

Intel·ligència artificial

Pràctica 1

David Guillen Fandos
Jordi Mundet Mas

Índex

0.	Introducció.....	1
1.	Identificació del problema.....	2
2.	Representació de l'estat.....	3
3.	Operadors i espai de solucions.....	5
4.	Generació de l'estat inicial.....	10
5.	Disseny i anàlisi de la funció heurística.....	15
6.	Experimentació.....	20
I.	Annexos.....	26

0. Introducció

La pràctica que es presenta està formada per un programa java i aquesta memòria. El disposa d'una interfície gràfica per tal de poder utilitzar-lo amb facilitat i poder fer experiments de forma senzilla. L'objectiu d'aquesta introducció és descriure aquest programa a la vegada que s'explica com està organitzada la memòria.

La interfície permet ajustar tots els paràmetres del problema així com generar paràmetres de forma aleatòria. Alguns d'aquests es poden lligar a distribucions de probabilitat, cosa que està disponible al botó “Prob”.

Un cop s'ajusten i es generen els paràmetres del problema s'ha d'ajustar els paràmetres de la resolució. En general és bastant intuïtiu i no requereix gaire explicació. Si es desitja veure la distribució final (estat final) de la solució cal fer click a “Resultats”. A continuació es mostra una captura de la interfície.

The screenshot shows the 'Problema - Heuristics' Java GUI. The interface is organized into several sections:

- Paràmetres del problema:** Includes sliders for 'Nº camions de 500kq' (23), 'Nº camions de 1000kq' (13), and 'Nº camions de 2000kq' (24). Below these is a table of 'Comandes (centre/hora/pes)' with columns for Centre, Hora, and Pes. The table lists 10 commands for 6 centres.
- Paràmetres de la resolució:** Includes sliders for 'N. iterations' (23733), 'Stiter' (100), 'K' (20), and 'Lambda (E-3)' (5). It also has radio buttons for 'Maximitzar el benefici', 'Minimitzar retard', 'Operadors 1', and 'Operadors 2'. There are also options for 'Inicialització' (desequilibrada, equilibrada, aleatòria, intel·ligent).
- Tables:** Two tables show data for 'Centre 0' through 'Centre 5'. The top table has 10 rows of data, and the bottom table has 10 rows of data.
- Informació sobre la solució:** Displays statistics for 'Hilclimbing' and 'Simul. Annealing'. For Hilclimbing: Beneficiis: 90700, Temps: 4108ms, Hores retard: 987, Pesos (total / no entregat / entregat / lliure): 79700 / 8100 / 71600 / 900. For Simul. Annealing: Beneficiis: 92410, Temps: 1837ms, Hores retard: 906, Num estats visitats: 23734, Pesos (total / no entregat / entregat / lliure): 79700 / 7700 / 72000 / 500.

En el que respecta a la memòria aquesta té una estructura lleugerament diferent a la especificada a l'enunciat. Alguns experiments estan inclosos a la part d'explicació i justificació del programa per tal de poder-los lligar millor amb el que s'està explicant (i així ho corroborarem experimentalment).

La resta d'experiments que no tenen cap part teòrica directament associada però que es demanaven explícitament a l'enunciat es troben al final. A més s'ha realitzat experiments de més per tal de provar coses que es consideren rellevants i que no estan incloses a l'enunciat. Alguns d'aquests estan inclosos en d'altres de manera que l'enunciat és un subconjunt dels experiments realitzats.

1. Identificació del problema

El problema que se'ns planteja és més semblant a un problema de distribució d'horaris que no pas a un problema d'entregues del tipus viatjant, degut al fet que no cal tenir en compte distàncies, velocitat ni cap tipus de desplaçament. Podríem resumir el problema en un problema de distribuir tasques en un horari sabent quines són les hores límit de cada tasca, el benefici que ens aporten i la penalització per retard associada.

La "gràcia" del problema és la distribució dels camions, més que no pas la distribució de les entregues (ja que això es pot fer amb algorismes del tipus *greedy* amb bons resultats). Si suposem que els camions ja han estat assignats a una hora i un centre concrets aleshores el problema es limita a trobar la distribució d'entregues que maximitza el benefici. Així doncs podem dir que són dos problemes simultanis: com distribuir els camions a cada hora i com distribuir els paquets en les hores i centres.

Aquest doble problema és precisament el que donarà més problemes i que serà una mica el maldecap de l'algorisme. Per una banda podem pensar que cal trobar distribucions de camions i, un cop trobades, repartir paquets. Però això és un error ja que una distribució de camions que, a priori, és molt bona ens pot dur a una distribució de paquets dolenta per molts motius (posar paquets pesants deixa espais lliures, potser més val sacrificar un centre en favor de la resta, etc.) i això és el que ens porta a dir que cal tractar els dos problemes simultàniament.

Finalment caldrà tenir en compte que no totes les comandes poden ser servides sempre, per aquest motiu cal tenir una llista de comandes que no seran servides. Una situació on no es serveix cap comanda és un estat correcte (tot i que poc productiu) i podria ben ser un estat inicial.

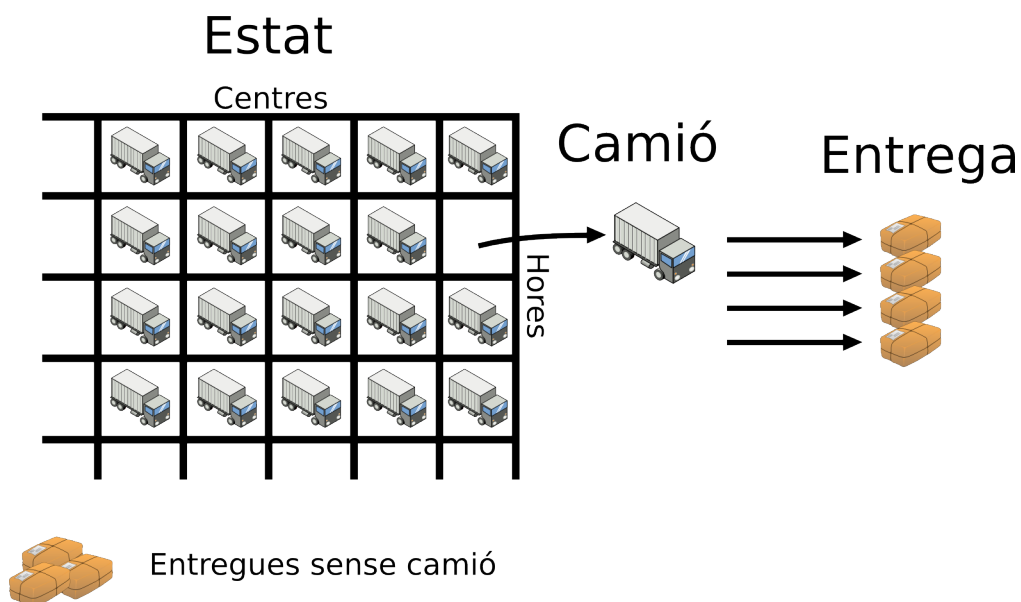
2. Representació de l'estat

Descripció d'un estat

Representarem amb les classes *Camio* i *Entrega* els dos elements principals del problema. L'entrega estarà formada per l'hora d'entrega, el pes i el centre al qual s'ha d'entregar. El camió tindrà una certa capacitat i un vector d'entregues que carrega. Afegim atributs com ara l'espai lliure al camió per tal d'accelerar el procés d'afegir i treure paquets o comprovar l'espai lliure que conté (no cal recalculer cada vegada).

L'estat doncs ens diu quin camió hi ha a cada hora del dia i a quin centre va. Això és representat per una taula (com un horari) on les files són les hores del dia (de 0 a 9) i les columnes cada un dels centres que tenim. A cada cel·la tenim un camió que, a la seva vegada i com ja hem dit, conté la llista d'entregues que realitza. A més a més l'estat té una llista d'entregues que no s'enviaran, ja que no totes hi caben de vegades.

Aquesta representació és senzilla i eficient en espai i, al fer ús de classes i referències, permet crear nous estats amb rapidesa i fer canvis de forma senzilla. Per intercanviar dos camions es pot fer canviant-ne les referències, així com també els paquets. A més es pot comprovar ràpidament l'espai lliure dels camions i si una entrega es pot encabir en un d'aquests. Al fer servir vectors i taules preservem la localitat espacial i temporal, d'aquesta manera tot el procés serà més ràpid.



Esquema on es representa l'estat del problema.

Descripció de l'espai de cerca

Podem considerar des del nostre punt de vista (la representació de l'espai de cerca triada) que aquest té 2 graus de llibertat. El primer seria totes les possibles reordenacions dels camions i la segona totes els possibles reordenacions d'entregues dins dels camions i a fora (a la llista de no entregades).

La grandària de l'espai de cerca és de forma aproximada (una cota superior) el nombre de possibles posicions de camions a l'horari pel nombre de posicions de les entregues dins dels camions d'un centre. A això cal afegir el nombre de combinacions d'entregues que estaran a fora (no hi caben). Òbviament serà menor per les restriccions d'espai, que no podem determinar a priori.

Siguin a i b el nombre de camions de dues capacitats diferents. El nombre de possibles ordenacions de camions és:

$$\binom{60}{b} \cdot \binom{60-a}{b}$$

Ara ve la part més complicada: calcular les possibles ordenacions d'entregues. Una cota superior és considerar que només hi cap una entrega per camió (pitjor cas) i que estan repartides pels centres equitativament. Per aquest cas:

$$\left(\binom{ne/6}{10} \cdot 10! \right)^6$$

Però aquest cas no té en compte que les entregues es distingeixen per la seva capacitat i hora d'entrega, pel que és una simple cota molt superior al pitjor cas.

Així que podem dir que l'espai de cerca té una cota superior donada per:

$$\binom{60}{b} \cdot \binom{60-a}{b} \cdot \left(\binom{ne/6}{10} \cdot 10! \right)^6$$

Com es veu tot i que les restriccions ens redueixen el nombre d'estats la grandària total és enorme, cosa que fa impossible el tractament amb algorismes com A* o IDA*.

3. Operadors i espai de solucions

L'elecció dels operadors és sense dubte la part més difícil del problema, ja que l'elecció d'uns operadors o uns altres canvia radicalment les solucions que trobarà el programa i el temps que trigarà.

Accessibilitat d'estats i anàlisi d'operadors

Potser el més important de la funció successors és que sigui capaç de generar uns estats que permetin arribar a qualsevol solució del problema. En el nostre cas el problema té molts graus de llibertat i pràcticament no hi ha restriccions, només les de capacitat. Pel fet d'haver triat aquesta representació de l'estat tenim de forma implícita les restriccions sobre un camió per cada hora i centre.

Tot i això podem trobar un conjunt d'operadors que ens permetin accedir a tots els estats del problema però que, en general, això no sigui possible perquè per arribar-hi cal "baixar" la qualitat de la solució (des del punt de vista de la funció heurística). Aquest fet és molt important en els algorismes de cerca local, especialment Hill Climbing. És per això que només tenim dues alternatives: o bé canviar els operadors i buscar-ne uns altres que ens permetin fer el mateix o bé modificar la funció heurística per tal que premii d'alguna manera aquests camins per tal que siguin triats. Això últim és perillós ja que és preferible que la funció heurística no infravalori o sobrevalori els estats, si no fos així podríem arribar a estats no òptims sense voler.

