Pràctica de Cerca

Intel·ligència Artificial Primavera 2022/23

Rubén Aciego

Mariona Jaramillo

Francesc Pifarré

Contingut

1. Descripció del problema

El problema plantejat és la necessitat de reduir la despesa energètica i les emissions de CO2 en les ciutats, on el transport privat és un dels majors generadors de contaminació. Una de les iniciatives proposades és la compartició de cotxes entre persones, és a dir, de manera rotativa durant un període, una persona utilitza el seu cotxe per traslladar persones al seu lloc de feina, de manera que hi ha més cotxes que tenen més places ocupades. D'aquesta manera es redueix el trànsit, el consum de combustibles i l'emissió de contaminants. Com que la solució espontània no és l'òptima, en aquesta pràctica estudiem si, fent servir algorismes de cerca local, podem obtenir solucions que indiquin directament com s'han d'organitzar els usuaris, sabent a priori on viuen i on treballen i si en un mes donat poden conduir el seu cotxe o no.

Elements del problema

Els elements que caracteritzen el problema són:

- Tenir un conjunt d'usuaris apuntats al servei de compartició de cotxes
- Tenir un subconjunt d'aquests usuaris que potencialment poden conduir en certs mesos, que estan disposats a oferir les places lliures del seu cotxe a altres persones i que poden adaptar el seu recorregut de casa a la feina per recollir altres usuaris i dur-los al seu lloc de feina
- Tenir la direcció en la que viuen i en la que treballen tots els usuaris
- Cada mes i de manera rotativa, tots els usuaris han de quedar assignats i han de poder arribar a la feina, verificant unes restriccions de problema i verificant que aquesta assignació pertanyi a l'espai de solucions definit més endavant

Les restriccions del problema són, doncs:

- Se suposa que la ciutat és un quadrat de 10x10km, amb carrers que formen una quadrícula d'illes de 100x100metres, per tant, els usuaris expressen les seves coordenades de casa i de feina sobre aquesta quadrícula (on una unitat de quadrícula equival a 100m)
- El càlcul de les distàncies es fa mitjançant la distància de Manhattan sobre la quadrícula

Cerca local

Per resoldre el problema farem servir la cerca local. Aquesta metodologia consisteix en suposar que podem navegar sobre un espai de possibles solucions realitzant operacions sobre elles que permetin millorar-les. En aquest cas, el cost de les operacions no importa, només ens fixem en la qualitat de les solucions. És per aquest motiu que necessitem tenir funcions heurístiques que segons criteris que definim com a coherents i correctes pel problema a resoldre, ens permetin ordenar les possibles solucions.

Per aquesta pràctica, la cerca local és un bon mètode, ja que, ens permet trobar una solució vàlida i a més a més, de manera més ràpida i eficient que amb altres mètodes com el backtracking. A més, el problema verifica els requisits per poder aplicar cerca local:

1. Ser un problema d'optimització que té per objectiu generar un estat final segons una funció heurística, on el camí per arribar a la solució és irrellevant
2. Ser possible trobar una solució inicial amb un cost relativament baix
3. Poder definir operadors de canvi de solució que manipularan solucions completes
4. Poder definir una funció de qualitat/ heurística que medeixi com de bona és una solució

En els següents apartats veurem que efectivament podem trobar solucions inicials, definir operadors i funcions heurístiques, condicions necessàries per poder treballar amb un problema de cerca local.

Espai de cerca i espai de solucions

Com s'ha dit, per poder tractar amb problemes de cerca local, necessitem treballar sobre un espai de cerca. Aquest espai de cerca, en el nostre cas, serà l'espai de solucions. Començarem amb una solució inicial que, valgui la redundància, serà sempre solució del problema, i en cas que aparegui alguna no-solució mentre s'executa l'algoritme per arribar a la solució final, se li aplicarà molta penalització per poder tornar a estar dins l'espai de cerca que volem: el de solucions.

Definim a continuació què serà considerat solució en el problema de compartició de cotxes d'aquesta pràctica:

- De l'enunciat es dedueix que cada conductor pot fer un màxim de 30km per dia

- La solució ha de ser una assignació completa que reculli i deixi totes les persones del servei
- En un cotxe hi pot haver com a màxim 3 persones en cada moment, incloent el conductor
- Només les persones que són designades com a conductores en un mes poden conduir

Analitzem a continuació la mida de l'espai de cerca. Siguin n els usuaris apuntats al servei i siguin m ($m<n$) les persones que poden conduir en un mes donat, aleshores l'ordre de l'espai de solucions és $O(n^m)$ (ja que cada un dels n usuaris pot ser potencialment assignat a un dels m conductors diferents). Per tant, al ser un espai de solucions tan gran, és molt millor fer servir cerca local que backtracking, perquè sinó, el temps d'execució dels algoritmes seria massa elevat o faria que no poguessin acabar.

2. Estats del problema i representació

Considerem un estat del problema com un conjunt de trajectes. Definim un trajecte com aquell recorregut que fa un conductor donat conduint un cotxe entre la seva posició d'origen i el seu destí, podent realitzar durant aquest trajecte diferents accions: recollir passatgers (únicament en la seva respectiva posició d'origen) i deixar-los (únicament en la seva respectiva posició de destí). Així, no permetem desplaçaments "intermitjos" i només es consideraran possibles desplaçaments íntegres entre l'origen i el destí d'un usuari.

Així, el principal element de la classe estat serà el conjunt de trajectes, el qual s'implementa internament en forma d'un mapa (un HashMap per millorar l'eficiència espaial de l'estructura de dades) que té com a clau l'usuari que condueix el cotxe del trajecte i com a valor el trajecte en sí. Així, l'estat es pot descriure com una assignació de conductors al trajecte que han de realitzar, que seria un element de l'espai de cerca prèviament definit.

El trajecte l'hem implementat com una llista d'accions, on cada acció és o bé recollir o deixar un usuari i el nombre de passatgers resultant en el cotxe després de fer-ho. Aquest darrer valor resultarà útil posteriorment a l'hora d'implementar operadors en ser necessari per comprovar que no es violin les condicions del problema (no hi poden haver més de 3 persones en un cotxe).

En cada estat ens guardem també el conjunt total d'Usuaris, que són les dades essencials del problema i són necessàries per la implementació de mètodes en l'estat.

Finalment, per cada estat s'emmagatzema també la distància total recorreguda per la configuració actual de l'estat. Aquesta s'obté sumant les distàncies recorregudes en cada trajecte, que es poden obtenir amb facilitat en guardar també la posició on s'efectua cada acció de recollida o deixada. No s'emmagatzema de forma independent el nombre total de cotxes ja que aquest es pot obtenir trivialment consultant la mida del mapa que relaciona conductors amb trajectes, sent aquest valor la quantitat de cotxes en ús.

3. Representació i anàlisi dels operadors

Els operadors són les transformacions que ens permeten, mitjançant un estat de partida, arribar a un altre estat per continuar explorant en el procés de cerca. Donat el problema prèviament descrit, els operadors que hem escollit d'implementar són els següents: permutacions internes de passatgers, desassignar conductor i moure passatger. Creiem que aquests operadors són adequats pel problema ja que permeten explorar gran part de l'espai de cerca.

Operador permutacions internes

Començant per les permutacions internes de passatgers, ens referim a, en un mateix trajecte (ruta que fa un cotxe d'inici a final recollint i deixant passatgers pel camí) permutar l'ordre de dues accions consecutives. La justificació de l'operador és que, tot i que fos molt més senzill deixar un passatger immediatament després de recollir-lo, pot resultar en una ruta més òptima recollir-ne d'altres entre mig o deixar-ne d'altres que ja eren al cotxe en cas que les seves rutes es solapin.

Per aquest operador, però, s'han de considerar algunes restriccions per assegurar-se que no es violen les condicions de solució del problema i que la permutació té sentit. Primerament, el conductor ha d'estar exempt de qualsevol permutació (ha de ser el primer en agafar el cotxe i el darrer en sortir-ne). En segon lloc, cal tenir present el nombre de passatgers que hi ha al cotxe en el moment de la potencial permutació per veure si aquesta de sentit. Per analitzar-ho, hem distingit els següents casos:

a. DEIXAR 1, RECOLLIR 2 -> RECOLLIR 2, DEIXAR 1

   Referit a quan, en l'ordre del trajecte, es deixa a un passatger que identifiquem com a 1 i, posteriorment se'n recull un altre que identifiquem com a 2.

   Aquesta permutació es podrà fer només si portem ja un passatger. En altre cas podríem superar la capacitat màxima del cotxe.

b. DEIXAR 1, DEIXAR 2 -> DEIXAR 2, DEIXAR 1

   Referit a quan, en l'ordre del trajecte, es deixa a un passatger que identifiquem com a 1 i, posteriorment se'n deixa un altre que identifiquem com a 2.

   Aquesta permutació es pot fer sempre.

c. RECOLLIR 1, DEIXAR 2 -> DEIXAR 2, RECOLLIR 1

Referit a quan, en l'ordre del trajecte, es recull a un passatger que identifiquem com a 1 i, posteriorment se'n deixa un altre que identifiquem com a 2.

Aquesta permutació es podrà fer només si portem ja un passatger. En cas que no, el passatger 1 i 2 podrien resultar ser el mateix i no tindria sentit recollir-lo abans que deixar-lo.

d. RECOLLIR 1, RECOLLIR 2 -> RECOLLIR 2, RECOLLIR 1

   Referit a quan, en l'ordre del trajecte, es recull a un passatger que identifiquem com a 1 i, posteriorment se'n recull un altre que identifiquem com a 2.

   Aquesta permutació es pot fer sempre.

