ranking posibleetatik gutxi batzuk agertu direla, eta gehienak ez direla behin ere agertu. Honen arrazoia LOParen simetrian dago. Diagonal nagusiaren gainean dauden parametroen batura maximizatzea diagonal nagusiaren azpian dauden parametroen batura minimizatzearen baliokidea da. Rankingeko lehenengo posizioan agertzen den soluzioa σ_1 bada, orduan, rankingaren azkenengo posizioan dagoen soluzioa, $\sigma_{n!}$, σ_1 en iraulia (*reverse*) izan behar da⁵. Ezaugarri hori rankingeko gainerako soluzioetara ere hedatzen da. Ondorioz, LOPak, $(n!)!$ soluzio-ranking sortu beharrean, gehienez, honako soluzio-ranking sortuko ditu:

$$\left(\frac{n!}{2}\right)! \sum_{i=0}^{n!/2} \binom{n!/2}{i} = \frac{n!}{2} 2^{n!/2} \quad (2)$$

non $\sum_{i=0}^{n!/2} \binom{n!/2}{i}$ aurreko propietatea betetzen duten soluzio-multzoen ko-

purua den (multzo horietan ez daude bata bestearen irauliak diren bi permutazio), eta $(n!/2)$ multzo bakoitzaren ranking posiblea guztiak dira. Gure adibidean, non $n = 3$ den, 48 ranking desberdin izango ditugu soilik.

Ranking-espazioaren hutsuneaz gain, irudiak erakusten duen histograma simetrikoa dela hauteman daiteke. Hau da, x ardatzean kokatzen diren lehen eta azken rankingak kopuru berean agertu direla ikusi dezakegu⁶. Horren arrazoia ere LOPak duen simetria-propietatea da. Irudian, x ardatzaren lehen erdian dagoen edozein rankingetako permutazioak banan-banan atzekoz aurrera jarriko bagenitu, irudiaren bigarren erdian dauden rankingak lortuko genituzke. Simetria hori matrizeen espazioan iraultze-eragiketarik sortzen duen simetriaren isla da.

Azkenik, irudiko histogramen altuerari erreparatuz, agerikoa da esatea aurrera eramandako esperimentu horretan parametroak uniformeki ausaz laginduz ez dela rankingen lagin uniforme bat lortu. Are gehiago, diferentziak nabarmenak direla ikusi dezakegu lau ranking-multzo identifikatuz:

- Gehien agertu diren rankingen taldea XL motakoa dela esango dugu aurrerantzean. Horien agerpen kopurua [3431, 3681] tartera mugatu da.
- Agerpen kopurua [2410, 2617] tartean izan duten rankingak L motakoak izendatu ditugu.

⁵ Adibidez, $\sigma_1 = (14235)$ bada dugun permutazioa, haren iraulia $\sigma_{n!} = (53241)$ da.

⁶ Histograma laginketa batean oinarritzen den hurbilketa da, gutxi gorabehera.

Zorizko instantzia uniformeak sortzen al dira optimizazio konbinatorioan?

- Agerpen kopurua [1139, 1345] tartean izan duten rankingak M motakoak izendatu ditugu.
- Agerpen kopurua [723, 797] tartean izan duten rankingak S motakoak izendatu ditugu.

$$\left. \begin{array}{l} 123 \rightarrow b_{12} + b_{13} + b_{23} \\ 132 \rightarrow b_{13} + b_{12} + b_{32} \\ 213 \rightarrow b_{21} + b_{23} + b_{13} \\ 312 \rightarrow b_{31} + b_{32} + b_{12} \\ 231 \rightarrow b_{23} + b_{21} + b_{31} \\ 321 \rightarrow b_{32} + b_{31} + b_{21} \end{array} \right\} \begin{array}{ll} b_{23} & > b_{32} \\ b_{12} + b_{32} & > b_{21} + b_{23} \\ b_{21} + b_{23} + b_{13} & > b_{31} + b_{32} + b_{12} \\ b_{32} + b_{12} & > b_{23} + b_{21} \\ b_{23} & > b_{32} \end{array}$$

3. irudia. XL motako ranking baten adibidea, dagozkion helburu-funtzioen balioen deskonposaketekin, eta ondoz ondoko soluzioen arteko murrizketekin.