Nosaltres hem fet un anàlisi de tots els operadors que se'ns han acudit:

- Moure entrega de no entregada a un camió (AFGR)

Aquest operador és necessari per tal de poder buidar la llista d'entregues sense camió. Aquesta llista pot estar no buida a causa d'una funció d'inicialització precària o bé degut a que hem tret entregues d'un camió. Pot aportar millora a l'estat actual (en general ho fa) i es pot aplicar sempre i quan hi hagi espai al camió i algun element a la llista de no entregats.

- Treure entregues d'un camió i no entregar-les (TREU): Aquest operador podria ser necessari per a poder accedir a tot l'espai de solucions. Per tal de moure un paquet d'un lloc a un altre podem aplicar aquest operador i l'anterior. Però degut a que sempre provoca un empitjorament de la qualitat de la solució (sigui quin sigui el criteri a millorar) és un operador que no s'utilitzarà mai. Caldria però considerar el seu ús en Simulated Annealing.

- Moure entregues entre camions (MOV): Per tal d'accedir a tot l'espai de solucions sense fer ús de l'operador anterior fem un nou operador que permeti girar dos entregues de dos camions qualsevols. També s'inclou la possibilitat de passar un paquet a un camió que estigui buit. Aquest operador pot aportar una millora a la solució actual i, per tant, és útil. Es pot aplicar sempre per cada parell d'entregues sempre i quan en moure-les es satisfacin les restriccions de pes dels camions.

- Girar camions (GIR): És molt important que es pugui distribuir els camions en les diferents hores i centres i, per aquest motiu, cal un operador que canviï dos camions. En principi aquest operador no aporta cap millora, ja que en el millor cas es giren els camions i les càrregues i tot queda exactament igual. Podríem fer ús de la funció heurística per a premiar aquest estat, però una solució més simple és la triada: Es reparteix la càrrega sobrant del camió gran entre els altres camions intentant millorar la solució; les entregues sobrants passen a ser no entregades. L'espai buit del camió gran s'omple amb entregues que no estaven a cap camió i d'aquesta manera, en aprofitar l'espai buit, ens avancem al que volíem fer: canviar les capacitats per aprofitar millor un camió gran en un centre amb molta càrrega.

Aquest operador es pot aplicar sempre per cada parell de camions, ja que els paquets sobrants es posen a la llista de no entregats, amb el que no es viola cap restricció del problema. Això és així degut a la representació de l'estat que hem triat.

- Girar entregues entregades i no entregades (GIE): Donada una entrega que no té camió i una altra que sí que en té es giren si els seus pesos ho permeten. Aquest operador permet treure entregues (decidir quines no s'entregaran) de forma efectiva enlloc de fer ús de TREU. Es podrà aplicar de forma similar a MOV però amb la restricció d'un camió, ja que només en triem un i la llista de no entregats no té cap restricció.

Càlcul del factor de ramificació

Operador	Ramificació	Estimació
AFGR	$NEntregSenseCamio \cdot NumHores$	$\max((ne - 233) \cdot 10, 0)$
TREU	$NEntregAmbCamio$	233
MOV	$\sum_c \left(\frac{NEntregAmbCamio_c}{2} \right) + NCamionsBuits_c \cdot NEntreg_c$	1265
GIR	$\left(\frac{NCamions}{2} \right)$	1770
GIE	$\sum_c (NEntregAmbCamio_c \cdot NEntregSenseCamio)$	$6 \cdot 4 \cdot (ne - 233)$

Per a realitzar l'estimació considerem les variables desconegudes com la seva esperança.

La capacitat mitjana d'un camió és $E\{Cap_{Camio}\} = \frac{1}{3} \cdot (500 + 1000 + 2000) = 1166$

La capacitat dels camions és $1166 \cdot 60 \approx 70000$

El pes mitjà d'una entrega: $E\{Pes_{Entrega}\} = \frac{1}{5} \cdot (100 + 200 + 300 + 400 + 500) = 300$

Nombre mitjà d'entregues amb camió: $E\{Num_{Entreg}\} = 70000 / 300 \approx 233$

El nombre d'entregues a cada camió és: $E\{Num_{EntregCamio}\} = 233 / 60 \approx 4$

Així doncs donat un nombre d'entregues *ne* calculem una estimació del nombre de successors. Per a MOV menyspreem el NCamionsBuits (que és molt proper a zero) i suposarem que els paquets estan distribuïts uniformement. Així doncs NentregAmbCamió serà 233/10. I com que el càlcul a priori de si dos paquets es poden girar és molt llarg i, en general, serà poc probable (els camions estaran molt plens) direm que és el 50% de probabilitat.

Conjunt d'operadors 1 (AFGR + MOV + GIR)

Triant els operadors AFGR, MOV i GIR podem accedir a tot l'espai de solucions. Potser l'única excepció és el cas en que es comenci amb un estat inicial que no té cap paquet sense enviar. En l'esmentat cas ens serà impossible arribar a la solució en que no s'entrega cap paquet. Això és cert per a Hill Climbing ja que no pot triar una solució pitjor que l'actual però no ho és per a Simulated Annealing, ja que sí que pot triar una solució dolenta. En aquest cas ens hauríem de qüestionar si afegim l'operador TREU. En qualsevol cas la definició d'estat accessible com a combinació d'operadors sobre un estat que porta a un altre és irrellevant en Hill Climbing ja que només tindrem en compte els operadors que ens fan millorar la solució.

Cal remarcar que la forma en que les entregues passen d'un camió a la llista de no entregades és fent ús de l'operador de girar camió. Això provoca que no puguem dir amb total seguretat si tot l'espai de solucions és accessible, ja que aquest operador treu els paquets que considera “poc beneficiosos” cosa que pot provocar que una entrega no es pugui treure i, per tant, estarem limitant l'espai de solucions accessibles a un subconjunt de l'espai total.

Per aquest conjunt d'operadors el factor de ramificació és de l'ordre de 3000. Contrastant les dades obtingudes durant l'experimentació veiem que aquesta ramificació és en situacions bastant desfavorables i que, en general, sempre està per sota. Així doncs el factor de ramificació és acceptable a priori, però cal saber el temps que es triga a generar aquests fills, ja que si fos molt elevat cada pas de Hill Climbing seria prohibitiu. En el cas del Simulated Annealing no ens hem de preocupar massa pel cost dels operadors, ja que en triem un a l'atzar sense generar-los tots. Experimentalment veiem que l'operador més costós és el GIR i la resta són de temps constant.

Conjunt d'operadors 2 (AFGR + GIE + GIR)

Aquest conjunt d'operadors es diferencia de l'anterior en que no té la possibilitat de girar dos paquets de dos camions qualsevols de forma directa. Ho ha de fer a través de dos operacions GIE. Això a priori no sabem si és millor o pitjor però la pràctica ens demostra que l'operador MOV no es fa servir gaire i que eliminar-lo no suposa cap pèrdua.

L'operador GIE ens dona la possibilitat de treure entregues que “ens molesten” de forma ràpida i directa. De fet si l'inicialització és molt dolenta aquest operador és decisiu (ho hem comprovat) ja que treu del mig entregues que, tot i donar benefici, priven a altres entregues d'agrupar-se per a donar un benefici equivalent. És com un túnel que comunica estats allunyats de benefici similar però radicalment diferents.

Experimentalment veiem que hi ha un nombre mitjà de 2000 a 3000 successors. Tot i això el factor de ramificació tendeix ràpidament a valors petits com ara de 1000 a 500.

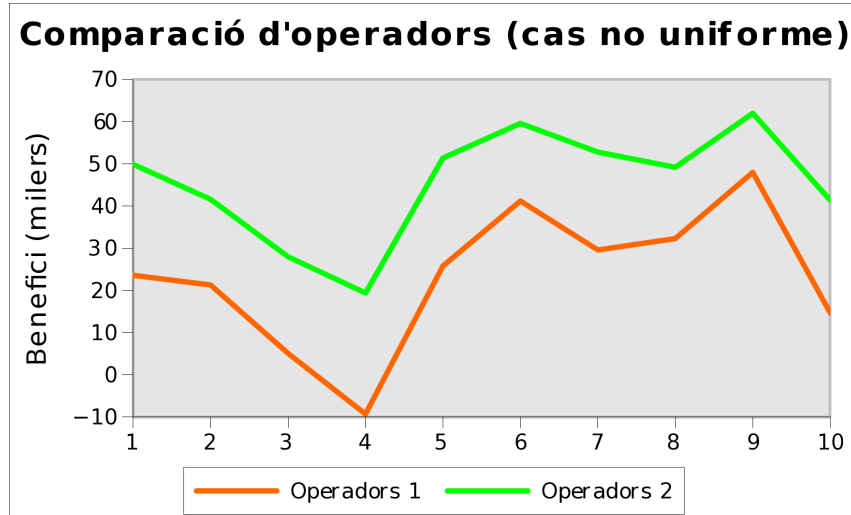
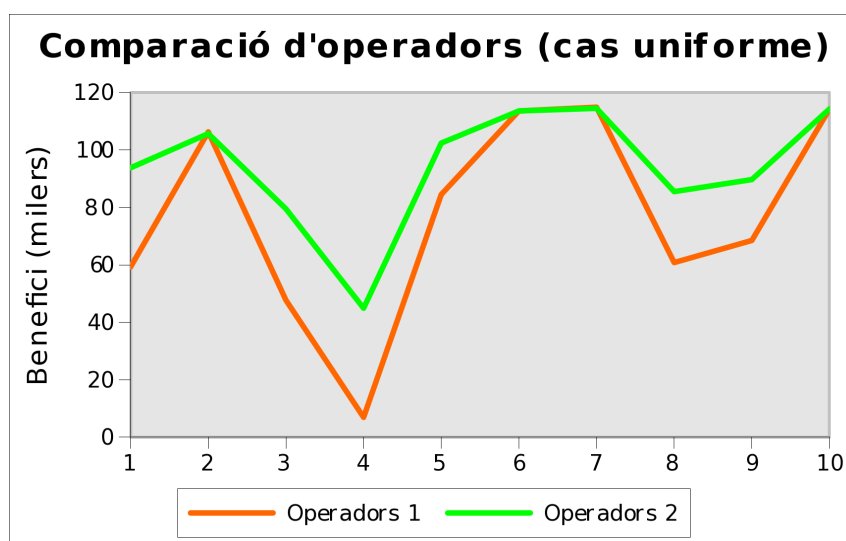
Experimentació: comparació de la qualitat dels operadors

Comparem els dos operadors sota les següents condicions inicials: probabilitat uniforme (tots els paràmetres), 250 entregues, Hill Climbing i inicialització intel·ligent. En aquest cas tracem unes gràfiques comparatives per a cada una de les proves.