Factor de ramificació de l'operador: els intercanvis sempre es realitzen entre posicions consecutives i per tant en cas pitjor, on tots els usuaris estarien en el trajecte d'un únic conductor, tindrem com a molt 2n-3 possibles intercanvis (no podem intercanviar ni la primera acció ni les dues darreres). Per tant, això implica que el factor de ramificació és de O(n).

Operador desassignar conductor

Pel segon operador, enlloc de centrar-nos en, dins d'un trajecte, optimitzar-lo per intentar optimitzar la distància total recorreguda, ens centrarem en intentar millorar el nombre de cotxes totals. En aquest segon operador de desassignar conductors el que considerem és transformar conductors amb el seu respectiu cotxe en passatgers d'algun altre cotxe, disminuint així el nombre total de cotxes en circulació.

Apliquem aquest operador quan un trajecte el realitza únicament el conductor del cotxe (no recull ni deixa cap passatger pel camí). En aquest cas, eliminem el trajecte i recol·loquem el conductor en forma de passatger en algun dels altres trajectes.

En fer-ho, s'afegeix el conductor al final del recorregut dels altres. A cada possible trajecte, s'afegirà el conductor previ, ara passatger, de forma que se'l recollirà i immediatament se'l deixarà.

Factor de ramificació de l'operador: es poden desassignar m conductors i assignar-los als m-1 restants, per tant, el factor de ramificació és O(m^2). Observem però, que generalment serà molt menor doncs la condició de tenir un conductor amb un trajecte individual és molt restrictiva i no es tindrà sovint.

Operador moure passatger

Respecte el darrer operador, moure passatgers, aquest considera la possibilitat de moure entre trajectes algun dels passatgers. Així, per cada passatger de qualsevol trajecte, intenta re-col·locar-lo en algun altre trajecte (afegint-lo de forma que se'l recollirà i immediatament se'l deixarà tot i que en el seu trajecte anterior aquestes dues accions no eren consecutives). En el moment de colocar-lo en un altre trajecte es tenen en compte, de nou, les condicions de màxim nombre de passatgers en un cotxe.

Així, aquest operador permet una gran versatilitat en moure passatgers entre trajectes i permetent que es donin les condicions suficients per a que la resta d'operadors puguin actuar correctament.

Factor de ramificació de l'operador: com a molt es poden moure els n usuaris a qualsevol dels 2n esdeveniments (recollir o deixar). Per tant el factor de ramificació és de O(n^2).

4. Mètodes de generació de solucions inicials

Per tal de generar les solucions inicials, hem considerat dos mètodes diferents.

Mètode 1

Per generar solucions inicials amb el primer mètode seguim el procés següent:

- Totes les persones que poden conduir aquest mes es converteixen en conductores
- La resta de persones no conductores (passatgers) s'assignen al final del trajecte del conductor pel qual es minimitza la distància total recorreguda per tots els conductors si s'assigna aquell passatger a un d'ells.
- El procediment es repeteix fins que no queden més passatgers sense conductor.
- Si en algun moment, la millor assignació d'un passatger a un conductor suposa que aquest conductor hagi de fer més de 30km de trajecte, aquesta assignació es desestimarà i es considerarà la següent millor assignació del passatger a un altre conductor (i així successivament). Això permet garantir la restricció que diu que suposem que la velocitat mitjana del trànsit a la ciutat és de 30km/h, que el cotxe inicia el recorregut a les 7 del matí i ha de ser a la feina a les 8 com a molt tard; és el mateix que dir, doncs, que la distància del seu trajecte no pot superar els 30km.
- Si totes les possibles assignacions fan que es superin els 30km de trajecte, s'escull la que incrementi menys el màxim de distàncies que recorren els conductors.

Deixant de banda la restricció de distància màxima, l'estratègia d'inicialització serà sempre solució, ja que a banda de les restriccions que obliguem a complir en el procediment descrit, també es verifiquen la resta, que són:

- La solució ha de recollir i deixar totes les persones del servei.
- Només les m persones designades com conductors per aquell mes poden conduir.

Per tant, no sempre partim d'una solució inicial dins de l'espai de solucions, però mitjançant l'heurística penalitzarem les solucions que no compleixen la restricció de distància.

A més, el cost de trobar aquesta solució inicial és baix, $O(m+(n-m)*m)$, per tant, podem intuir que és un bon mètode.

Mètode 2

La nostra segona opció a estratègia d'inicialització consisteix en:

- Totes les persones que poden conduir aquest mes es converteixen en conductores
- La resta de persones no conductores (passatgers), s'assignen al conductor pel qual es minimitza la distància total recorreguda per tots els conductors si s'assigna aquell passatger a un d'ells. La diferència amb l'estratègia 1 és que ara no cal assignar-lo al final del trajecte del conductor, sinó que es proven tots els llocs possibles on recollir i deixar el nou passatger dels trajectes dels conductors existents i triem l'assignació que minimitza la distància total recorreguda per tots els conductors, com fèiem abans.
- El procediment es repeteix fins que no queden més passatgers sense conductor.
- Si en algun moment, la millor assignació d'un passatger a un conductor suposa que aquest conductor hagi de fer més de 30km de trajecte, es segueix el mateix tractament que en l'estratègia del primer experiment.
- Es comprova en tot moment que un cotxe no tingui mai més de 2 passatgers.

Com en el primer cas, aquesta estratègia d'inicialització donarà també solució inicial menys potser per la restricció de la distància màxima, ja que a banda de les restriccions que obliguem a complir en el procediment descrit, també es verifiquen la resta que havíem especificat en l'experiment 1.

El cost de trobar aquesta solució inicial és O(n^2). Tot i que és un cost més elevat que en el primer mètode, segueix sent un cost baix, per tant és un mètode vàlid. A més, aquest mètode aconsegueix una solució teòricament més bona ja que genera una assignació tenint una mica en compte les heurístiques, és a dir, minimitzar localment la distància total recorreguda (per fer la minimització global i tenir en compte la minimització de conductors de la segona heurística, és necessari córrer els algoritmes de cerca local).

5. Anàlisi de les funcions heurístiques

Per l'exploració i cerca d'una solució del problema hem plantejat dues heurístiques, considerant els dos objectius principals del problema que serien reduir la distància total recorreguda i el nombre de cotxes en circulació.

Aquests són, així, els factors a tenir en compte en possibles heurístiques del problema ja que ens indiquen la qualitat de la nostra solució: distància total recorreguda i el nombre de cotxes en circulació, on associarem a més petits siguin aquests valors millor podrem dir que és la solució.

La primera heurística que proposem, la més senzilla, calcularà les unitats de distància totals recorregudes per tots els trajectes del problema (serà equivalent a la distància total de l'estat anteriorment descrita). Aquesta heurística, doncs, penalitzarà distàncies totals més llargues alhora que afavoreix aquelles accions que permetin reduir la distància total.

Per tal de penalitzar aquells possibles estats que, tot i estar sent explorats, no siguin una solució del problema en violar la restricció de màxima distància recorreguda (que faria que el conductor no arribi a temps al seu destí), l'heurística, enlloc de donar la distància total recorreguda, donarà aquest valor escalat per un factor de 2.

Aquesta decisió ha estat presa per intentar conduir l'algoritme de cerca cap a estats continguts en el conjunt de solucions quan sigui possible, penalitzant les no solucions.

La segona heurística considera no només la distància recorreguda sinó també el nombre total de cotxes utilitzats. Per fer-ho, estimarem el factor de penalització d'un cotxe pel ratio distància/cotxes que s'obté a la solució inicial. Sumarem per tant la distància total i el nombre de cotxes multiplicat per la penalització per cotxe. Si suposem una solució inicial raonable, observem que els dos sumands tenen ordres de magnitud similars (de fet són iguals per la solució inicial) i per tant els dos tindran la mateixa rellevància a minimitzar en l'heurística.

$h2 = d + c * d0/c0$

on d és la suma de distàncies actuals (igual a la primera heurística), c és el nombre de cotxes (o conductors) de l'estat actual, d0 la suma de distàncies de la solució inicial i c0 el nombre de cotxes (o conductors) de la solució inicial.

Observem que des d'un punt de vista d'unitats aquesta heurística té sentit, doncs el segon terme també té unitats de distància.

Es manté també la penalització d'aquells estats que no formen part del conjunt de solucions, de nou amb un factor d'escalat de 2.

Utilitzar una heurística o altra hauria de tenir importants efectes en el procés de solució del problema, especialment en el relatiu al nombre de cotxes. Aquest, en no tenir cap impacte en el càlcul de la primera heurística, en solucionar el problema utilitzant-la no es tendirà a optimitzar el nombre de cotxes. Es previsible que el nombre de cotxes es pugui reduir igualment des de la solució inicial, ja que en reduir cotxes es pot reduir indirectament la distància total que és el que busca aquesta heurística, pero no es prioritzarà mai respecte una disminució en distància total.

Per la segona heurística, en canvi, sí que es beneficiarà directament aquelles solucions que tinguin un menor nombre de cotxes, així que podem preveure que tindrà un impacte important en el nombre de cotxes de les solucions obtingudes, que hauria de ser menor que si utilitzen l'altra heurística.

6. Experiments

Per l'experimentació tots els experiments es faran, per coherència, en el mateix ordinador Macbook Pro amb processador M1 Pro i amb la versió de Java 19.0.2.

Es correran els experiments amb un script de Python que passarà al programa els paràmetres necessaris per cada experiment i posteriorment, el mateix script, realitzarà les gràfiques de resultats utilitzant la llibreria ploty.