$$\left. \begin{array}{l} 213 \rightarrow b_{13} + b_{21} + b_{23} \\ 123 \rightarrow b_{12} + b_{13} + b_{23} \\ 132 \rightarrow b_{13} + b_{12} + b_{32} \\ 231 \rightarrow b_{23} + b_{21} + b_{31} \\ 321 \rightarrow b_{32} + b_{31} + b_{21} \\ 312 \rightarrow b_{31} + b_{32} + b_{12} \end{array} \right\} \begin{array}{ll} b_{21} & > b_{12} \\ b_{23} & > b_{32} \\ b_{13} + b_{12} + b_{32} & > b_{23} + b_{21} + b_{31} \\ b_{23} & > b_{32} \\ b_{21} & > b_{12} \end{array}$$

4. irudia. L motako ranking baten adibidea, dagozkion helburu-funtzioen balioen deskonposaketekin, eta ondoz ondoko soluzioen arteko murrizketekin.

Aurreko emaitzak ulertzeko asmotan, multzo bakoitzeko ranking bat aukeratu dugu ausaz, eta 3, 4, 5 eta 6. irudietan irudikatu dugu bakoitza. Horrez gain, ranking bakoitzaren alboan, soluzio bakoitzaren helburu-funtzioaren balioa kalkulatzeko erabilitako parametroak (b_{kl}) adierazi ditugu. Azkenik, soluzio-rankinga sortzeko, bete behar diren parametroen murrizketak adierazi ditugu.

3. irudiko adibidera joz gero, ikus daiteke (123) dela permutazio onena, eta (132) permutazioak jarraitzen diola. Lehenengo permutazioari dagokionez, haren helburu-funtzioaren balioa $b_{12} + b_{13} + b_{23}$ da, eta (132) soluzioari

dagokiona $(b_{13} + b_{12} + b_{32})$ baino handiagoa izan behar da. Sinplifikatuz, $b_{23} > b_{32}$ murrizketa bete behar da (gogoratu helburu-funtzioak injektiboak direla, eta, beraz, berdinketarik ez dagoela).

Irudiak konparatzen baditugu, ranking multzo bakoitzean murrizketen egiturak desberdinak direla ikusten da. S multzoko adibidera joz gero, begi-bistakoa da haren murrizketak, nabarmen gogorrak direla XL multzokoenak baino (multzo berean dauden rankingek murrizketa-egitura berbera partekatzen dute). Argi dago murrizketa horiek $[0, 100]n$ espazioan definitzen dituzten hiperbolumenen tamainak proportzionalak direla ranking bakoitzaren agerpen kopuruarekiko. Zehazki, hiperbolumenen tamainak, hurrenez hurren, $P(XL) = 13/30$, $P(L) = 19/60$, $P(M) = 19/120$ eta $P(S) = 11/120$ dira multzo bakoitzerako, eta bat datoz atera ditugun agerpenen kopuruaren tarteeekin.

$$\left. \begin{array}{l} 321 \rightarrow b_{32} + b_{31} + b_{21} \\ 132 \rightarrow b_{13} + b_{12} + b_{32} \\ 312 \rightarrow b_{31} + b_{32} + b_{12} \\ 213 \rightarrow b_{21} + b_{23} + b_{13} \\ 231 \rightarrow b_{23} + b_{21} + b_{31} \\ 123 \rightarrow b_{12} + b_{13} + b_{23} \end{array} \right\} \begin{array}{ll} b_{31} + b_{21} & > b_{13} + b_{12} \\ b_{13} & > b_{31} \\ b_{31} + b_{32} + b_{12} & > b_{21} + b_{23} + b_{13} \\ b_{13} & > b_{31} \\ b_{31} + b_{21} & > b_{13} + b_{12} \end{array}$$

5. irudia. M motako ranking baten adibidea, dagozkion helburu-funtzioen balioen deskonposaketekin, eta ondoz ondoko soluzioen arteko murrizketekin.