	Benefici final (milers) (equiprob.)										
Operadors 1	59.3	106.3	47.7	6.9	84.5	113.6	114.9	60.8	68.5	114.3	77.7
Operadors 2	93.8	105.7	79.5	44.9	102.4	113.6	114.5	85.5	89.7	114.3	94.4

A més a més realitzem unes proves en una situació desequilibrada per posar a prova els dos conjunts d'operadors. Aquesta prova consisteix en un centre amb probabilitat 50% i la primera i última hora amb probabilitat 50%.

	Benefici final (milers) (no equiprob.)										
Operadors 1	23.6	21.3	5.1	-9.3	25.8	41.2	29.6	32.3	48.0	14.7	24.2
Operadors 2	49.9	41.6	28.0	19.4	51.4	59.6	52.8	49.2	62.0	41.4	45.5



Així doncs el conjunt d'operadors 2 és el millor segons aquests resultats, pel que d'ara en endavant el triarem com a operador per defecte. Com es pot veure és especialment bo en el cas d'una distribució no uniforme, on és capaç de treure un benefici molt superior. En el cas habitual (uniforme) el benefici també és igualment superior però en alguns casos és el mateix que el conjunt d'operadors 1.

Experimentació: comparació de la velocitat dels operadors

Sota les mateixes condicions anterior fem un anàlisi del temps de càlcul.

	Temps (segons)										
Operadors 1	6.6	3.1	4.3	12.0	13.9	1.5	6.4	13.4	10.7	4.6	7.7
Operadors 2	32.9	4.5	25.2	25.3	22.3	1.4	6.0	27.7	24.7	3.5	17.4

Ja veiem que els segon conjunt d'operadors és més lent (tot i ser bo). Per aquest motiu hem de decidir si volem una millor solució a costa de més temps o una solució pitjor en molt menys temps. Per això calcularem un criteri de qualitat que relaciona el benefici amb el temps.

	Benefici per unitat de temps (€/s)										
Operadors 1	111.3	29.2	90.3	1739	164.5	13.2	55.7	220.7	156.0	40.3	262.0
Operadors 2	350.4	42.5	317.0	563.5	217.8	12.3	52.4	324.0	275.0	30.6	218.6

La disparitat de valors ens dona una idea vista a classe i és que trobar una solució bona pot portar molt poc temps o bé molt independentment de la qualitat de la solució. Això és característic dels algorismes voraçs, ja que avancen sempre en una direcció i només s'aturen quan acaben. Val a dir que per reduir el temps es poden fer operadors que facin més d'una operació a la vegada (així es redueix el nombre de passos) o bé el contrari, descomposar una operació complexa en altres de simples, amb el que es creen menys successors a cada pas. Existeix un trade-off entre el nombre de successors i el temps de computar els successors.

Observem que els dos operadors troben la solució en un temps més o menys proporcional a la qualitat d'aquesta (en mitjana però).

4. Generació de l'estat inicial

Hem creat quatre possibles funcions d'inicialització. Aquestes intenten anar des de molt dolenta fins a molt bona. Diem "intenten" ja que moltes vegades un estat inicial dolent pot portar a un de molt bo.

Nosaltres només ens hem centrat en la repartició dels camions a cada centre i hora, ja que la repartició dels paquets és comuna a totes les inicialitzacions. Aquesta repartició intenta afegir el màxim nombre d'entregues d'una forma conservadora. Tot i que podríem haver fet que cada inicialització tractés les comandes de forma diferent hem vist que l'algorisme d'IA ho arregla amb facilitat i que el que li costa més i és més crític sempre és la tria de les capacitats dels camions.

Descripció de les funcions d'inicialització

- Inicialització desequilibrada

Aquesta inicialització la considerem la més dolenta (tot i que pot no ser-ho perfectament) ja que el que fa és assignar de forma molt descompensada les capacitats dels camions. Assigna els camions de gran a petit del centre 1 al 6.

- Inicialització equilibrada

La rutina és senzilla i intenta repartir les capacitats dels camions equitativament entre tots els centres (sense tenir en compte quin té més demanda) i tampoc té en compte les hores de més demanda. Veurem que això és suficient per arribar més ràpidament a la solució ja que, en mitjana, una solució uniforme és molt propera a qualsevol altra solució, més que no pas una solució extrema (la desequilibrada) a una altra solució extrema (imaginem per exemple un desequilibri diferent a l'inicial).

- Inicialització aleatòria

Aquesta solució dona més bons resultats dels esperats. En principi, donat que els nombres estan uniformement distribuïts, qualsevol estat inicial és possible. Però també és cert que hi ha molts més estats inicials possibles que són equitatius que no pas estats no equitatius.

- Inicialització intel·ligent

Intentarem triar els camions de forma que minimitzem l'espai sobrant i l'espai que ens falta per omplir-lo. Per a fer-ho calculem a cada hora i centre quin volum de demanda hi ha i assignem els camions més grans als centres que més ho necessiten. Pot semblar un bon inici però sempre tindrem el perill d'arribar a un màxim local (ja que comencem suposadament molt amunt en la funció heurística).

Anàlisi de les funcions d'inicialització

Podríem comparar el cost del càlcul de les solucions inicials, però aquest cost és molt petit. Això és degut a que només repartim els camions i no pas les entregues. A més l'únic algorisme intel·ligent que tenim és *greedy* i no itera

millorant la solució ni res semblant. Cada decisió que fa no es canvia.

	Cost computacional
No equilibrat	$O(\text{NumCentres} * \text{NumHores})$
Equilibrat	$O(\text{NumCentres} * \text{NumHores})$
Aleatori	$O(\text{NumCentres} * \text{NumHores})$
Intel·ligent	$O((\text{NumCentres} * \text{NumHores})^2)$

Donat el poc cost de generar la solució inicial la relació entre la bondat de la solució i el cost serà directament la bondat d'aquesta. Ara per tal de poder fer els experiments necessitem un criteri per a avaluar la bondat de la solució. Dos possibles criteris (que creiem rellevants) són: el benefici de la solució inicial (o el retard, segons el que busquem) i el benefici de la solució final (que hi està directament relacionat, sobretot amb Hill Climbing que és determinista).

Experimentació: qualitat de les solucions inicials

Realitzem 10 experiments amb *seeds* del 1 al 10 i mirem quin és el benefici de la solució inicial i final. Per al cas aleatori farem de cada experiment 5 realitzacions amb *seeds* del 1 al 5 i en farem la mitjana. Triem el nombre de 250 entregues i fem ús del conjunt d'operadors 2 amb Hill Climbing.

	Benefici inicial (milers)										
No equilibrat	-21.5	63.3	14.9	-33.4	5.7	61.7	45.2	8.7	-4.6	67.0	20.7
Equilibrat	47.7	97.0	28.1	-7.2	76.0	103.8	104.8	50.0	49.1	117	66.6
Aleatori	27.6	68.2	14.4	-21.7	45.0	97.7	80.4	27.6	42.2	99.7	48.1
Intel·ligent	35.3	91.1	27.5	-16.0	61.4	109.1	97.2	38.1	45.9	110.5	60.0

	Benefici final (milers)										
No equilibrat	81.7	101.5	80.6	46.8	94.7	110.9	109.8	74.8	81.1	108.1	89.0
Equilibrat	98.9	109.7	89.3	57.2	108.9	115.4	116.9	94.7	99.7	117.0	100.8
Aleatori	91.2	103.5	76.2	47.3	101.7	112.3	112.3	82.8	93.7	113.3	93.4
Intel·ligent	93.9	105.7	79.5	44.9	102.4	113.6	114.5	85.5	89.8	114.2	94.4

Podem veure com en benefici inicial l'inicialitzador equilibrat (que simplement distribueix camions de forma equitativa a cada centre) és millor que l'intel·ligent. En estat final el supera molt lleugerament. Cal recordar que en les proves els centres eren equiprobables així com les hores del dia, cosa que explica el que està passant: per al cas equiprobable equilibrat i intel·ligent són algorismes equivalents.

A més a més destaquem que el benefici de l'estat final és quasi igual per a tots els estats inicials. Això vol dir que si hem triat bé aquests operadors (produeixen bones solucions) no ens hem de preocupar massa per l'estat inicial. Quasi podríem dir que la qualitat de l'estat final no depèn de l'estat inicial!

Experimentació: cas no equilibrat

Ara volem provar la hipòtesi de que tots els inicialitzadors són molt semblants degut a que les condicions inicials distribueixen camions i entregues de forma uniforme. Així doncs farem la prova següent: en les mateixes condicions anteriors variem la concentració d'entregues que serà amb 50% de probabilitat al centre 6 i 50% a la resta. Similarment per les hores l'última i la primera hora rebran el 50% de les comandes.