Experiment 1

L'objectiu del primer experiment serà analitzar l'ús dels operadors que hem desenvolupat en el procés de resolució del problema. Així, analitzarem quants cops s'usa cada operador en el camí que faci l'algoritme de Hill Climbing fins a la solució que ens proporciona del problema. Considerarem un ús de l'operador cada cop que, en la següent iteració, l'estat que s'explori hagi estat creat del seu estat pare amb l'operador en qüestió.

Observacions:

Cal escollir, per analitzar l'ús dels operadors, un dels mètodes de solució inicial desenvolupats. És raonable considerar, tot i això, que les ràtios de crides entre operadors serien similars entre els dos mètodes de solució inicials.

Cal fixar també una de les dues heurístiques. Considerarem la primera en ser la més senzilla. Cal tenir en compte que, en no considerar aquesta el nombre total de cotxes, l'operador desassignar conductor pot resultar penalitzat, usant-se menys cops que si utilitzessim la segona heurística que sí que ho considera.

Fixem també la mida del problema a 200 usuaris i 100 conductors.

Plantejament:

Quantificarem, per aquest experiment, el nombre d'usos totals que té cada un dels nostres operadors, per posteriorment analitzar-ho.

Hipòtesis:

Amb els tres operadors escollits, el primer que es fàcil observar és que desassignar conductor (*remove*) s'hauria d'usar tants cops com la diferència de cotxes entre la solució inicial i la final. Intuïtivament, també, l'operador moure passatgers (*move*) serà el que es cridarà més cops en ser el que ofereix una major versatilitat de canvis.

Mètode:

1. S’utilitzarà el mètode 1 anteriorment descrit per generar la solució inicial
2. Es resoldrà el problema 10 cops amb l’algorisme de Hill Climbing, cada cop amb una llavor diferent per al generador de nombres aleatoris.
3. Posteriorment, considerarem la mitjana entre l'ús de cadascun dels operadors en aquests 10 problemes considerats.

Resultats:

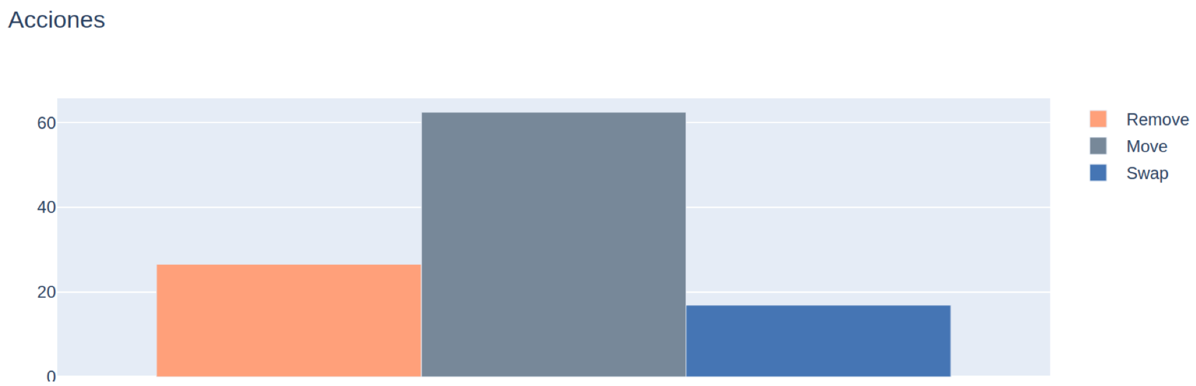

Figura 1: Número d’usos de cada operador per la solució inicial amb el mètode 1

Conclusions:

Com haviem teoritzat, l’operador *move* és el més cridat doncs és el que explora més l’espai de solucions i permet millores més substancials. L’operador *remove* depèn fortament del nombre de conductors i usuaris del problema, no es crida tant com l’operador *move* però aquest és essencial per a poder millorar les solucions del problema (doncs també volem minimitzar quantitat de conductors), especialment si considerem la segona heurística. Finalment, l’operador *swap* és el menys cridat dels tres, aquest operador introdueix poca ramificació i està subjecte a prous restriccions, per tant no explora gaire l’espai de solucions. Tot i així, degut a la seva baixa complexitat val la pena seguir tenint-lo en compte.

Experiment 2

L'objectiu d'aquest segon experiment és comparar les dues estratègies de solució inicial. Per fer-ho, ens fixarem tant en la qualitat del resultat al que s'arriba a partir de cada una com en el temps que es tarda en arribar-hi, usant l'algorisme de Hill Climbing.

Observacions:

Cal fixar una heurística per l'experiment. Anàlogament a l'apartat anterior, utilitzarem la primera, més senzilla.

Fixem també la mida del problema a 200 usuaris i 100 conductors.

Plantejament:

Mesurarem el temps tardat per resoldre el problema utilitzant Hill Climbing (incloent el temps de càlcul de la solució inicial) per veure quin mètode ens dóna millor eficiència.

També mesurarem la distància total recorreguda de les solucions i el nombre de conductors, per poder comparar la qualitat de les solucions obtingudes.

Hipòtesis:

Per com han estat construïdes les estratègies de solució inicial, semblaria que ambdues haurien de portar, amb el posterior ús dels operadors del problema, a solucions similars en distància i nombre de cotxes.

En quant a temps, la primera solució inicial sembla més ràpida de calcular que la segona, així que podem preveure un temps menor en aquest cas.

Mètode:

1. Es resoldrà el problema 10 cops amb l'algorisme de Hill Climbing per cada solució inicial, cada cop amb una llavor diferent per al generador de nombres aleatoris.
2. Posteriorment, considerarem la mitjana entre les mètriques que volem obtenir en aquests 10 problemes considerats.

Resultats:

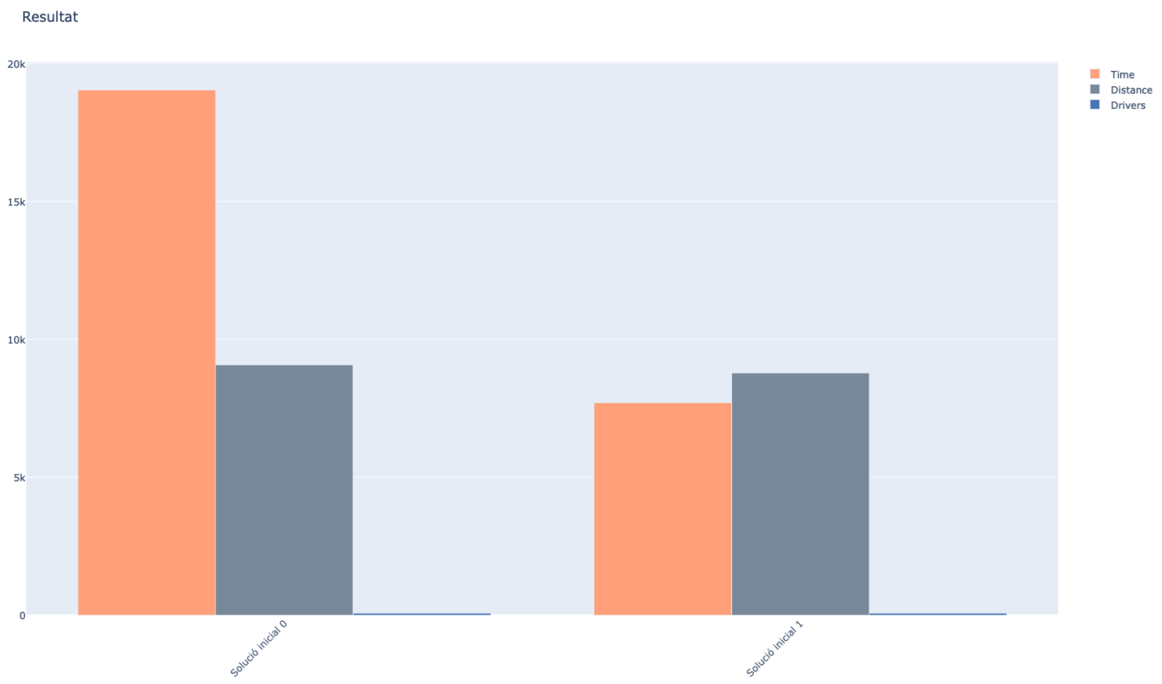

Figura 2: Temps, distància total i nombre de cotxes obtinguts per solució inicial en l'experiment.

Conclusions:

En quant a la distància total (9077.1 per el primer mètode i 8774.7 per al segon) i nombre de cotxes (76.6 al primer i 76.1 al segon) ambdues semblen apuntar a que la segona solució inicial proporciona millors resultats del problema, tot i ser una diferència petita.

En quant a temps trobem que la diferència sí que és considerable (19 segons amb el primer mètode i 7.7 amb el segon). Així, al contrari del que havíem suposat en les hipòtesis, el segon mètode ens dona menors temps de resolució per al problema.

Això es deu, probablement, a que la solució inicial construïda per aquest mètode convergeix més ràpidament cap a la solució que la del primer, fent que tot i que pugui arribar a ser més costosa de calcular inicialment l'algorisme de Hill Climbing hagi d'efectuar menys passes i calcular menys estats per arribar a la solució final.

Experiment 3

L'objectiu del tercer experiment és trobar el conjunt de paràmetres òptim per usar a l'algorisme de Simulated Annealing. Amb aquest objectiu, realitzarem un seguit de proves amb cada conjunt de paràmetres i compararem els resultats obtinguts.

Observacions:

Fixarem la primera heurística utilitzada i la segona solució inicial, que ja hem vist que proporciona millors resultats.

També es mantindrà fixa la mida del problema a 200 usuaris i 100 conductors.