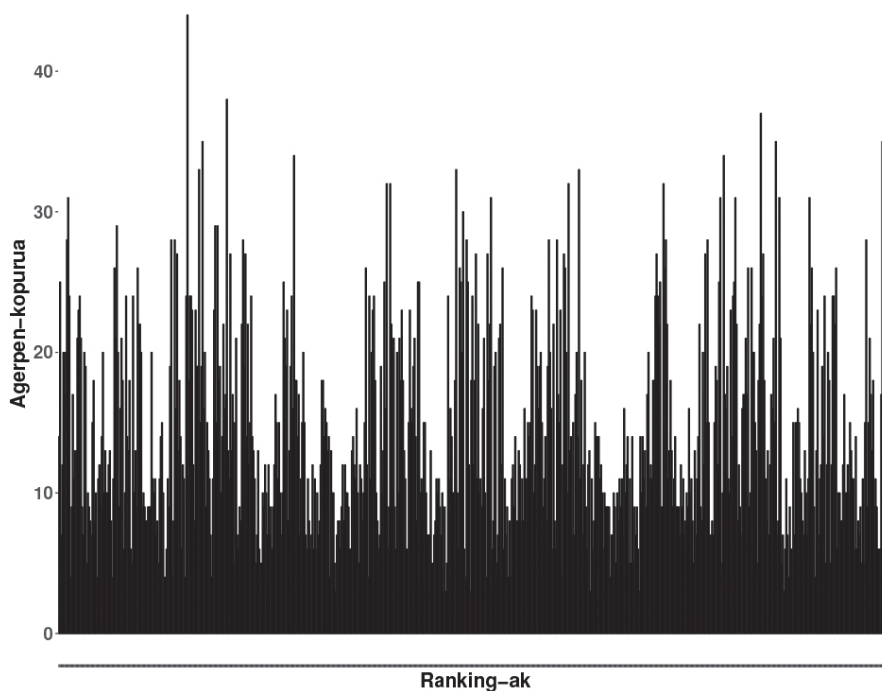
$$\left. \begin{array}{l} 213 \rightarrow b_{21} + b_{23} + b_{13} \\ 321 \rightarrow b_{32} + b_{31} + b_{21} \\ 132 \rightarrow b_{13} + b_{12} + b_{32} \\ 231 \rightarrow b_{23} + b_{21} + b_{31} \\ 123 \rightarrow b_{12} + b_{13} + b_{23} \\ 312 \rightarrow b_{31} + b_{32} + b_{12} \end{array} \right\} \begin{array}{ll} b_{23} + b_{13} & > b_{32} + b_{31} \\ b_{31} + b_{21} & > b_{13} + b_{12} \\ b_{13} + b_{12} + b_{32} & > b_{23} + b_{21} + b_{31} \\ b_{21} + b_{31} & > b_{12} + b_{13} \\ b_{13} + b_{23} & > b_{31} + b_{32} \end{array}$$

6. irudia. S motako ranking baten adibidea, dagozkion helburu-funtzioen balioen deskonposaketekin, eta ondoz ondoko soluzioen arteko murrizketekin.

3.2. QAP eta PFSP

Aurrekoak ikusirik, esperimentua beste bi permutazio-problematara hedatu dugu: esleipen-problema koadratikora (*Quadratic Assignment Problem*, *QAP*, ingelesez), eta *Permutation Flowshop Scheduling Problem* problemara (PFSP) [20]⁷.

QAP (n = 3) – 100000 lagin



7. irudia. 10^5 QAP instantzia dagozkien parametroak $[0, 100]$ tartean uniformeki ausaz lagindu ostean sortutako rankingak eta haien kopurua. Kontuan izan x ardatzean sortu daitezkeen $(n)!$ rankingak kontsideratzen direla; hau da, $n = 3$ rako, 720.

Quadratic Assignment Problem (QAP). *Izan bitez n lantegi, n kokapen posible, $\mathbf{H} \in \mathbb{R}^{n \times n}$ (lantegien arteko lan-kargen matrizea), eta $\mathbf{D} \in \mathbb{R}^{n \times n}$ (kokapenen arteko distantzien matrizea). QAP problemaren helburua da*

⁷ Saltzaile ibiltariaren problema beste permutazio-problema ezagun bat da; tamalez, kasu hartan f ez da injektiboa, eta beraz, esperimentuaren emaitzak ezin izango lirateke testuinguru berean aztertu.

lantegi bakoitza kokapen batean kokatzea kostu totala minimizatuz. Kokapen-esleipena n tamainako permutazio baten bitartez kodifikatuz gero (non σ_i baita i -garren kokalekuari esleitutako lantegia), helburu-funtzioa honela adieraz daiteke:

$$f(\sigma) = \sum_{i=1}^n \sum_{j=1}^n h_{\sigma_i, \sigma_j} \cdot d_{i,j} \quad (3)$$