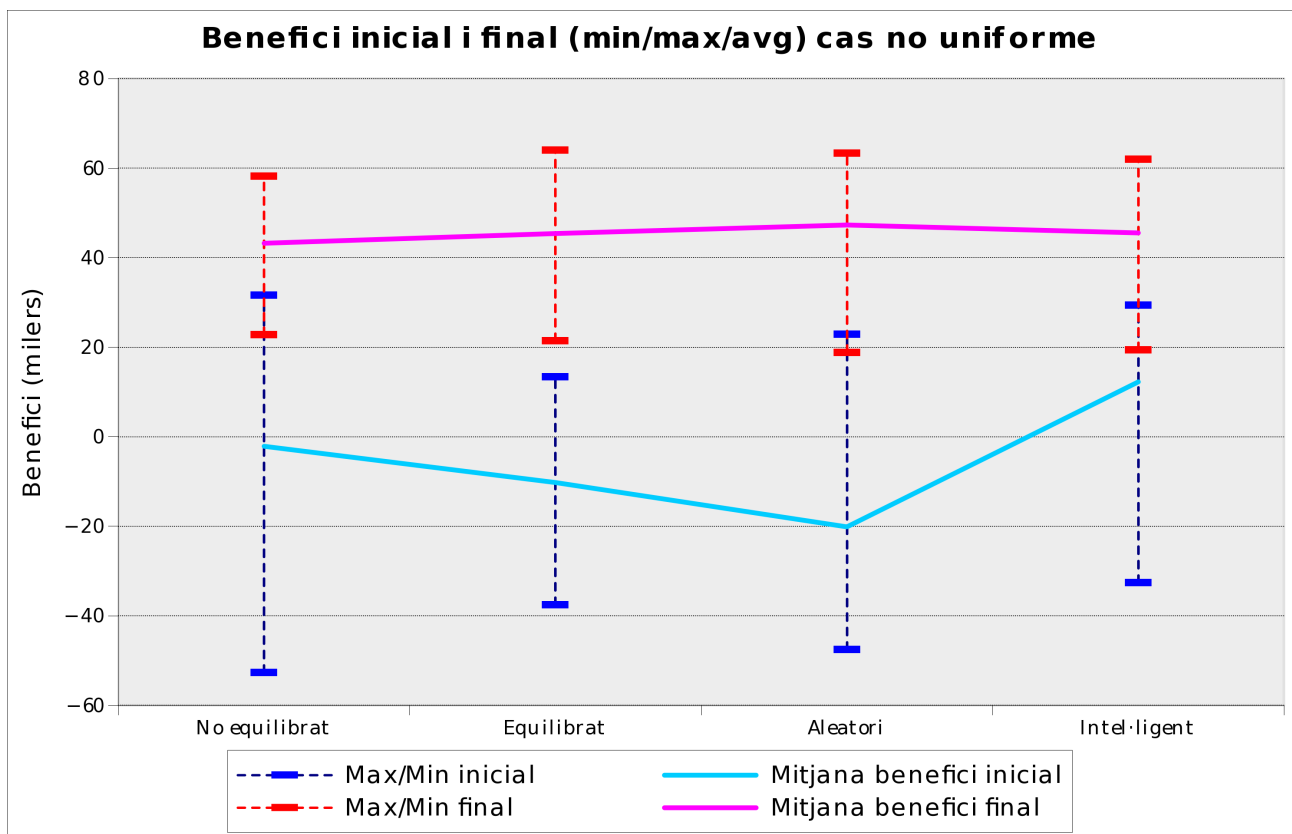
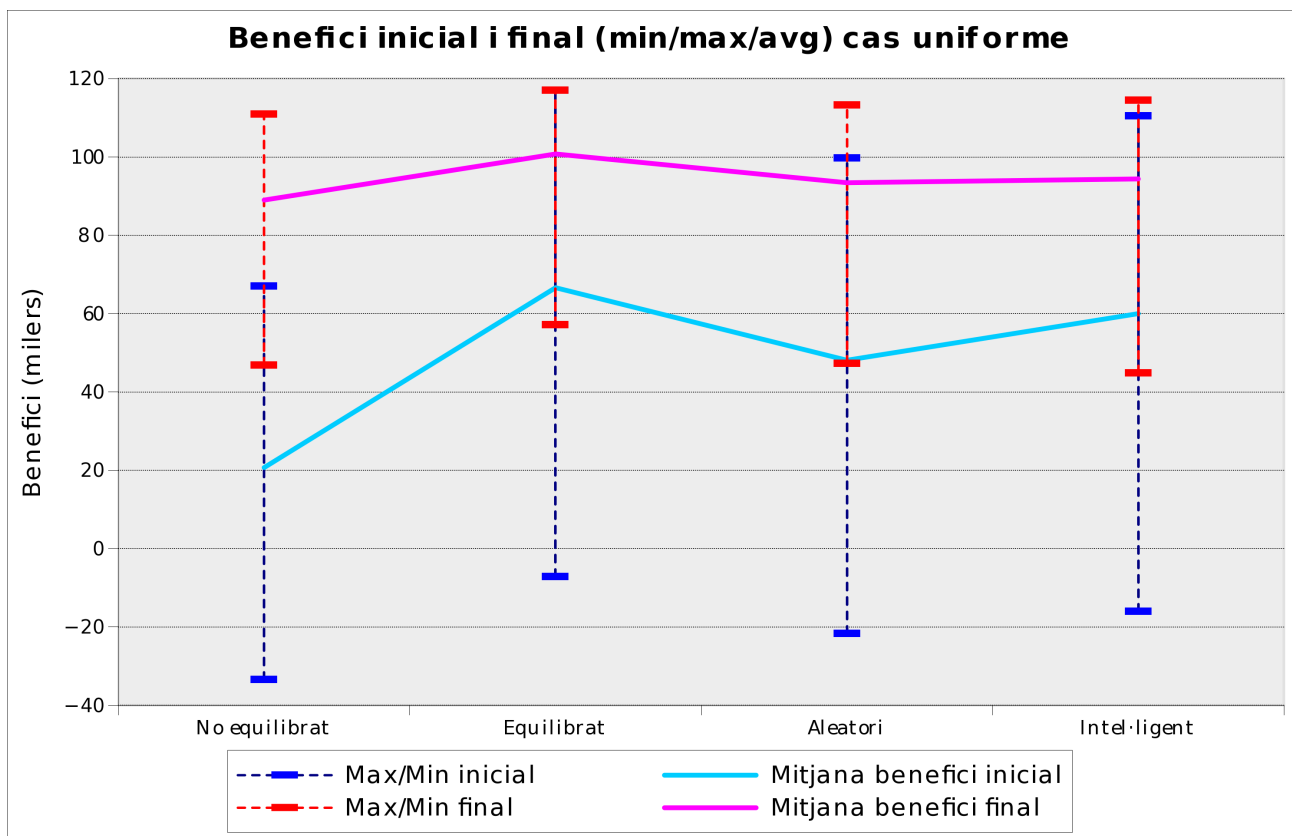
	Benefici inicial (milers)										
No equilibrat	-0.7	12.3	-13.3	-52.7	-8.5	31.6	5.9	-6.9	-1.5	12.5	-2.3
Equilibrat	-16.1	-9.5	-34.2	-37.6	-7.3	13.4	0.5	-9.3	3.5	-5.7	-10.3
Aleatori	-46.6	-32.1	-41.8	-47.6	-15.3	12.8	22.9	-37.8	-23.1	7.3	-20.1
Intel·ligent	14.5	19.1	-2.3	-32.6	19.5	24.9	22.4	13.6	29.4	14.1	12.2

	Benefici final (milers)										
No equilibrat	44.9	39.0	22.8	22.9	46.5	57.6	49.7	50.5	58.2	40.2	43.2
Equilibrat	43.5	43.3	25.0	21.4	52.9	64.0	54.4	45.5	58.0	46.0	45.4
Aleatori	53.7	43.4	28.2	18.8	53.9	61.0	52.3	56.0	63.3	42.5	47.3
Intel·ligent	50.0	41.6	28.0	19.4	51.4	59.6	52.8	49.2	62.0	41.4	45.5

Com veiem només la versió intel·ligent és capaç de generar estats inicials amb bon benefici mentre que les altres no. Novament veiem com el resultat final no difereix massa de la inicialització. Això reforça la idea que ja hem comentat. Fins i tot podem veure que l'aleatori, tot i començar amb la pitjor solució, iguala i fins i tot supera a la intel·ligent.

Il·lustrem els resultats en unes gràfiques min/max/mitjana on s'aprecia la qualitat de cada inicialització així com una idea aproximada de la variança.

Destaquem que el benefici final és una recta quasi plana i que no hi ha massa relació de proporcionalitat entre beneficis de solucions inicials i finals.



5. Disseny i anàlisi de la funció heurística

S'ha dissenyat dues funcions heurístiques tal com es demanava a l'enunciat. La primera maximitza el benefici a la vegada que intenta minimitzar el retard, mentre que la segona funció minimitza el retard sense tenir en compte el benefici.

Descripció de les funcions

- Maximització del benefici

La funció retorna simplement el benefici de l'estat calculat tal com es diu a l'enunciat. Per a minimitzar el retard de forma simultània el que fem és fer ús dels decimals (donat que retornem un real), així la part entera és el benefici i els decimals expressen el retard. D'aquesta forma d'entre dos estats amb el mateix benefici es podrà distingir quin és millor gràcies al retard.

- Minimització del retard

La funció és senzilla, simplement retornem el retard calculat de la forma que diu l'enunciat.

Anàlisi de les funcions heurístiques

L'avantatge principal de fer ús de dos paràmetres és que difícilment acabarem en una zona plana (un altiplà). Recordem que l'algorisme de Hill Climbing es pot quedar estancat en una zona plana (on no hi ha millora i només s'igualava o s'empitjora la solució actual) i això provoca l'aturada de la cerca. Si tenim un segon paràmetre a millorar aleshores és molt possible que una solució amb el mateix benefici però amb menor retard oferirà més qualitat a la solució i d'aquesta manera l'algorisme continuarà iterant.

De forma experimental veiem que de vegades passem per un o dos estats amb el mateix benefici que l'anterior amb el que aconseguim eliminar un dels problemes típics de HC que són les zones planes. El guany en general no és molt (creiem) però sí que ajuda en situacions properes a un màxim on el nombre de successors és petit i no es millora la solució.

Les dues funcions trien sempre l'estat que millora la solució ja que sempre retornen el valor real del paràmetre a optimitzar sense afegir o treure valor. La primera a més fa ús d'un petit nivell d'optimisme per tal de premiar les solucions empatades a benefici però manté la propietat que dos estats amb beneficis b_1 i b_2 tals que $b_1 < b_2$ compleixen que $h(e_1) < h(e_2)$. D'aquesta manera sempre es tria l'estat millor i només diferenciem entre estats iguals.

Comparació de les funcions heurístiques

Volem veure si pel fet de millorar una característica l'altra es millora també o no. I volem saber si la funció que millora el benefici és capaç d'obtenir menor retard que la que només minimitza el retard (cosa que podria ser donat

que fa servir el retard com a segon paràmetre).

Les condicions de l'experiment són 250 entregues, Hill Climbing, var. aleatòries unif. distribuïdes, inicialització intel·ligent i conjunt d'operadors 2.

	Benefici final										
Max. benef.	93.9	105.7	79.5	44.9	102.4	113.6	114.5	85.5	89.8	114.2	96.6
	76.0	106.9	101.4	113.1	99.90	112.6	108.8	110.0	120.6	40.0	
	66.1	101.3	110.0	113.6	92.7	118.6	116.2	45.1	83.8	116.2	
Min. retard	66.8	103.1	44.6	104.7	72.8	113.4	112.1	57.5	68.9	114.2	85.9
	48.9	91.1	97.5	112.5	85.2	110.6	104.4	110.2	120.8	15.8	
	27.5	95.2	111.1	111.8	68.1	117.9	109.7	15.1	51.0	115.4	