Plantejament:

Compararem les distàncies obtingudes per als següents possibles valors dels paràmetres:

k $\in$ {1, 5, 25, 125, 625, 3125}

stiter $\in$ {10, 20, 30, 40, 50, 60, 70, 80, 90}

$\lambda \in \{1, 0.1, 0.01, 0.001, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}\}$

Hipòtesis:

En un principi esperem millors resultats amb un nombre elevat d'iteracions entre canvis de temperatura (stiter) doncs un nombre massa petit no permetrà millorar molt les solucions (els canvis que es fan a cada iteració són petits). Pel que fa als altres paràmetres, són molt dependents del problema i és difícil fer-se una idea de com afecten a aquest cas en particular.

Mètode:

1. Es resoldrà el problema 10 cops amb l'algorisme de Simulated Annealing per cada combinació dels possibles paràmetres, cada cop amb una llavor diferent per al generador de nombres aleatoris.
3. Posteriorment, considerarem la mitjana entre les mètriques que volem obtenir en aquests 10 problemes considerats.

Resultats:

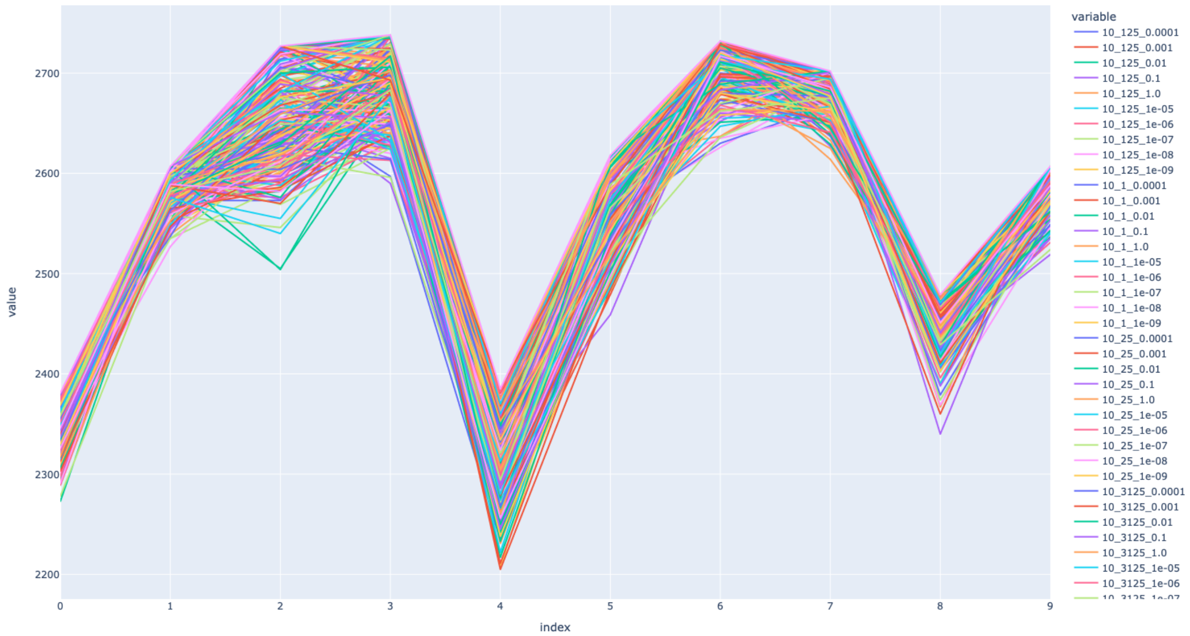

Figura 3: Distància obtinguda amb cada configuració de paràmetres en funció de la seed (nº d'execució)

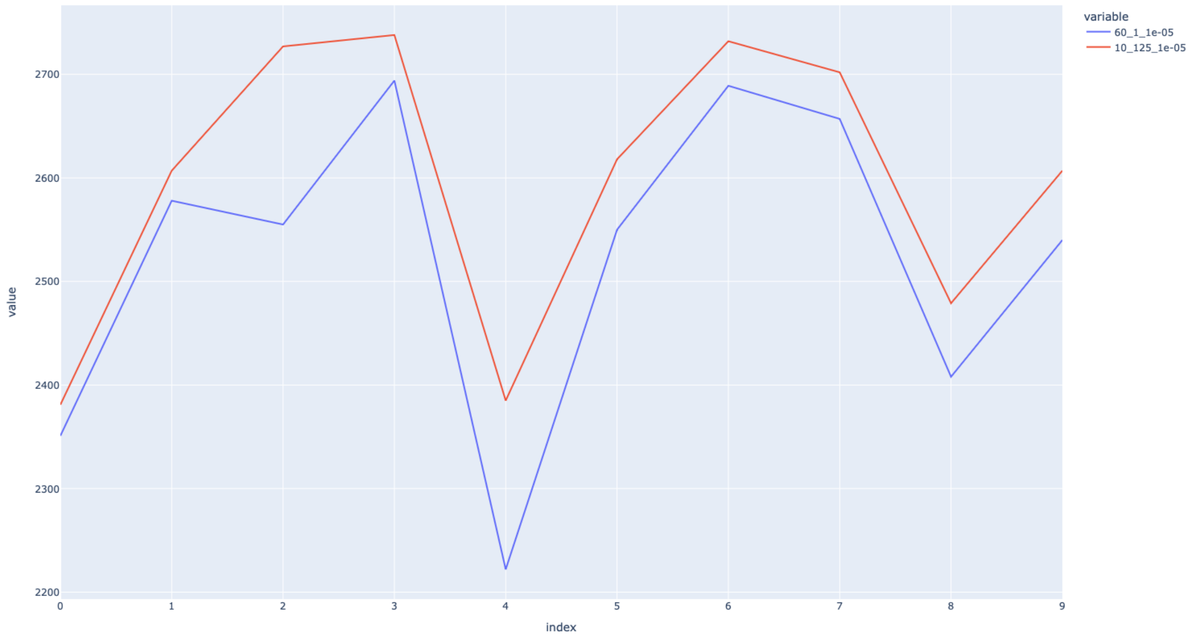

Figura 4: Conjunt de paràmetres amb mitjana més alta i més baixa

Conclusions:

Per tant, de la segona gràfica podem deduir que el conjunt de paràmetres que en mitjana obté distàncies més baixes és:

stiter = 60 k = 1 $\lambda = 10^{-5}$

Usarem aquests paràmetres d'ara en endavant quan haguem de fer servir Simulated Annealing.

Experiment 4

En aquest experiment volem analitzar la complexitat de la resolució del problema en variar la mida del mateix; fonamentalment el nombre d'usuaris.

Observacions:

D'acord amb els resultats obtinguts prèviament sobre les solucions inicials, fixarem el segon mètode en ser més eficient temporalment i proporcionar lleugerament millors resultats.

Fixarem també la primera heurística per simplicitat i coherència amb els problemes anteriors.

Considerarem, per cada mida d'usuaris que provarem en la resolució del problema, que el nombre de conductors és sempre la meitat dels usuaris totals.

Plantejament:

Resoldrem el problema amb mides d'usuari entre 200 i 500 i mesurarem el temps que es tarda de mitjana en resoldre'ls.

Hipòtesis:

En augmentar la mida, el temps augmentarà de forma no lineal amb aquesta, amb un factor al menys cúbic de complexitat.

Mètode:

1. Es resoldrà el problema 10 cops amb l'algorisme de Hill Climbing per mides 200, 300, 400 i 500, cada cop amb una llavor diferent per al generador de nombres aleatoris.
2. Posteriorment, considerarem la mitjana entre les mètriques que volem obtenir en aquests 10 problemes considerats.

Resultats:

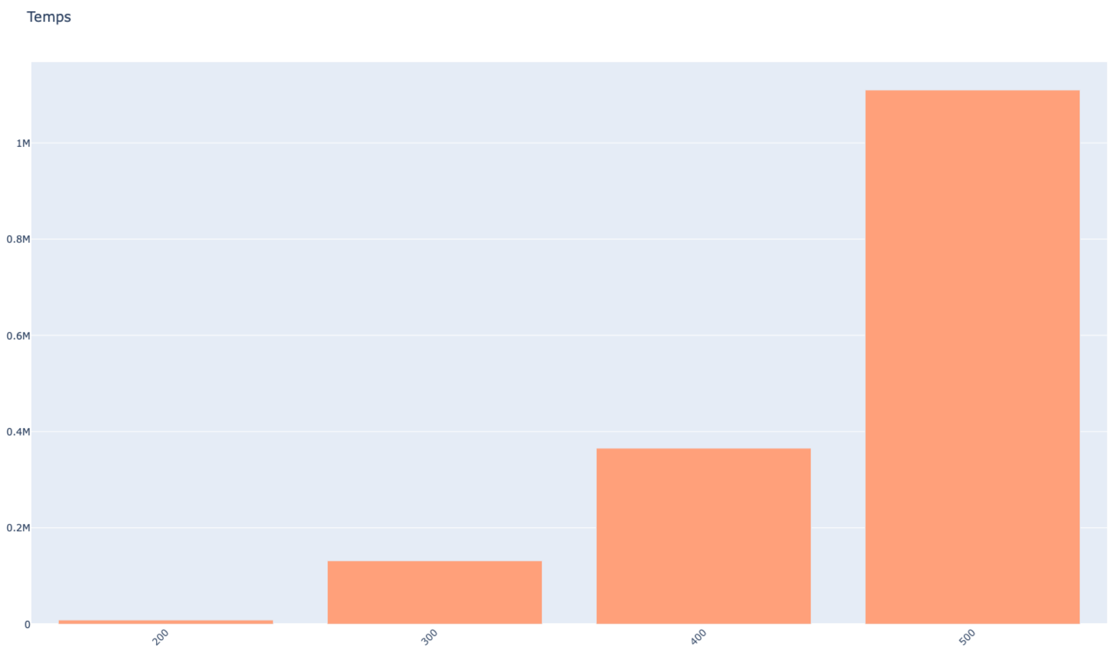

Figura 5: Temps de resolució per nombre d'usuaris obtingut en l'experiment 4

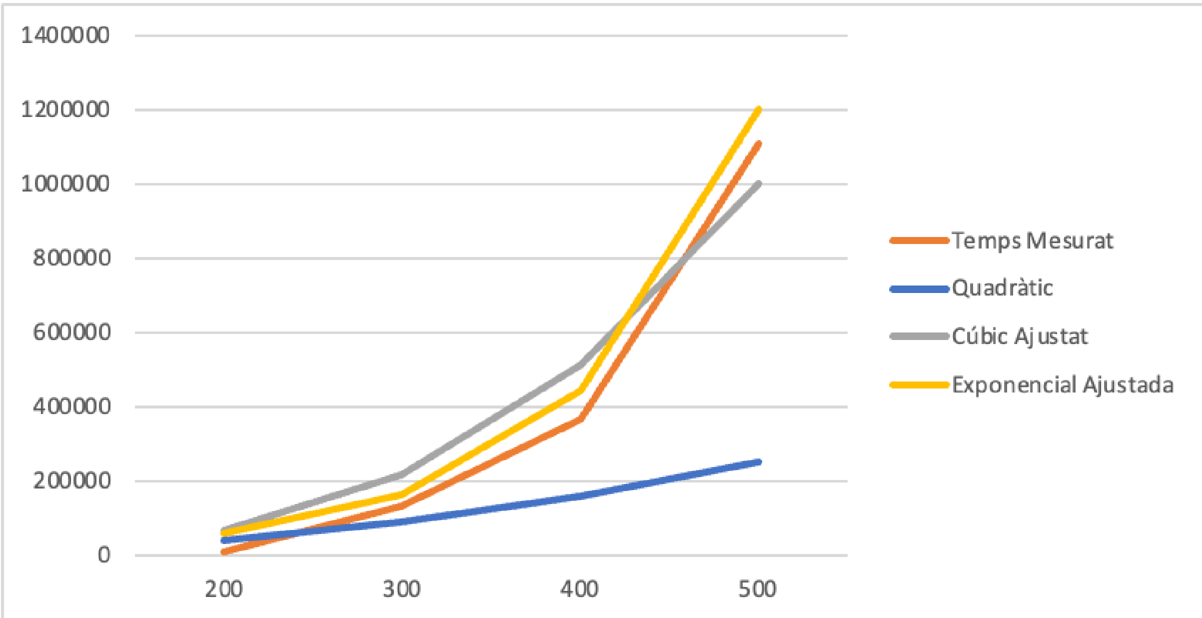

Figura 6: Temps de resolució per nombre d'usuaris obtingut en l'experiment 4 comparat amb altres gràfiques de funcions conegudes (ajustades referint-se a que han estat multiplicades per constants per adaptar-les a l'escala de la figura)

Conclusions:

És fàcil observar de la gràfica que l'augment en temps no correspon a una relació lineal. De fet, la forma de la gràfica sembla induir que fins i tot, enlloc d'una complexitat polinòmica, ens poguéssim estar trobant davant d'una complexitat exponencial.

Experiment 5

Per aquest experiment, volem comparar les dues heurístiques. De forma similar a com hem comparat anteriorment les solucions inicials en l'experiment 2 ens fixarem tant en la qualitat del resultat al que s'arriba a partir de cada una com en el temps que es tarda en arribar-hi, usant l'algorisme de Hill Climbing.

Observacions:

Fixem com a solució inicial la segona, en ser la que millors resultats ens dona en l'experiment 3.

Fixem també la mida del problema a 200 usuaris i 100 conductors.

Plantejament:

Mesurarem el temps tardat per resoldre el problema utilitzant Hill Climbing (incloent el temps de càlcul de la solució inicial) per veure quin mètode ens dóna millor eficiència.

També mesurarem la distancia total recorreguda de les solucions i el nombre de conductors, per poder comparar la qualitat de les solucions obtingudes.

Hipòtesis:

Tenint en compte que la primera heurística només considera distàncies totals i no nombre de cotxes, seria raonable suposar que les resolucions que aportarà del problema tindran un major nombre de cotxes que per la segona.

Respecte el temps, és possible que el de la segona heurística sigui més elevat ja que necessita minimitzar més *variables,* la distància i el número de cotxes, per tant, és probable que trigui més a aconseguir un mínim en ambdues components.

Mètode:

1. Es resoldrà el problema 10 cops amb l'algorisme de Hill Climbing per cada heurística, cada cop amb una llavor diferent per al generador de nombres aleatoris.
2. Posteriorment, considerarem la mitjana entre les mètriques que volem obtenir en aquests 10 problemes considerats.

Resultats:

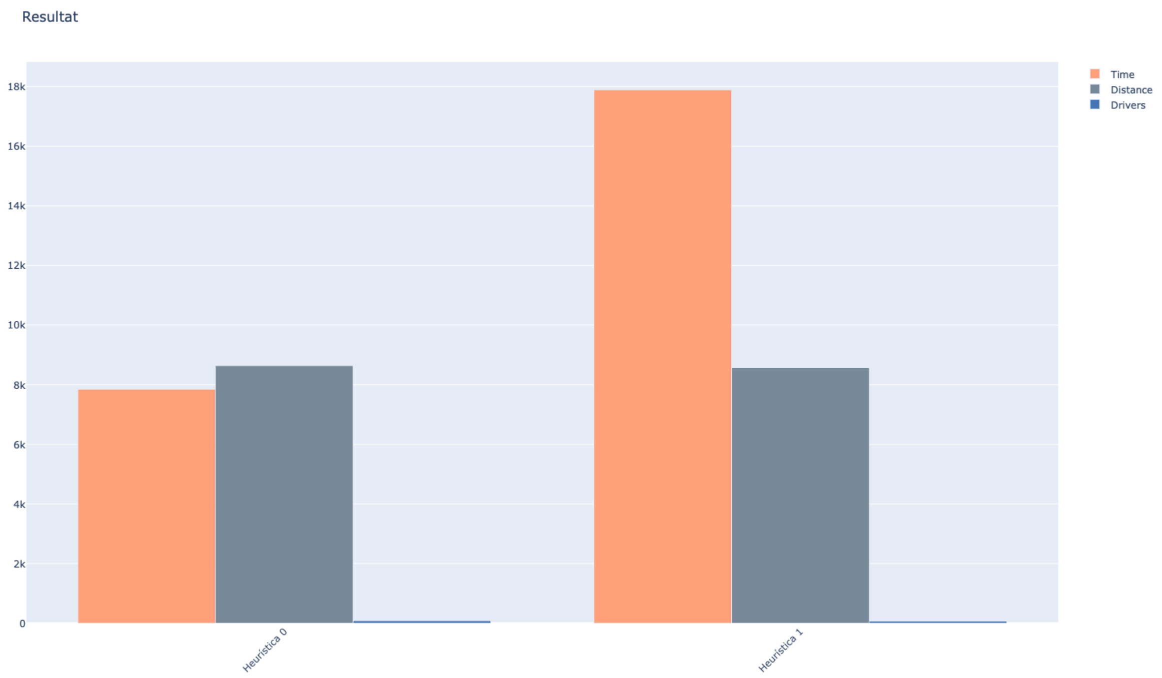

Figura 7: Temps, distància total i nombre de cotxes obtinguts per heurística en l'experiment.

Conclusions:

El primer que s'observa en els resultats és que la resolució dels problemes amb la segona heurística requereix de molt més temps que la primera, pràcticament el doble.

En quant a les solucions obtingudes, la diferència en distància total es mínima (8637.2 per la primera i 8582.4 per la segona) mentre que, tal com era d'esperar, es nota una diferència més apreciable en nombre de cotxes (77 per la primera i 61.4 per la segona).

Així, la segona heurística sembla proporcionar millors resultats però a canvi de requerir una quantitat de temps considerablement més elevada que la primera per obtenir-los.

Experiment 6

Per aquest experiment, volem comparar de nou les dues heurístiques, anàlogament a com ho havíem fet en l'experiment anterior pero ara per l'algorisme de Simulated Annealing. Degut al diferent funcionament d'aquest algorisme, és raonable esperar resultats diferents, especialment en el relatiu al temps.

Observacions:

Fixem com a solució inicial la segona, en ser la que millors resultats ens dona en l'experiment 3.

Fixem també la mida del problema a 200 usuaris i 100 conductors.

Per els paràmetres del Simulated Annealing, fixarem els òptims obtinguts en l'experiment 3.

Plantejament:

Mesurarem el temps tardat per resoldre el problema utilitzant Simulated Annealing (incloent el temps de càlcul de la solució inicial) per veure quin mètode ens dóna millor eficiència.

També mesurarem la distancia total recorreguda de les solucions i el nombre de conductors, per poder comparar la qualitat de les solucions obtingudes.

Hipòtesis:

Tot i el diferent funcionament dels algorismes que fa que puguem esperar diferents resultats en termes absoluts, caldria esperar un comportament relatiu similar a l'obtingut en l'experiment anterior: amb la segona heurística donant millors resultats en quant a nombre de cotxes tot i que sigui a costa d'un major cost en temps.

Mètode:

1. Es resoldrà el problema 10 cops amb l'algorisme de Simulated Annealing per cada heurística, cada cop amb una llavor diferent per al generador de nombres aleatoris.
2. Posteriorment, considerarem la mitjana entre les mètriques que volem obtenir en aquests 10 problemes considerats.