Permutation Flowshop Scheduling Problem (PFSP). Izan bitez n lan ($i = 1, \dots, n$), m makina ($j = 1, \dots, m$) eta $P \in \mathbb{R}^{n \times m}$, prozesatze-denboren matrizea, non $p_{i,j}$ -k adierazten baitu zenbat denbora behar duen i lana j makinan prozesatzeko. Lan bakoitza egiteko, m prozesu aplikatu behar dira; bakoitza, makina batean. Behinjin makinan sartzen denean, i lana eten barik prozesatuko da, eta denbora zehatz bat emango du bertan, $p_{i,j}$. i lana j makinatik irteten denean, $j + 1$ makinara pasatuko da, hurrengo prozesua egitera, baldin eta makina libre badago. PFSParen helburua prozesatze-denbora totala minimizatzen duen n lanen sekuentzia optimoa aurkitzea da.

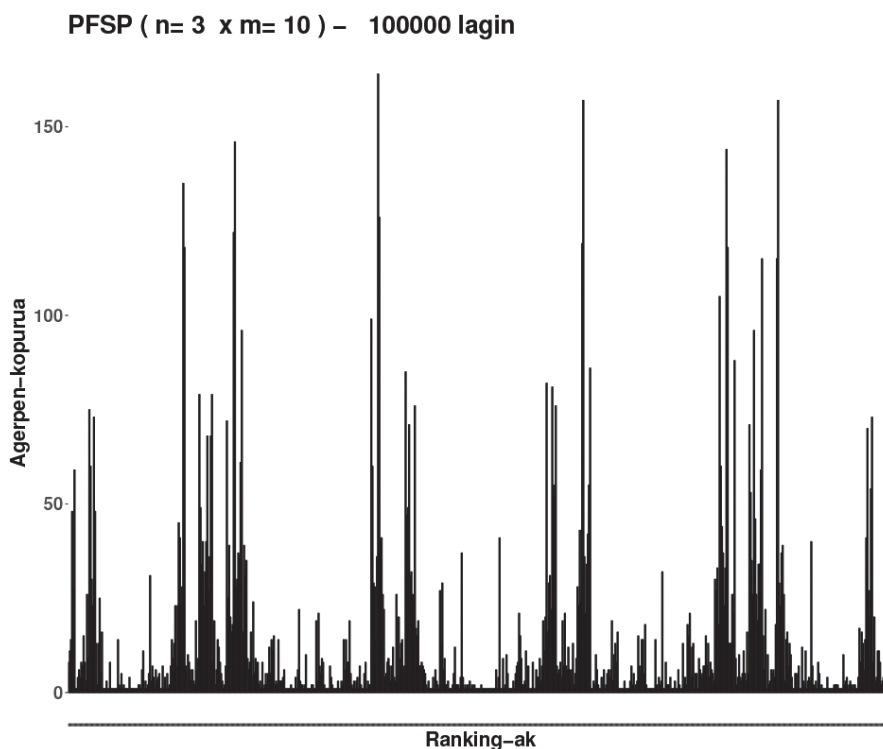
Gure esperimentazioan, denbora totala *total flow time (TFT)* izango da. Irizpide horrek lan bakoitza prozesatzeko behar izan diren denbora guztiak batzen ditu. Soluzio bat emanik (σ) TFTaren adierazpen matematikoa jarraian dagoen ekuazioak formalizatzen du:

$$f(\sigma) = \sum_{i=1}^n c_{\sigma_i, m} \quad (4)$$

non $c_{\sigma_i, m}$ baita σ_i lana burutzeko behar izan den denbora; honela kalkulatu da:

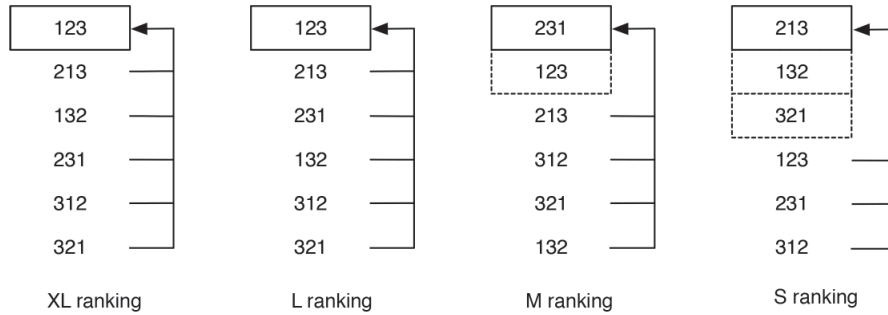
$$c_{\sigma_i, m} = \begin{cases} p_{\sigma_i, j} & i = j = 1 \\ p_{\sigma_i, j} + p_{\sigma_{i-1}, j} & i > 1, j = 1 \\ p_{\sigma_i, j} + p_{\sigma_{i, j-1}} & i = 1, j > 1 \\ p_{\sigma_i, j} + \max\{c_{\sigma_{i-1}, j} + c_{\sigma_{i, j-1}}\} & i > 1, j > 1 \end{cases}$$