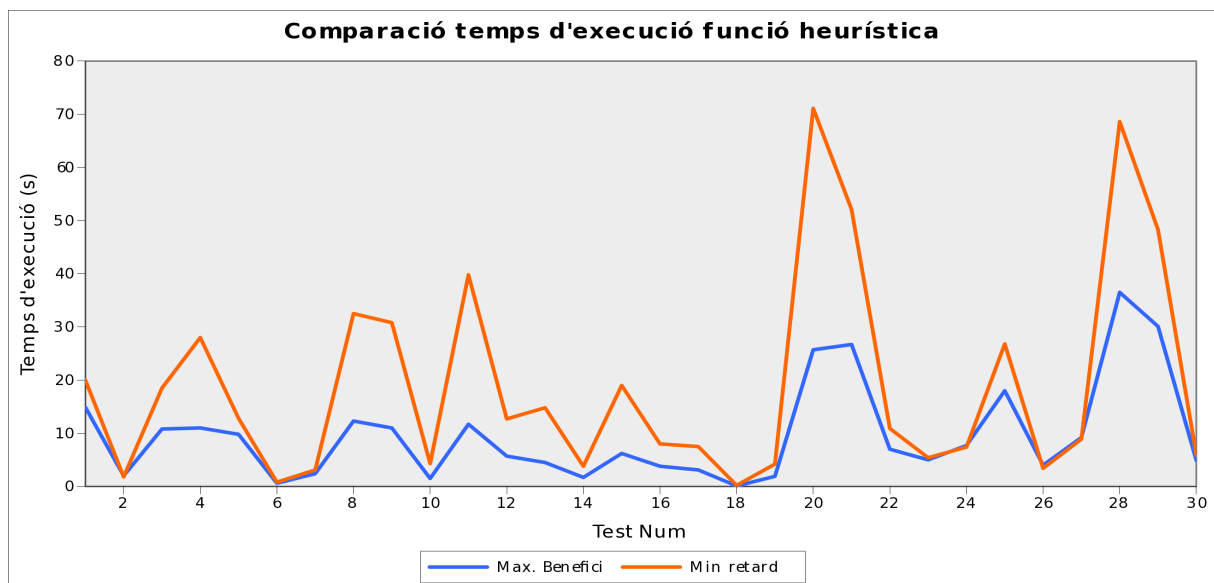
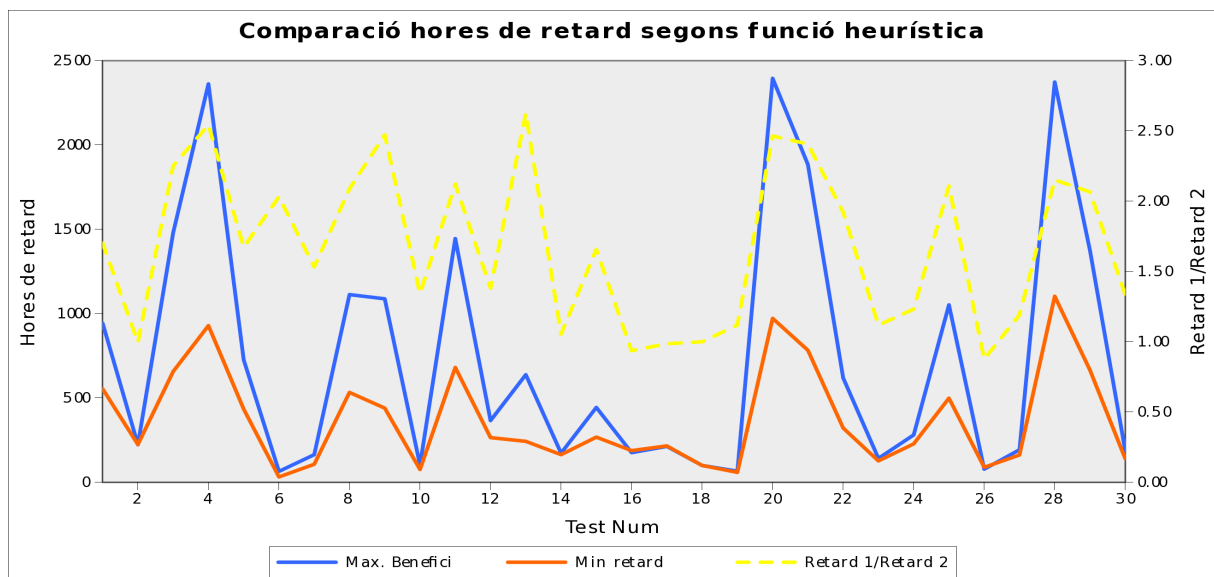
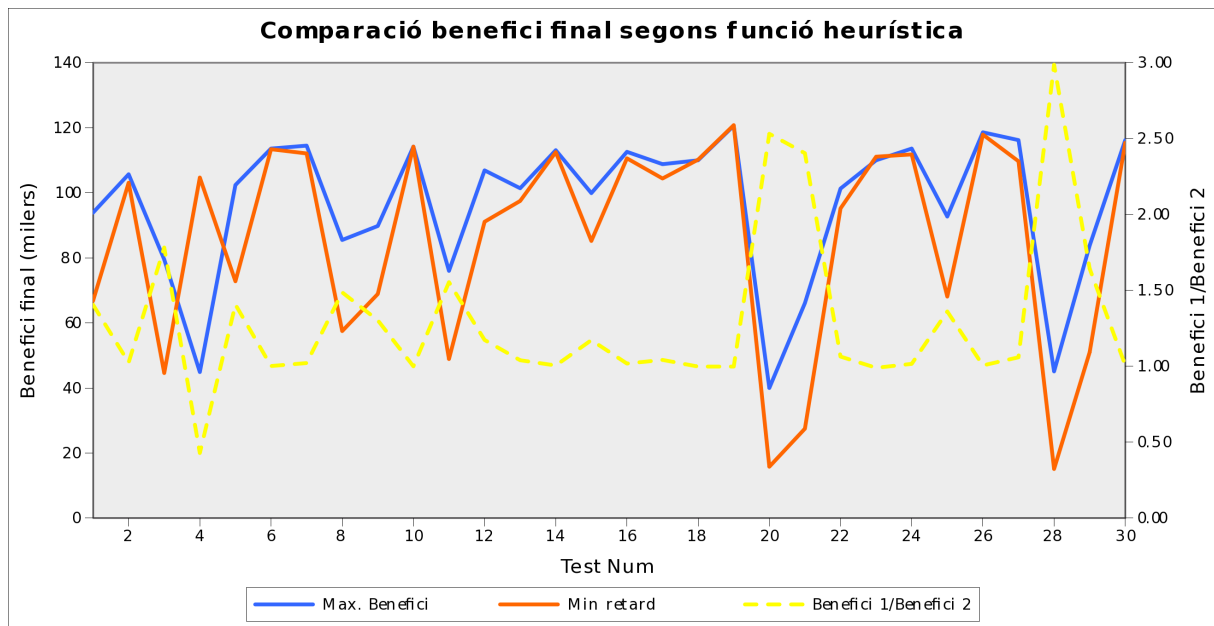
	Hores de retard										
Max. benef.	943	223	1480	2362	726	65	164	1113	1088	102	748.8
	1445	366	638	173	444	176	213	100	66	2395	
	1884	619	142	281	1052	78	193	2373	1374	187	
Min. retard	554	224	657	929	434	32	107	533	440	76	375.3
	681	265	243	164	268	188	216	100	59	972	
	783	322	127	228	499	89	162	1103	665	140	

Arribem a confirmar les hipòtesis de les que havíem partit: maximitzar un criteri de qualitat ajuda a maximitzar l'altre. Evidentment cada funció és la millor optimitzant la seva variable, seria absurd optimitzar una variable i obtenir una optimització millor en una altra variable.

En observar les gràfiques (al final) veiem que sembla que existeix una relació de proporcionalitat entre els dos gràfics. En alguns punts són iguals però en d'altres el gràfic sembla que hagi estat escalat. Per aquest motiu tracem una corba que expressa la relació entre les dues solucions (corba groga) i veiem com en el cas de l'optimització del benefici final la corba es mou sempre al voltant del 1 amb pics puntuals molt estrets. Veiem que existeix una relació de proporcionalitat entre les dues solucions que té un factor de proporcionalitat d'aproximadament 1.30. Això vol dir que el criteri de mínim retard obté en mitjana un 77% de benefici respecte l'altre criteri.

Establir aquesta relació amb el segon criteri no és tan obvi, ja que la corba no sembla estar sempre al voltant d'un valor. El que sí és veritat és que es mou entre un rang de valors de 1 i 2.5 aproximadament. L'única conclusió a la que podem arribar és que, en general i en pitjor cas, la solució fent ús del criteri de màxim benefici té fins a 2.5 vegades més retard que la solució que minimitza el retard.