Resultats:

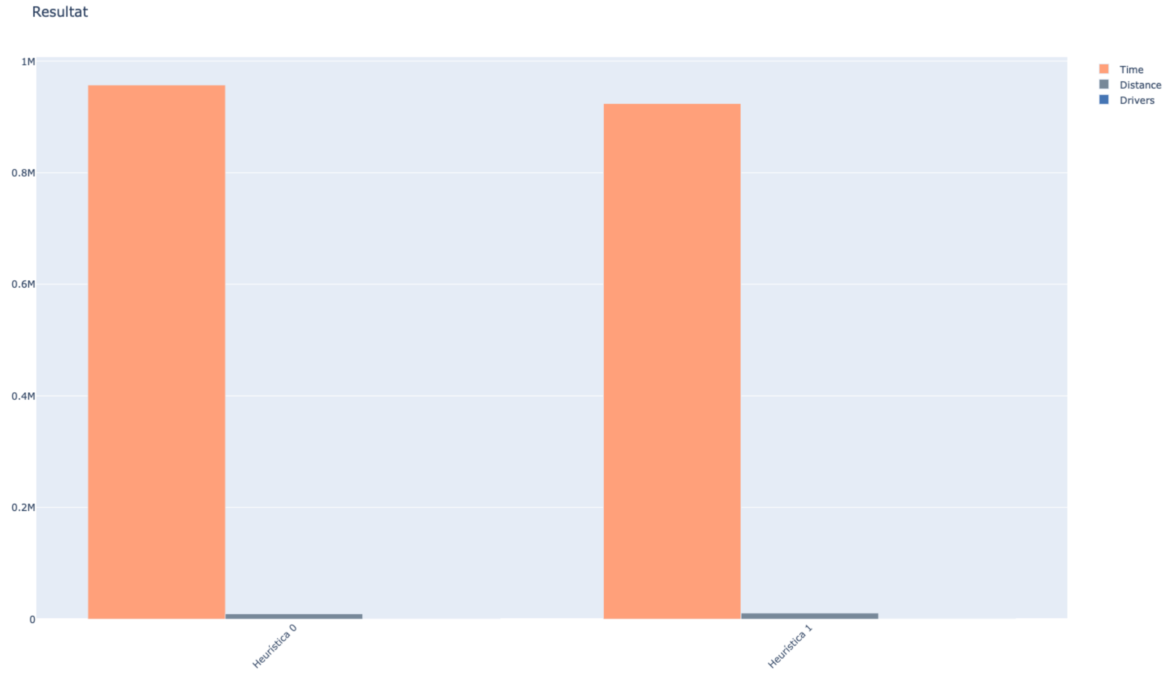

Figura 8: Temps, distància total i nombre de cotxes obtinguts per heurística en l'experiment.

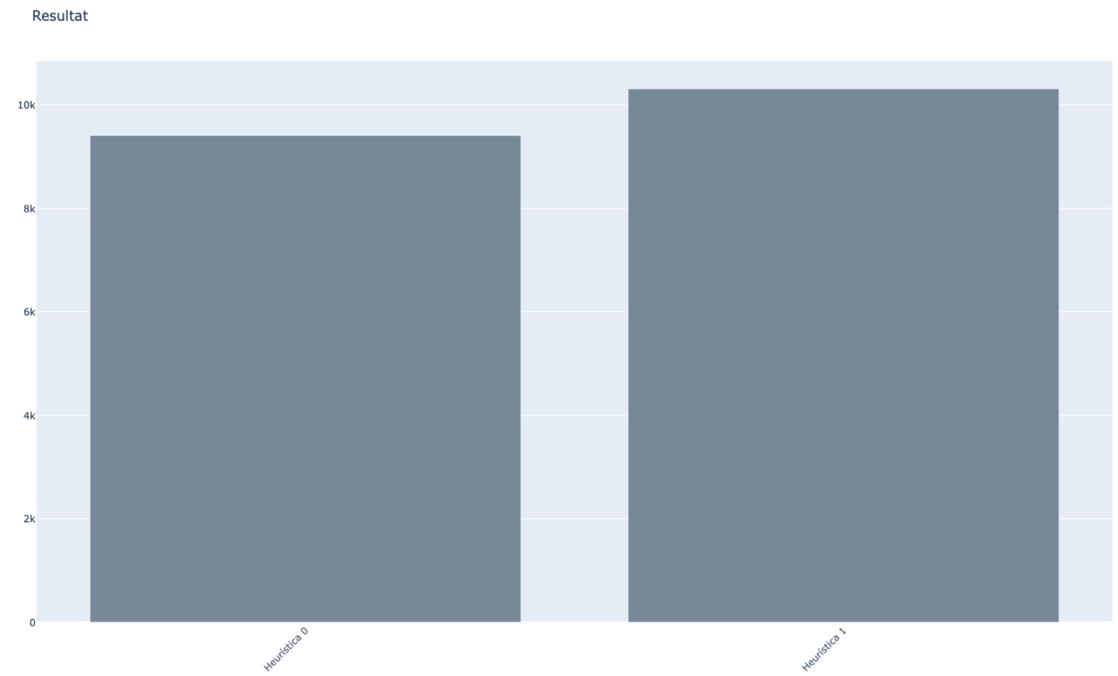

Figura 9: Distància total obtinguda per heurística en l'experiment.

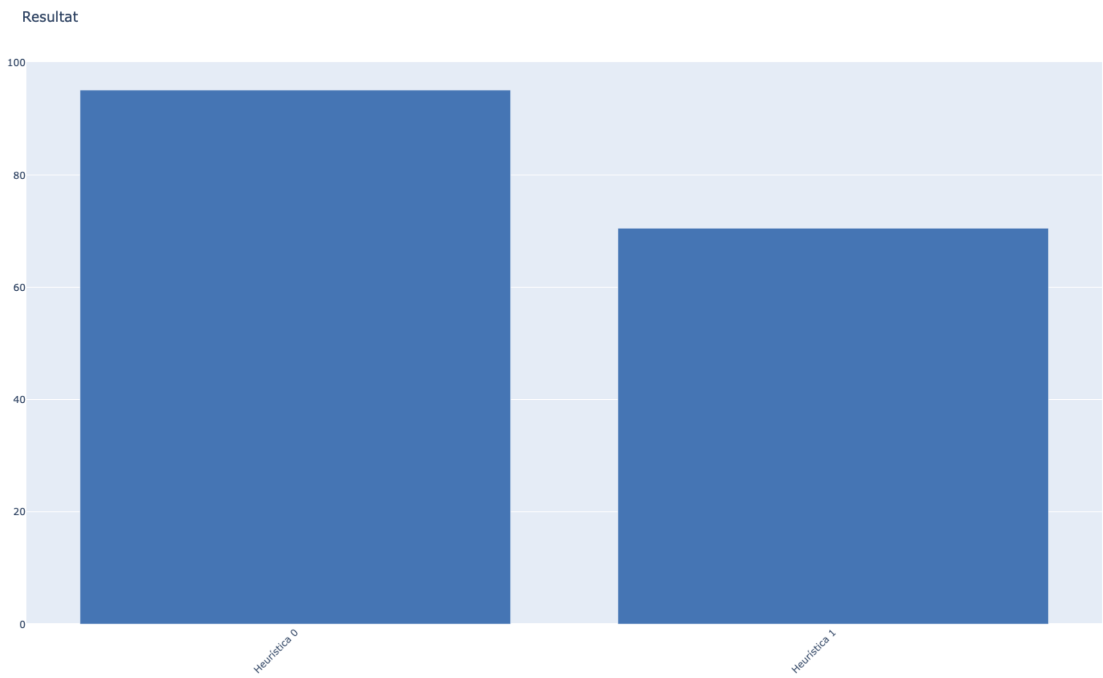

Figura 10: Nombre de cotxes obtinguts per heurística en l'experiment.

Conclusions:

Observem que, tal com preveiem, els costos temporals han augmentat en un ordre de magnitud respecte al mateix experiment utilitzant Hill Climbing. Tot i així, observem també que amb les dues heurístiques es requereixen temps similars, sent menor en aquest cas el de la segona heurística de forma oposada amb el que passa en l'experiment anterior.

Sí que es manté la similaritat en la qualitat de la solució obtinguda, en què la segona heurística mostra un molt millor resultat en quant a nombre de cotxes i un resultat similar en quant a distància total recorreguda. Cal remarcar també que, en aquest cas, de nou al contrari que en l'experiment anterior, la segona heurística dona resultats lleugerament pitjors en quant a distància total mentre que en l'experiment anterior en donava de lleugerament millors.

Experiment 7

En aquest darrer experiment volem veure els efectes de variar la ràtio de conductors d'entre el total d'usuaris, que fins ara ha estat sempre la meitat d'aquests. Avaluarem l'impacte de canviar aquesta ràtio tant en la qualitat de les solucions com en el temps invertit per obtenir-les.

Observacions:

Fixem com a solució inicial la segona, en ser la que millors resultats ens dona en l'experiment 3.

Fixem també la mida del problema a 200 usuaris.

En tenir resultats prou diferents per cada heurística, farem l'experiment independentment per cada una de les heurístiques.

Plantejament:

Mesurarem el temps tardat per resoldre el problema utilitzant Hill Climbing (incloent el temps de càlcul de la solució inicial) per veure quin mètode ens dóna millor eficiència.

També mesurarem la distancia total recorreguda de les solucions i el nombre de conductors, per poder comparar la qualitat de les solucions obtingudes.

Hipòtesis:

Es raonable suposar que, a més nombre de conductors, més costós serà calcular la solució del problema en quant a temps, en créixer l'espai de cerca i els possibles estats a calcular.

També, degut a aquest major nombre d'estats possibles, sembla raonable que a més conductors millor pugui ser la qualitat de la solució en quant a distancia total, tot i que en tenir més conductors disponibles podem preveure també que a més conductors més cotxes s'usaran en la solució final.