Lehen esan bezala, LOPan egindako esperimentua QAP eta PFSP problemetan errepikatu ostean lortutako emaitzak 7. eta 8. irudietan aurkezten dira. Kasu honetan ere, problemaren tamaina (n) 3 izan da. Irudi horiek LOP kasuan lortutakoekin alderatuta, zenbait puntu azter ditzakegu.



8. irudia. 10^5 PFSP instantzia dagozkien parametroak $[0, 100]$ tartean uniformeke ausaz lagindu ostean sortutako rankingak eta haien kopurua. Kontuan izan x ardatzean sortu daitezkeen $(n)!$ rankingak kontsideratzen direla; hau da, $n = 3$ rako, 720.

Hasteko, LOP kasuan ez bezala, bai QAPko esperimentazioan bai PFSPkoan ranking guztiak agertzen dira. Horrez gain, nahiz eta irudietan ondo ez ikusi, bi problemetan simetria dago rankingetan; alegia, permutazio guztien kokapena bere irauliarekin erlazionatuta dago. Azkenik, problema horietan ere rankingak multzokatzen dira. Alabaina, PFSParen kasuan nabarmen altuagoa da ranking-multzoen agerraldi kopuruaren bariantza LOPan eta QAPan baino. Balioak ikusirik, portaera horren jatorria helburu-funtzioaren definizioan dagoela dirudi. LOParen kasuan, $n(n - 1)$ parametroren batuketa egiten da. QAParen kasuan, bi parametroren arteko n^2 biderketen batura egiten da. PFSPan ordea, funtzioa askoz ere konplexuagoa da, errekurtsiboki kalkulatzen baita, eta baturez gain, pauso bakoitzean batura maximoak kontsideratzen ditu.



9. irudia. LOPko ranking-multzo bakoitzeko (XL, L, M eta S) ranking bat erakusten da. Marra etenez inguratzen diren soluzioak *interchange* ingurunerako optimo lokalak dira. Eta marra etengabez, optimo globala adierazten da. Geziek adierazten dute non amaitzen den bilaketa optimo lokala ez den soluzio bakoitzetik abiatuta.

Laburbilduz: esperimentalki frogatuta geratu da ezen, instantziak parametroen espazioan uniformeki ausaz laginduz, ranking batzuk gehiagotan sortzen direla beste batzuk baino eta, beraz, ez dugula probarako funtzioen lagin uniformerik lortzen.

Emaitzak horrek, beste galdera aipagarri pare bat dakarzkigu:

- Instantziak parametroen espazioan uniformeki ausaz laginduz sortutako rankingen agerpen kopuruan oinarrituta, posible al da rankingen zailtasunaren inguruko aipamenik egitea? (Demagun zailtasuna optimo lokal kopuru gisa neurtzen dugula)? Aurreko prozedurari jarraituz, esan al genezake ranking zailak lagintzeko probabilitatea baxua dela, eta aldiz ranking errazak lagintzekoa altuagoa?
- Zer erlazio dago laginduak izateko antzeko probabilitatea duten rankingen artean? Adibidez, LOParen kasuan, $n = 3$ tamainarako, lau ranking-multzo ikusi ditugu (XL, L, M edo S). Talde bakoitzaren barruan dauden rankingak, zailtasun-ikuspuntutik, baliokideak al dira? Erantzuna baiezkua balitz, kasu horretarako, LOPak 4 kasu besterik ez luke izango (720tik) algoritmoak probatzeko.