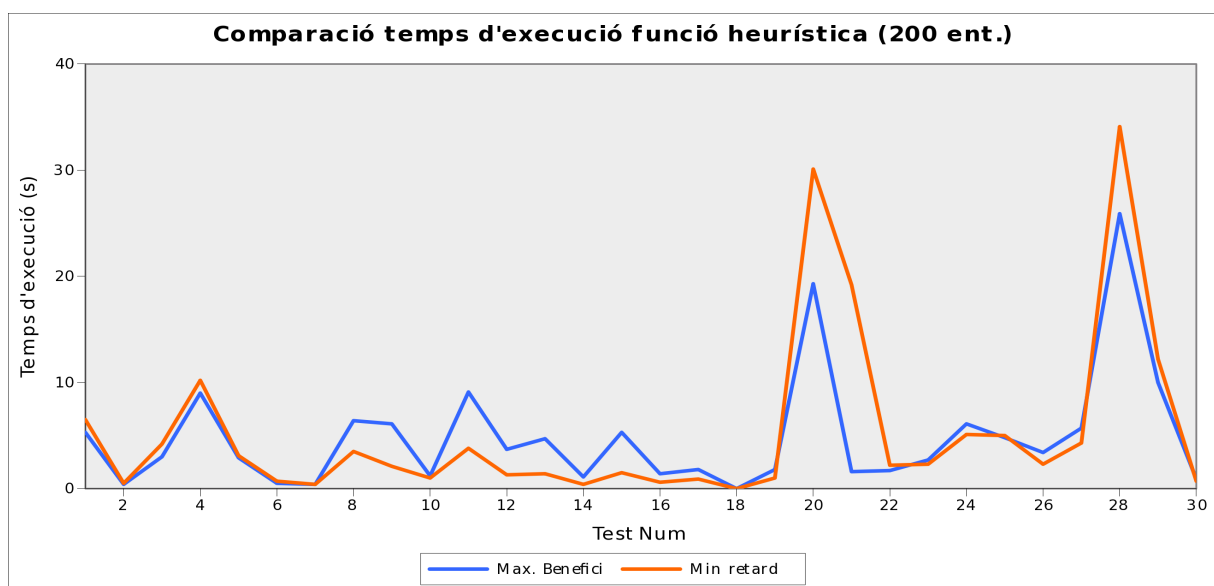
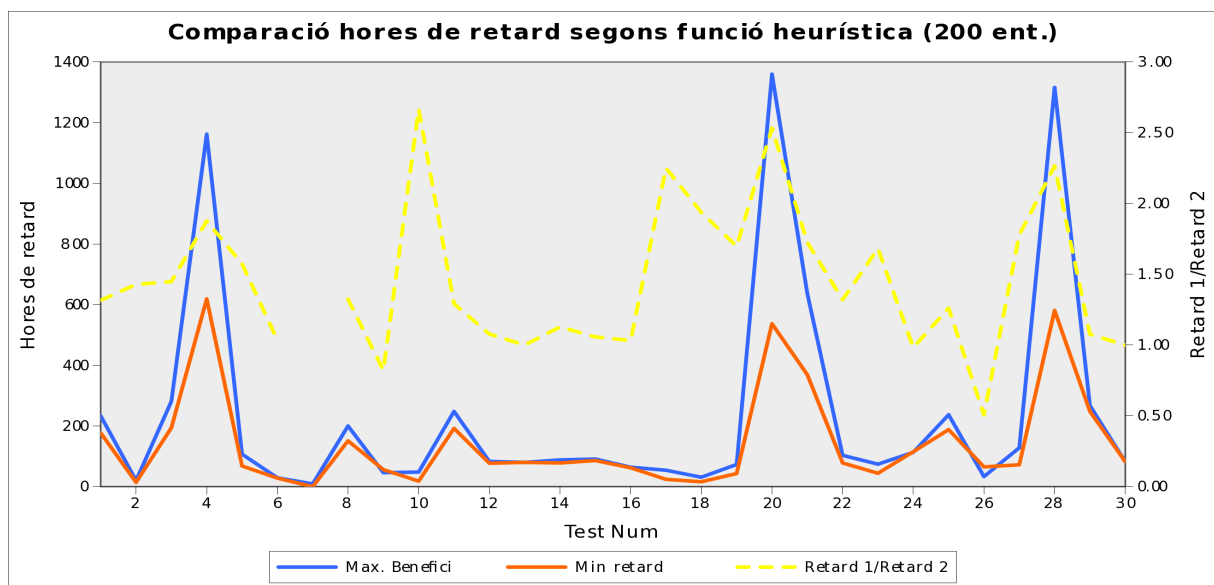
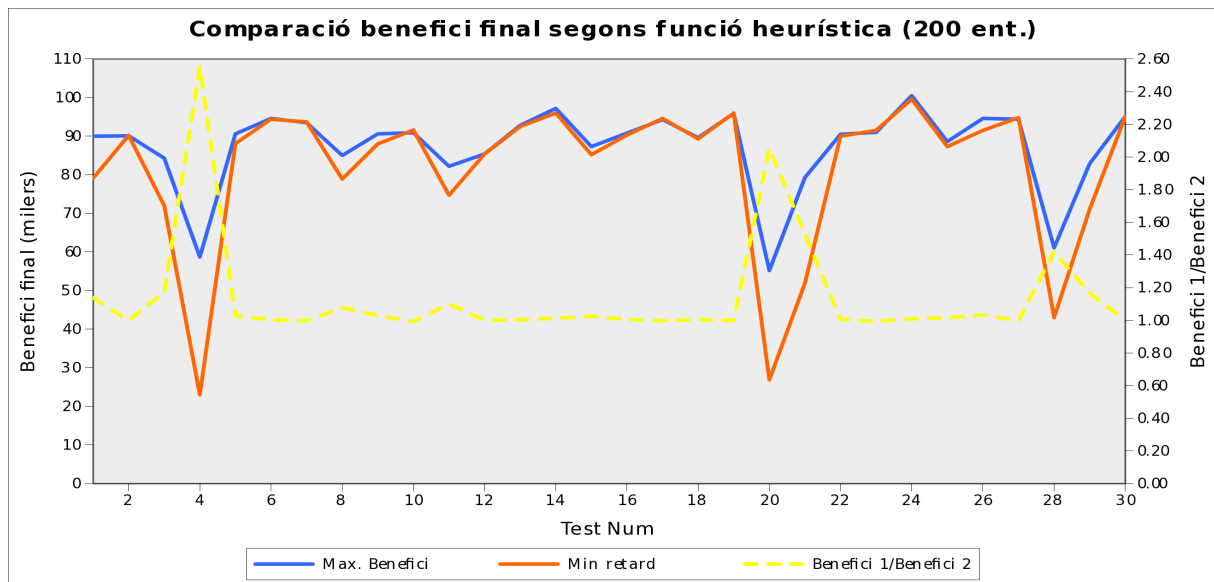
	Temps d'execució										
Max. benef.	14.9	2.0	10.8	11.0	9.8	0.6	2.4	12.3	11.0	1.5	9.6
	11.7	5.7	4.5	1.7	6.2	3.8	3.1	0.1	1.9	25.7	
	26.7	7.0	5.0	7.7	18.0	4.0	9.2	36.5	30.1	4.9	
Min. retard	20.0	1.8	18.5	28.0	12.8	0.8	3.1	32.5	30.8	4.3	19.0
	39.8	12.7	14.8	3.8	19.0	8.0	7.5	0.2	4.2	71.1	
	52.1	10.9	5.4	7.4	26.8	3.4	8.9	68.6	48.3	5.9	

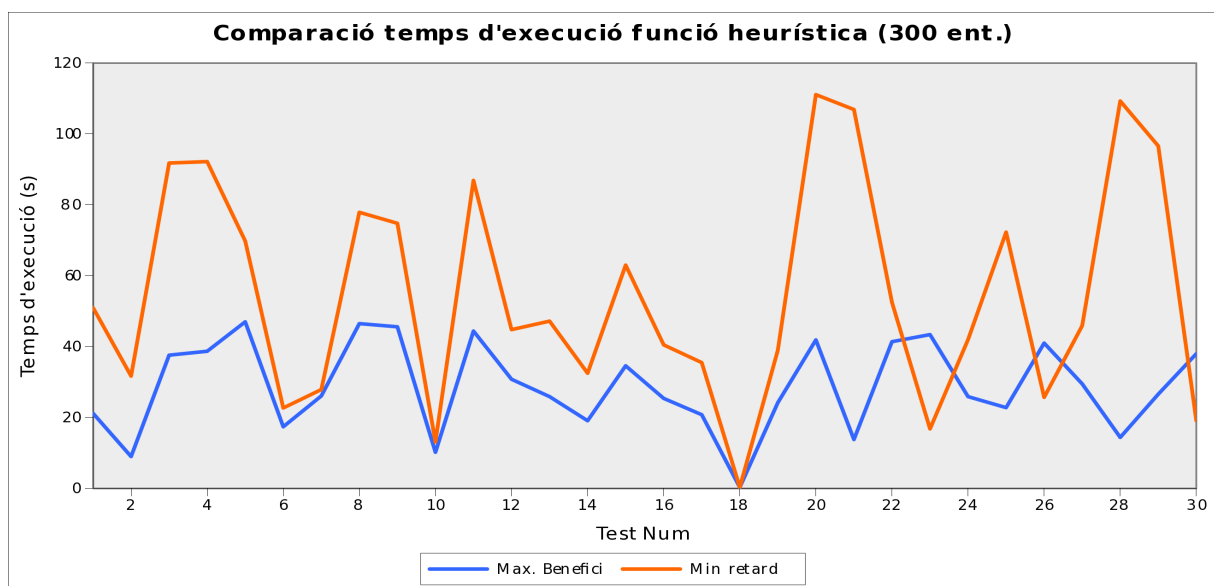
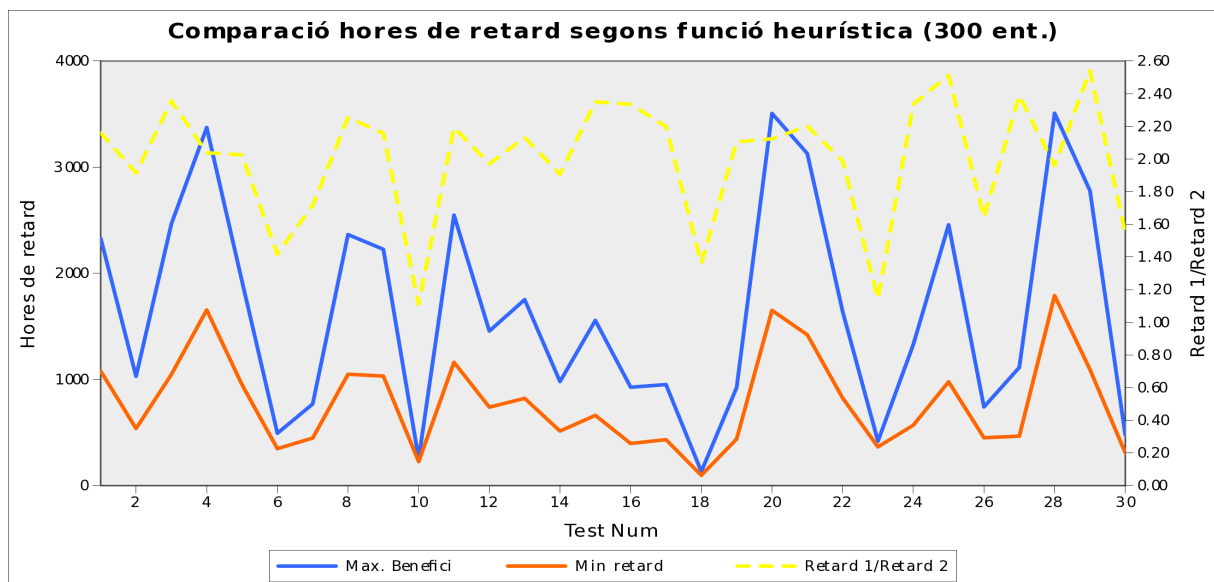
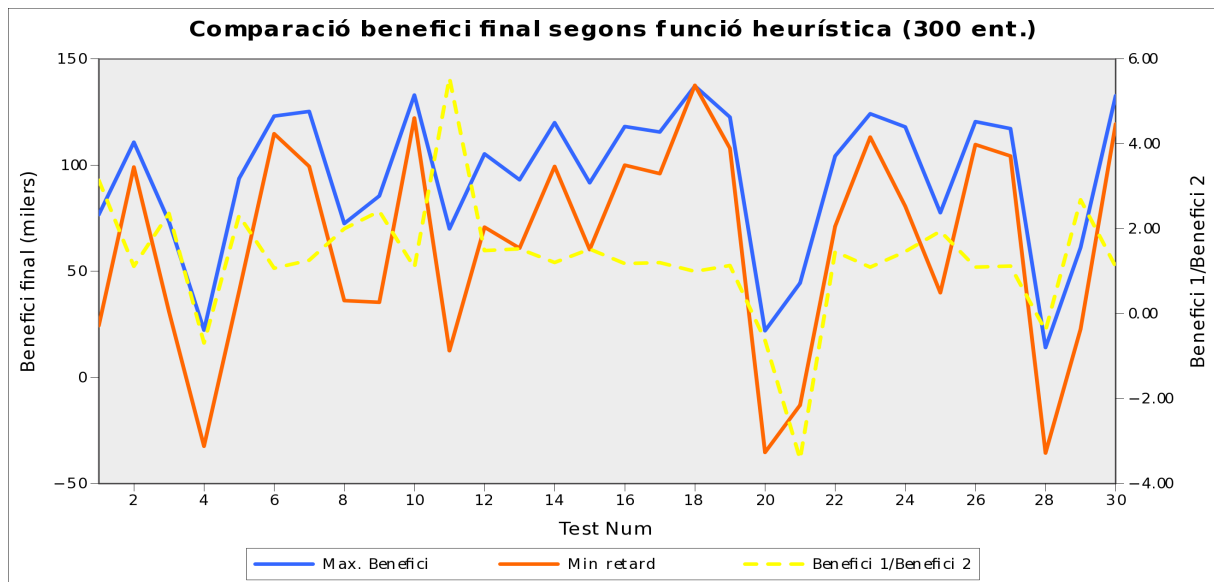


Tal com se'ns demana i per a corroborar els resultats amb les hipòtesis, provem ara 200 i 300 entregues. En aquests dos casos comprovem com es manté la relació entre beneficis esmentada així com la relació entre el retard de les entregues.