En quant al nombre de cotxes en particular, aquest creixement, en tenir més conductors disponibles, hauria de ser més moderat usant la segona heurística en primar aquesta una reducció en nombre de cotxes.

Mètode:

1. Es resoldrà el problema 10 cops amb l'algorisme de Hill Climbing per cada nombre de conductors, que variarà en 25 en 25, de 50 a 200, cada cop amb una llavor diferent per al generador de nombres aleatoris.

2. Posteriorment, considerarem la mitjana entre les mètriques que volem obtenir en aquests 10 problemes considerats.

Resultats:

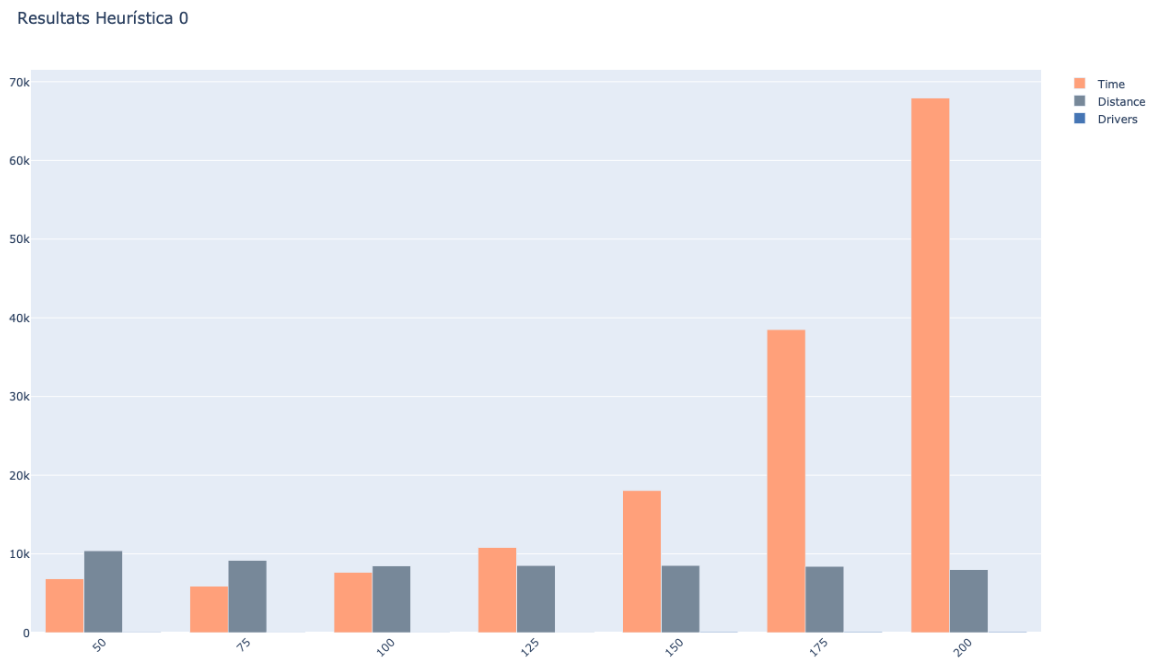

Figura 11: Temps, distància total i nombre de cotxes obtinguts per nombre de conductors en l'experiment usant la primera heurística.

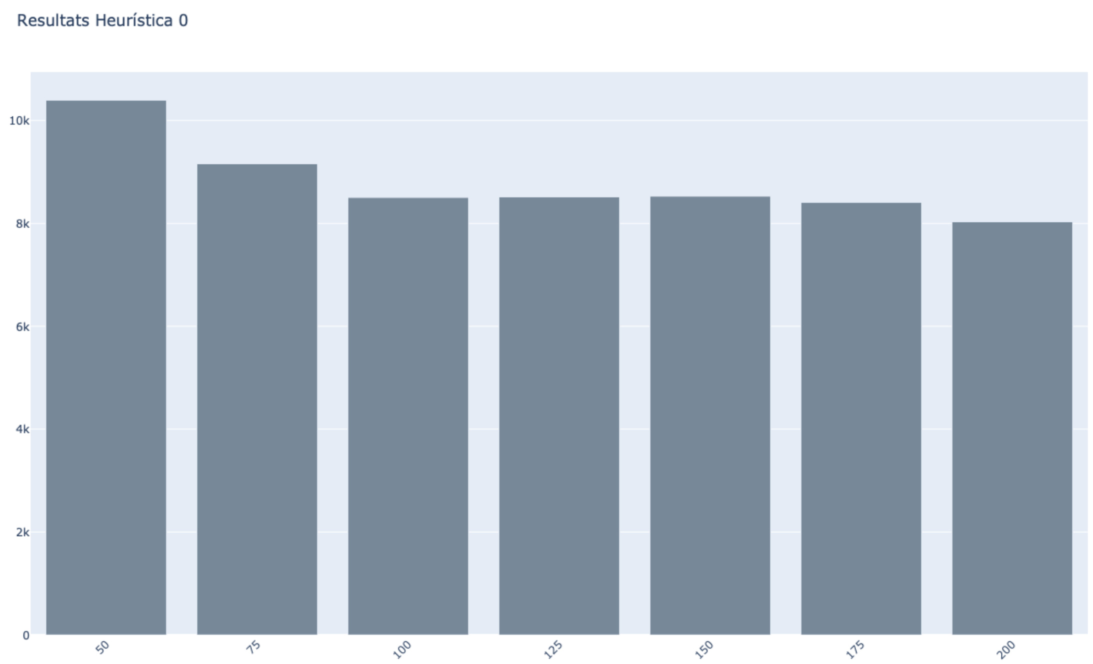

Figura 12: Distància total obtinguda per nombre de conductors en l'experiment usant la primera heurística.

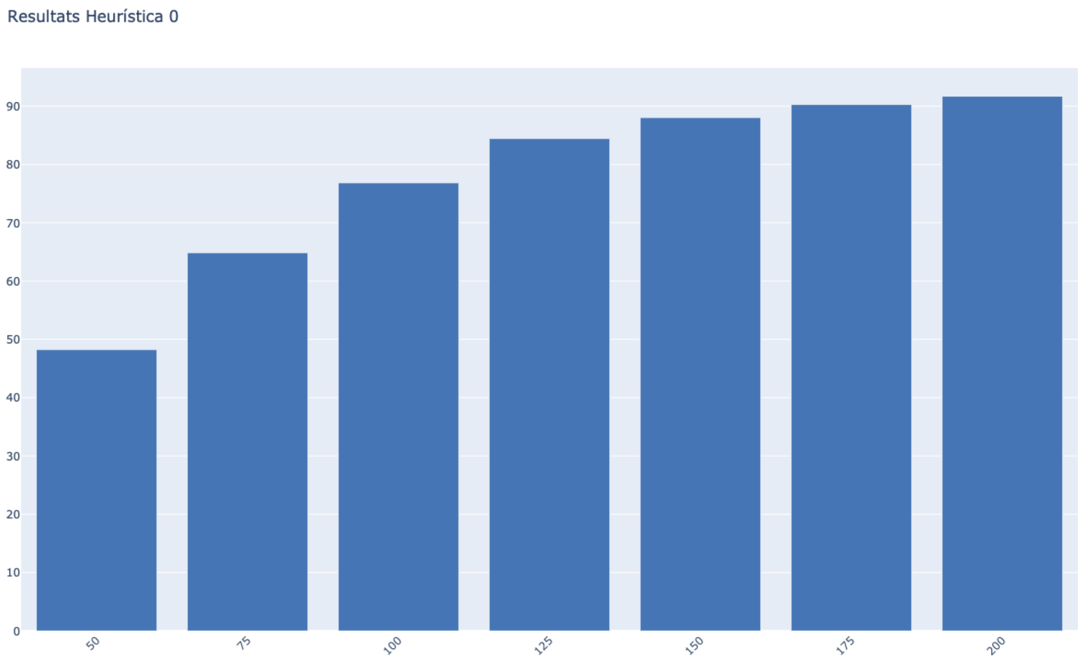

Figura 13: Nombre de cotxes obtinguts per nombre de conductors en l'experiment usant la primera heurística.

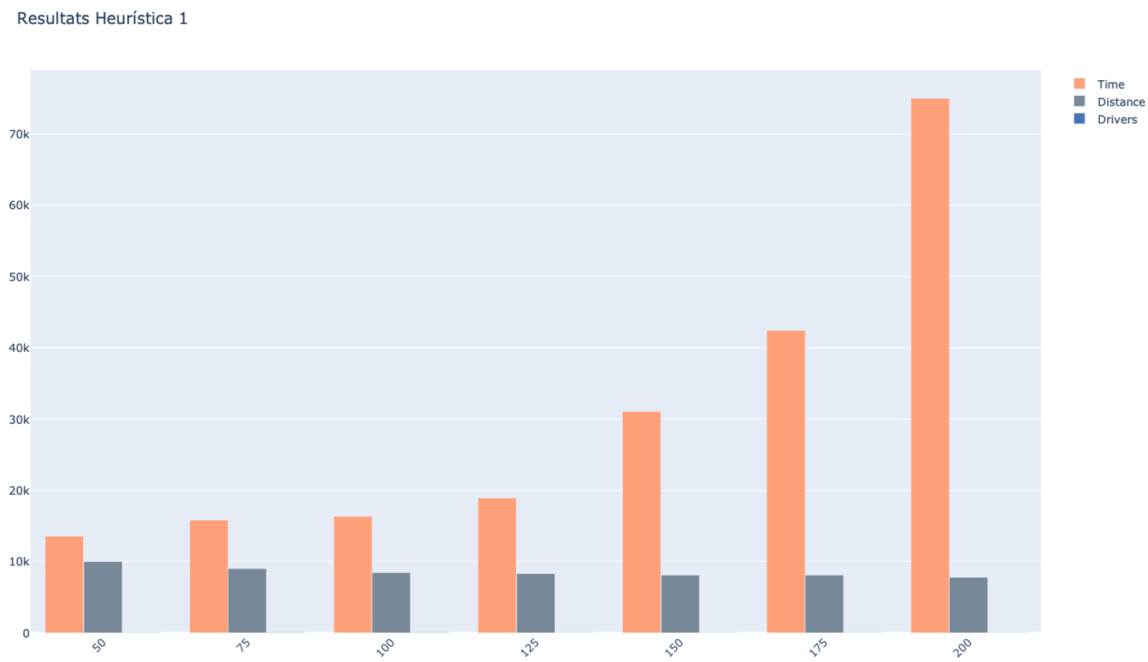

Figura 14: Temps, distància total i nombre de cotxes obtinguts per nombre de conductors en l'experiment usant la segona heurística.