Galdera horiei erantzuteko saiakera bat egiteko asmoz, LOPan zentratuko gara berriro. Dagoeneko aipatu dugu ranking-multzo bakoitzean dauden rankingek murriztapen-egitura partekatzen dutela. Baina zein da erlazioa zailtasunarekiko erlazioa? Galderari erantzuteko, esperimentu labur bat garatuko dugu: *interchange* ingurunean⁸ darabilen bilaketa lokaleko al-

⁸ σ_b permutazio bat σ_a ren bizilaguna dela esango dugu baldin eta batetik bestera posizio banatan dauden bi elementu elkar lekuz aldatuz heltzen bagara batetik bestera.

goritmo bat exekutatu dugu ikusitako 48 rankingetan, ranking bakoitzeko soluzio bakoitza abiapuntu-soluziotzat hartuta. Bilaketa amaitzean, bukaerako soluzioak identifikatu ditugu, halabeharrez optimo lokalak (edo globala) izan behar dutenak, hain zuzen. Rankingaren zailtasuna, lortutako optimo lokalen kopuruarekin erlazionatuko dugu [21]. Emaitzen arabera, ranking-multzo berean dauden ranking guztiek optimo lokalen kopuru bera dute, eta soluzio-rankingean posizio berberetan daude kokatuta. Emaitzak 9. irudian aurkeztuak dira era laburbilduan.

Irudian ikus daitekeen bezala, XL eta L motetako rankingek, parametroak uniformeki ausaz laginduz agerpen gehien izan dituztenek, optimo lokal bakarra dute (globala). Aldiz, S motako rankingek, agerraldi gutxirekin, hiru optimo lokal dituzte.

4. ONDORIOAK

Lan honetan, optimizazio-algoritmoen errendimendua neurtzeko erabiltzen diren instantzia artifizialak ausaz sortzearen inguruan eztabaidatu dugu. Aztertutakoaren arabera, ausazko instantziak sortzerakoan, haiek zer algoritmo motarentzat diren kontuan izan behar dugu. Izan ere, ez dira berdinak helburu-funtzioaren balioak era esplizituan kontuan izaten dituzten algoritmoen kasua eta kontuan izaten ez dituztenena. Bigarrenen kasuko algoritmoak izan ditugu aztergai, bilaketa-espazioko soluzioak ranking bat balira bezala ikusten dituztenak. Hiru problemaz baliaturik, frogatu dugu erabat desberdinak direla instantziak parametro-espazioan lagintzea eta ranking-espazioan uniformeki ausaz lagintzea. Are gehiago, emaitzen arabera, laginketa parametroen espazioan eginez gero, rankingak multzokatu daitezke beren agerpen-probabilitatearen arabera. Ez hori bakarrik: garatutako azterketaren arabera, agerpen-probabilitate bera duten rankingek zailtasun-ezaugarriak partekatzen dituzte. Probabilitate baxuena duten instantziek optimo lokalen kopuru handiena dutela erakutsi du gure esperimentazioak.

Garatutako esperimentuen tamaina oso mugatua izan arren ($n = 3$), ondorioak beste tamaina eta problema batzuetan ere aplikagarriak dira. Hau da, ohiko prozedurak (parametroen espazioan uniformeki ausaz lagintzeak) ez ditu ematen ustez suposatzen ditugun zailtasun uniformeko funtzio/instantzia/proba-kasuak. Zentzu horretan, garrantzitsua da ikertzaileek emaitza horien berri izatea, prozedura egokiak garatu behar direlako instantzia artifizialak sortzeko.

Ikusi dugun bezala, $n = 3$ tamainarekin LOPak 48 ranking desberdin sortu ditu, eta espresio itxi bat eman dugu horiek kontatzeko; baina, $n > 3$ denean, posibleak al dira simetria betetzen duten ranking guztiak? Ez dirudi konplexua denik simetria betetzen duen ranking bat sortzea, non on-

doz ondoko bi soluzioren arteko murrizketen artean kontraesan bat dagoen (ikusi 3, 4, 5 eta 6. irudietako inekuazioak). Era berean, QAPri eta PFSPri dagokienez, 720 ranking desberdin agertu dira, hau da, (3!)!-ak; baina, (4!)!-rako, sor al daitezke 24! rankingak? Galdera horiek guztiek etorkizunerako ikerketa-ildo interesgarri bat proposatzen dute.

5. ESKER ONAK

Eskerrak eman nahi dizkiegu lan hau egiteko finantziazioa eman diguten erakunde eta programei; horien artean, Eusko Jaurlaritzaren Ikerketa Taldeen 2013-2018 programa (IT-567-13) dago, eta Espainiako Ekonomia, Industria eta Lehiakortasun Ministerioa (TIN2016-78365R), BERC 2014-2017 programa, Eusko Jaurlaritzaren Elkartek programak eta Severo Ochoa programa (SEV-2013-0323).