Pel que fa al temps de procés que encara no hem comentat, es veu clarament que no segueix cap mena de patró. Tal com hem dit en repetides ocasions el temps de procés sol ser independent de la qualitat de la solució i pren valors molt dispars. Això és lògic donat que depèn del nombre de passos, que no podem determinar ni relacionar a priori amb les solucions ni les heurístiques.

A continuació mostrem les corbes per als casos esmentats. Les taules es troben al final de la memòria ja que no aporten res i són molt grans.





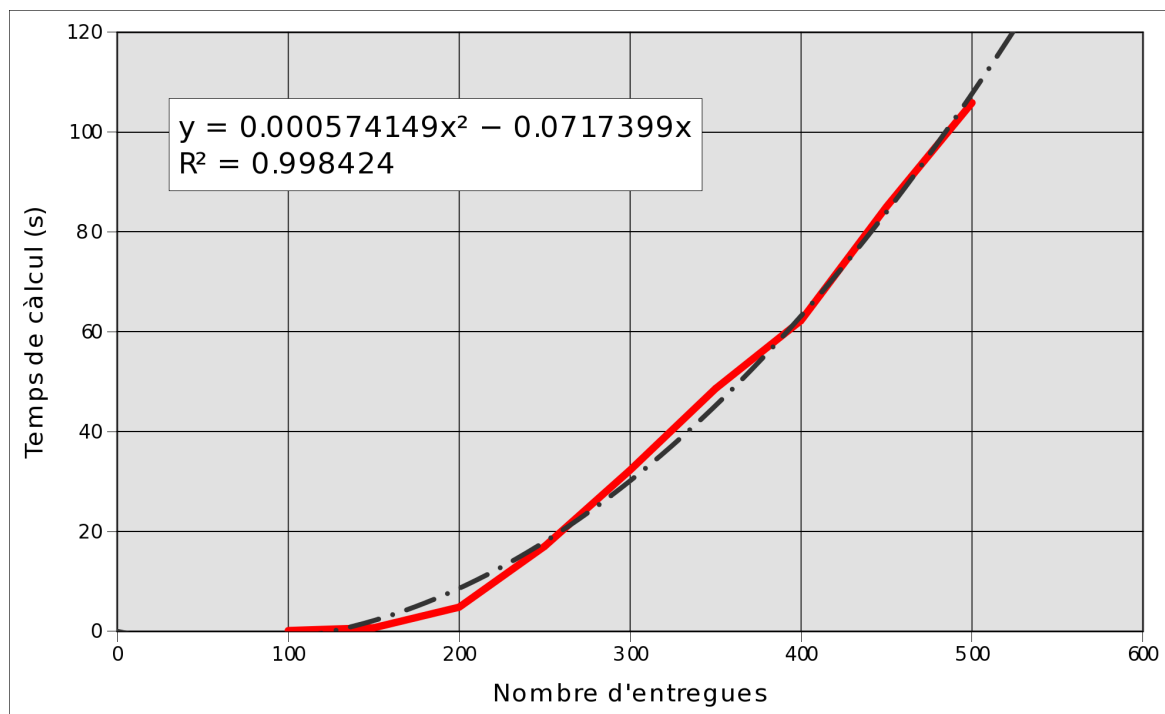
6. Experimentació

En aquest apartat s'exposen els resultats dels experiments que es demanen a l'enunciat i es comenten els resultats per veure si tot funciona com s'esperava i, en cas de no ser així, el perquè.

Anàlisi temporal de l'algorisme

En aquest experiment s'intenta relacionar el temps d'execució de l'algorisme amb el nombre d'entregues del problema. Es prova començant en 200 i a passos de 50 entregues. Les taules són grans i és per això que les posem als annexos.

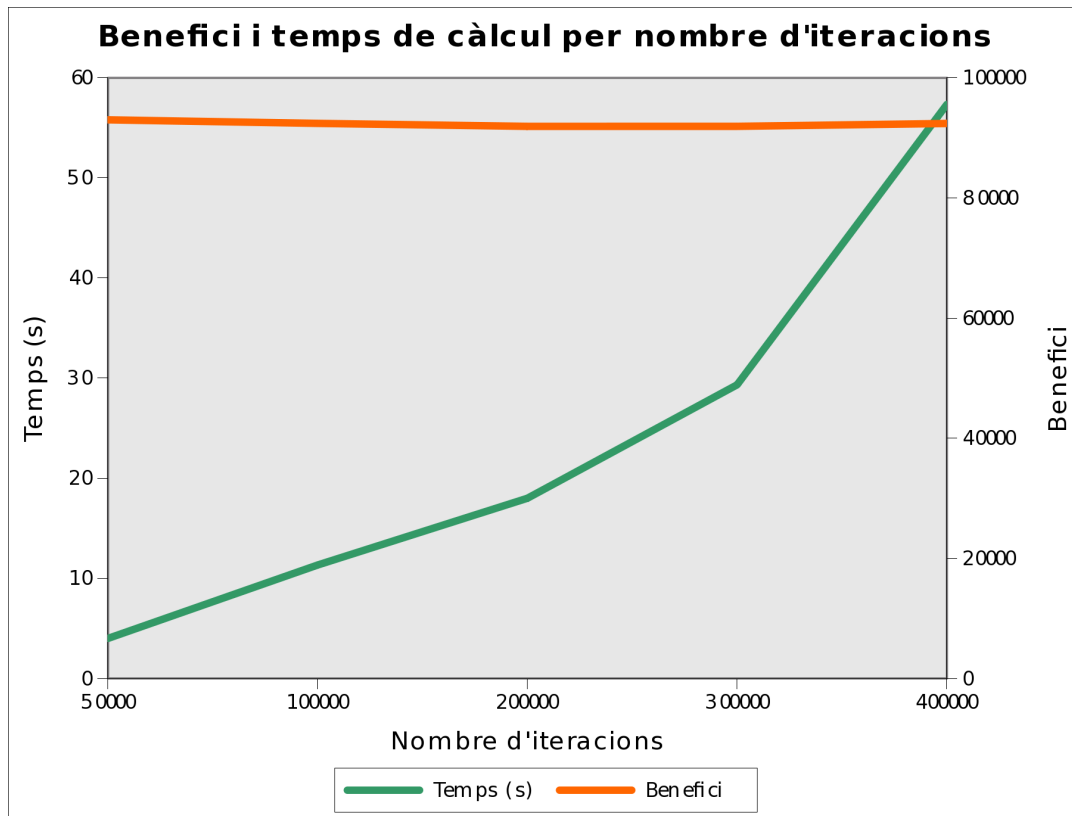
La gràfica de la funció temps és la següent.



Es regeix per un polinomi de 2on grau tal com es veu a la gràfica de la regressió. A més l'error (R^2) és molt proper a 1, això vol dir que les gràfiques són molt semblants i per tant podem dir sense cap problema que l'algorisme té una complexitat $O(n^2)$.

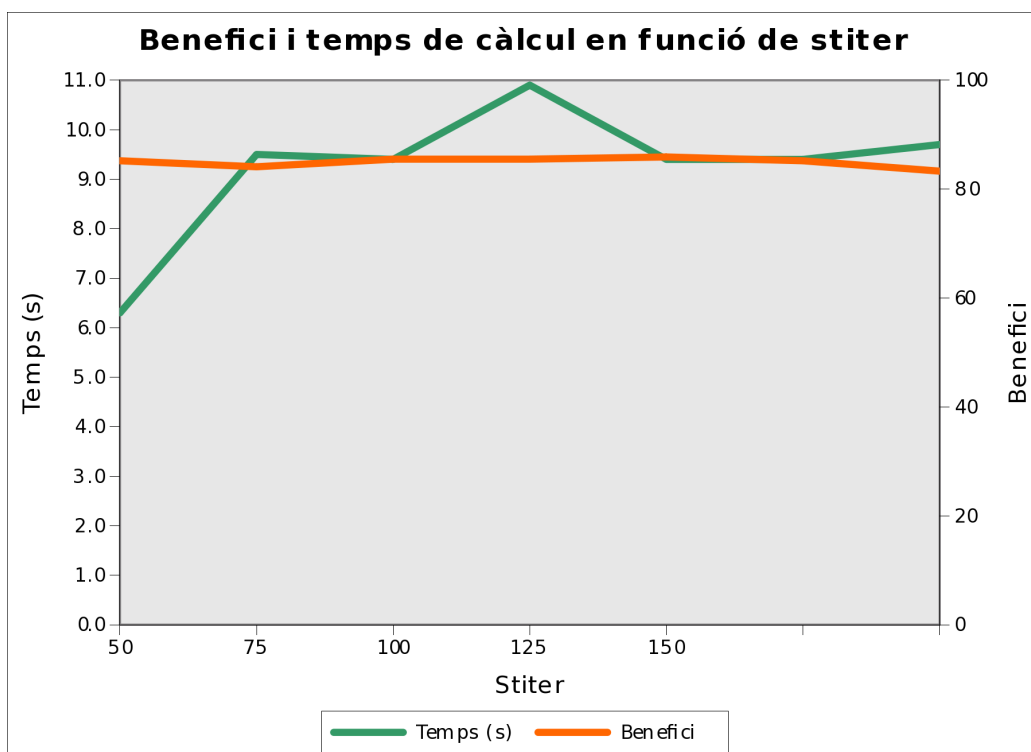
Càlcul dels paràmetres òptims per Simulated Annealing

Amb l'escenari de 250 entregues, operadors 2, variables aleatòries distribuïdes de forma uniforme i inicialització intel·ligent busquem la millor combinació de paràmetres per a S.A. Donat que hi ha moltes combinacions possibles de valors en fixem uns i fem un escombrat amb el que queda. Aquesta part és molt qualitativa, així que no la farem gaire exacta. A més que no hi ha massa diferència en benefici, només en temps és significativa. Triem els valors per defecte per als valors que no escombrem.