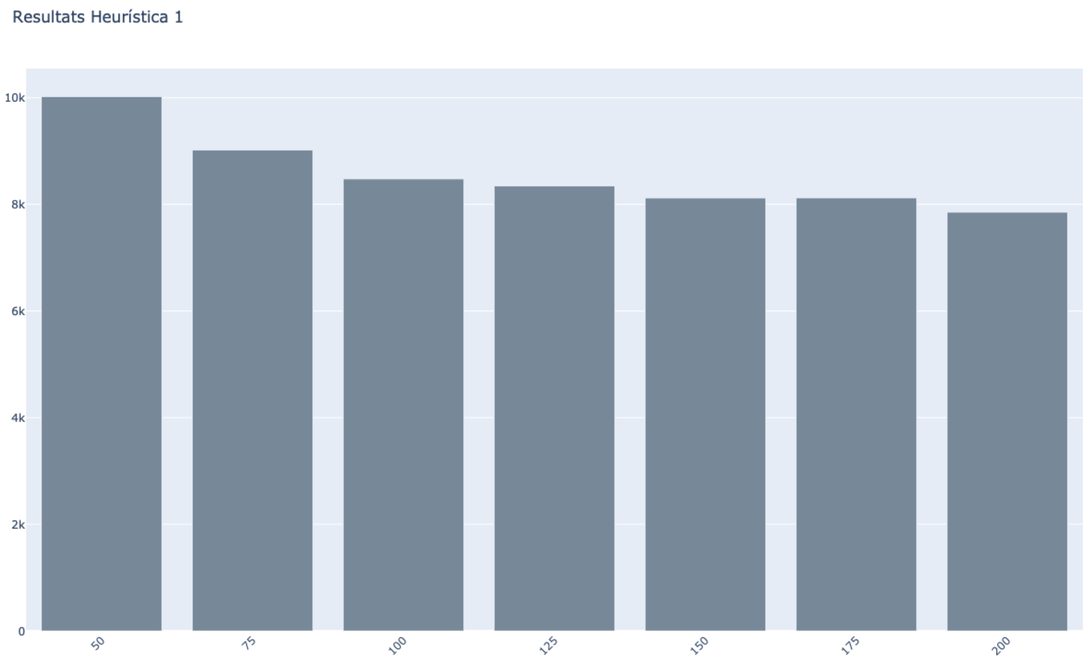

Figura 15: Distància total obtinguda per nombre de conductors en l'experiment usant la segona heurística.

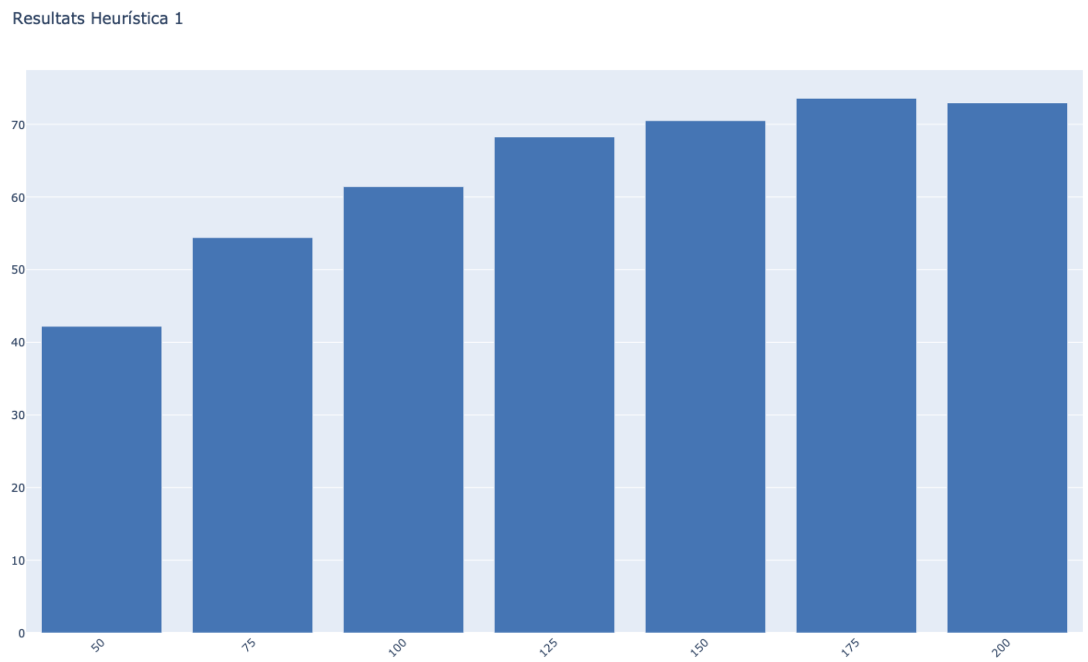

Figura 16: Nombre de cotxes obtinguts per nombre de conductors en l'experiment usant la segona heurística.

Conclusions:

Tal i com podíem preveure, el temps requerit per arribar a les solucions creix a mesura que augmenta el nombre de conductors, amb un cost que sembla similar al que havíem obtingut en l’experiment 4. En termes absoluts, la segona heurística segueix tenint un cost temporal major a la primera, tot i que la diferència no es tan accentuada com havíem vist en l’experiment 6.

També com havíem previst, la distància total és més baixa en augmentar el nombre de conductors alhora que el nombre de cotxes creix usant ambdues heurístiques. Per la segona, però, el nombre de cotxes és menor es valors absoluts, tot i que segueix tenint el mateix patró creixent.

7. Conclusions

En conclusió, amb aquesta pràctica de cerca local, hem cobert els objectius que se'ns proposaven inicialment.

En primer lloc, hem après com funciona AIMA i com es poden utilitzar les seves classes per solucionar problemes de la vida real amb intel·ligència artificial, com és el cas del problema de la compartició de cotxes que se'ns proposava en aquesta pràctica.

En segon lloc, hem raonat sobre tots els elements necessaris per poder plantejar l'espai de solucions del problema, trobar-ne una d'incial amb facilitat, determinar i argumentar quins són potencialment els millors operadors de transformació de solucions a fer servir i quines heurístiques són més coherents i útils.

Finalment, hem ajustat tots els paràmetres rellevants per obtenir una bona solució mitjançant l'execució del problema amb dos algoritmes diferents, Hill Climbing i Simulated Annealing (explorant també quins són els millors paràmetres pel darrer), dues solucions inicials, dues heurístiques i proporcions d'usuaris i conductors diferents, presentant tots els resultats rellevants en l'apartat específic de conclusions de cada experiment.

8. Treball d'Innovació: Traductor NLLB-200

El tema escollit pel treball d'innovació és el Traductor NNLB-200 creat per Meta AI (NNLB significa No Language Left Behind). És el primer traductor capaç de traduir amb una qualitat sense precedents a 200 idiomes diferents, superant les tecnologies que hi havia fins ara en un 44% de mitjana.

Repartiment de tasques

Per poder entendre tots bé com funciona el traductor, estem fent conjuntament la descripció de les tècniques de IA i la descripció de l'utilització d'aquestes tècniques (i també la introducció i conclusions). Els apartats restants ens els hem dividit: *Innovació* pel Rubén Aciego, *Impacte Empresarial* per Francesc Pifarré i *Impacte Social* per Mariona Jaramillo.

Referències

Facebook. 200 idiomas con un solo modelo de IA: todo un logro en la traducción automática de alta calidad. https://ai.facebook.com/blog/nllb-200-high-quality-machine-translation/es-es/ [Online; accedit 6 març 2023]

Facebook Research. Facebook AI Research Sequence-to-Sequence Toolkit written in Python. https://github.com/facebookresearch/fairseq/tree/nllb [Online; accedit 6 març 2023]

Facebook. About No Language Left Behind. https://ai.facebook.com/research/no-language-left-behind/ [Online; accedit 6 març 2023]

NLLB Team et al, No Language Left Behind: Scaling Human-Centered Machine Translation, Arxiv, 2022. https://scontent-bcn1-1.xx.fbcdn.net/v/t39.8562-6/292295068_402295381932691_8903854229220968087_n.pdf?_nc_cat=102&ccb=1-7&_nc_sid=ad8a9d&_nc_ohc=EH5cRPYCMZgAX8sK6dl&_nc_ht=scontent-bcn1-1.xx&oh=00_AfAHKlM0lbrr6xnk_Da5OV6idww4E4sOYgwT-qZnJTutoQ&oe=642C8753 [Online; accedit 1 abril 2023]

Dificultats

La major dificultat d'aquest treball és que, al ser el producte tan nou, només podem recórrer a la documentació oficial, és a dir, un article científic dens i tècnic i a una web de github.