6. BIBLIOGRAFIA

- [1] CEBERIO J., MENDIBURU A. eta LOZANO J.A. 2017. «Are we generating instances uniformly at random?». In *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*, 1645-1651.
- [2] MENDIBURU A., MIGUEL-ALONSO J., LOZANO J.A, OSTRA M. eta UBIDE C. 2016. «Parallel EDAs to create multivariate calibration models for quantitative chemical applications». *Journal of Parallel and Distributed Computing*, **66(8)**, 1002-1013.
- [3] CEBERIO J., IRUROZKI E., MENDIBURU A. eta LOZANO J.A. 2014. «A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem». *IEEE Transactions on Evolutionary Computation*, **18(2)**, 286-300.
- [4] SRINIVASAN D. eta HUA Z. 2009. «Solving Time-Tabling Problems Using Evolutionary Algorithms and Heuristics Search». *Multi-objective Memetic Algorithms*, 53-69. Springer Berlin Heidelberg.
- [5] BROWNLEE A.E.I, PELIKAN M., McCALL J.A.W. eta PETROVSKI A. 2008. «An application of a multivariate estimation of distribution algorithm to cancer chemotherapy». In *Proceedings of the 2008 Genetic and Evolutionary Computation Conference*, 463-464.
- [6] CEBERIO J. 2014. *Solving permutation problems with estimation of distribution algorithms and extensions thereof*. Faculty of Computer Science, University of the Basque Country.
- [7] GENT I.P. eta WALSH T. 1996. «The TSP phase transition ». *Artificial Intelligence*, **88(1-2)**, 349-358.
- [8] TAILLARD E. 1993. «Benchmarks for basic scheduling problems». *European Journal of Operational Research*, **64(2)**, 278-285.

- [9] BRATLEY P., FOX B.L. eta SCHRAGE L.E. 1986. *A guide to simulation (2Nd Ed.)*. Springer-Verlag New York, USA.
- [10] BURKARD R.E., KARISCH S.E. eta RENDL F. 1997. «QAPLIB – a quadratic assignment problem library». *Journal of Global Optimization*, **10(4)**, 391-403.
- [11] DREZNER Z., HAHN P.M. eta TAILLARD E.D. 2005. «Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods». *Annals of Operations Research*, **139(1)**, 65-94.
- [12] GRÖTSCHEL M., JÜNGER M. eta REINELT G. 1983. «Optimal triangulation of large real world input-output matrices». *Statistische Hefte*, **25(1)**, 261-295.
- [13] SCHIAVINOTTO T. eta STÜTZLE T. 2005. «The linear ordering problem: Instances, search space analysis and algorithms». *Journal of Mathematical Modelling and Algorithms*, **3(4)**, 367-402.
- [14] DUARTE A., LAGUNA M. eta MARTÍ R. 2011. «Tabu search for the linear ordering problem with cumulative costs». *Computational Optimization and Applications*, **48(3)**, 697-715.
- [15] MARTÍ R. eta REINELT G. 2011. *The linear ordering problem: exact and heuristic methods in combinatorial optimization*. Volume 175. Springer.
- [16] CHENERY H.B. eta WATANABE T. 1958. «International comparisons of the structure of production». *Econometrica*, **26(4)**, 487-521.
- [17] CEBERIO J., MENDIBURU A. eta LOZANO J.A. 2014. «The linear ordering problem revisited». *European Journal of Operational Research*, **241(3)**, 686-696.
- [18] KIRKPATRICK S., GELATT C.D. eta VECCHI M.P. 1983. «Optimization by simulated annealing». *Science*, **220(4598)**, 671-680.
- [19] CEBERIO J., MENDIBURU A. eta LOZANO J.A. 2016. «A note on the Boltzmann distribution and the linear ordering problem». In *Conference of the Spanish Association for Artificial Intelligence. Advances in Artificial Intelligence*. 441-446. Springer.
- [20] BAKER K.R. 1974. *Introduction to sequencing and scheduling*. Wiley.
- [21] ECHEGOYEN C., MENDIBURU A., SANTANA R. eta LOZANO J.A. 2013. «On the taxonomy of optimization problems under estimation of distribution algorithms». *Evolutionary Computation*, **21(3)**, 471-495.