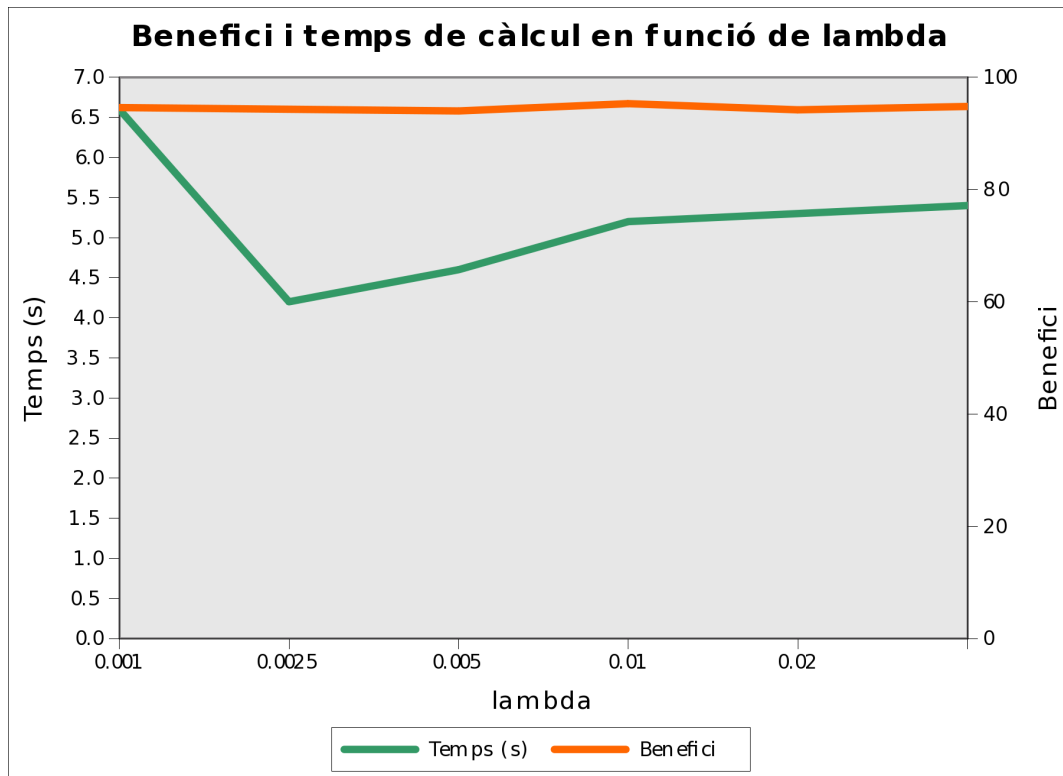
Com es pot apreciar el benefici no depèn gaire del nombre d'iteracions pel que podem triar un nombre baix d'iteracions i aconseguir així un temps molt menor de càlcul. Triarem 50.000 com a valor per defecte ja que és dels més petits i creiem que suficientment gran com per a evitar caure en realitzacions dolentes.

El següent experiment intenta determinar el paràmetre stiter.

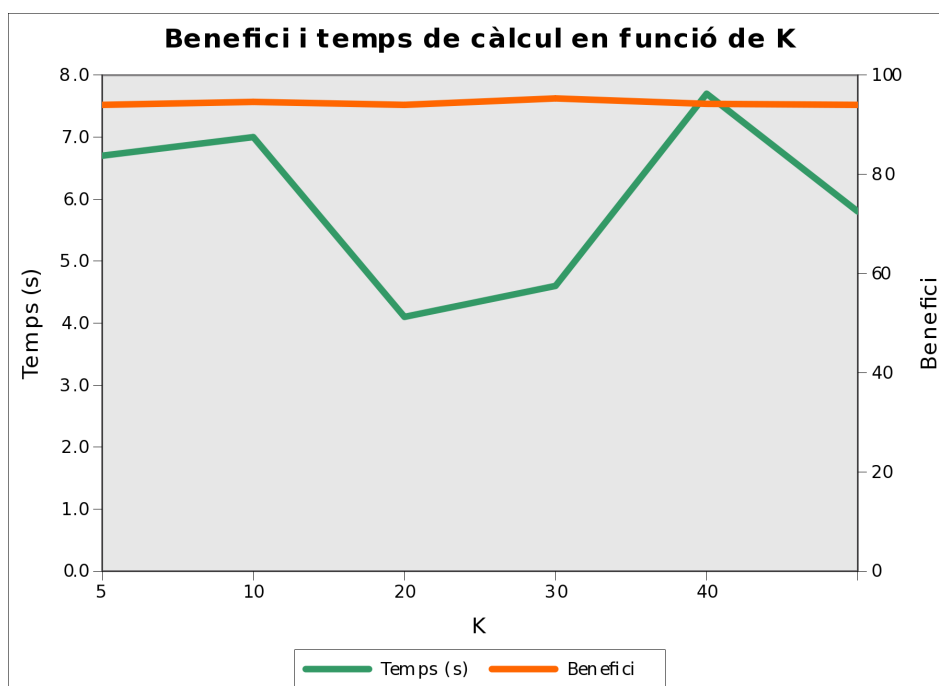


Es pot veure com el paràmetre no afecta pràcticament al problema. Per aquest motiu deixarem el seu valor per defecte que és 100.

Ara calculem el paràmetre lambda. Veiem que el benefici és igual per als valors proposats i que només varia el temps. Això probablement sigui degut a l'elevat nombre d'iteracions que provoca que es trobi sempre una solució bona tot i que els paràmetres facin de vegades empitjorar la solució. Triarem el valor de 0.005 per tenir menor temps i un benefici igual a la resta.



Finalment el valor K. Novament ens decantem pel temps de procés ja que el benefici és quasi igual per tots els valors.



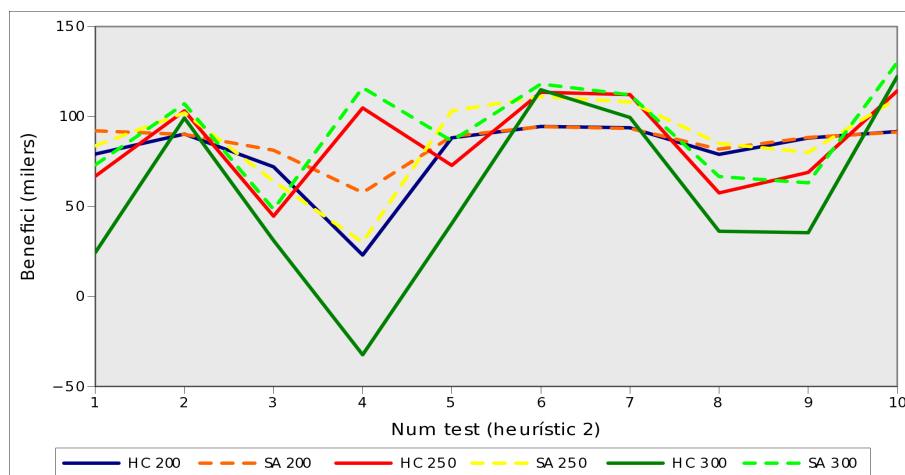
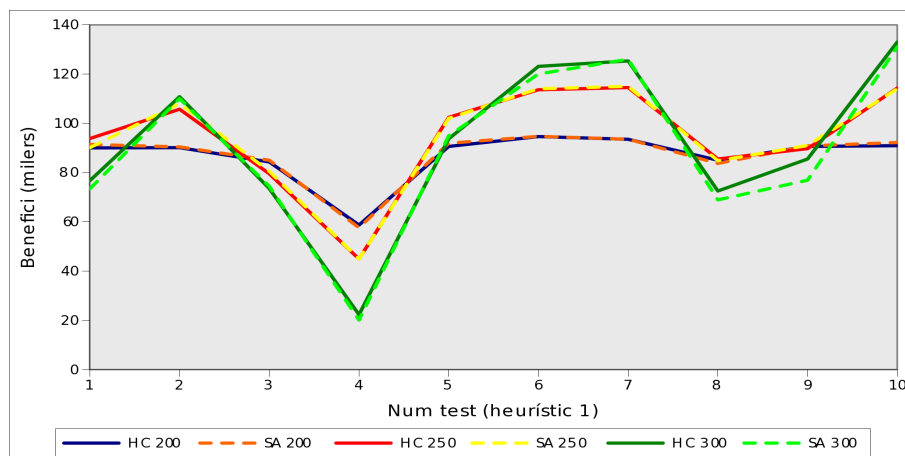
El que està clar és que el nostre algorisme és molt independent dels paràmetres. Sabent que els paràmetres del S.A. controlen la velocitat de baixada així com la probabilitat de pujar i baixar només ens porta a deduir una cosa: el nostre algorisme té uns operadors que generen moltes més millores de la solució que empitjoraments.

Només cal veure com vam definir el conjunt d'operadors 2 per comprovar que els canvis negatius són molt més petits que els positius (sobretot l'operador GIR, que provoca grans pujades de qualitat).

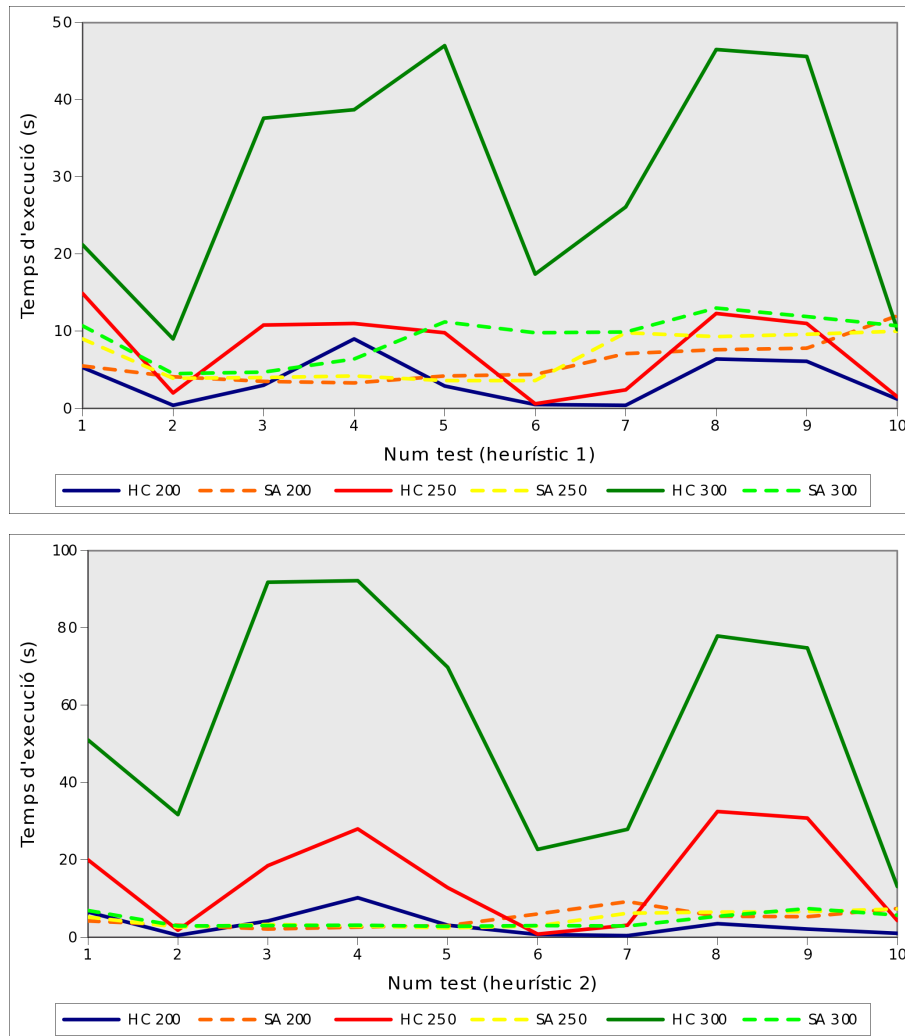
Comparació entre Hill Climbing i Simulated Annealing

Aquest experiment (número 6) compara el benefici i el temps d'execució dels dos algorismes principals del programa. A més ho fa en diferents casos (3 quantitats d'entregues) i per a les dos heurístiques. D'aquesta manera si existeixen diferències significatives les podrem detectar, alhora que veure si el tipus d'heurística afecta els algorismes.

Les hipòtesis que formulem són que Simulated Annealing té un temps menor que Hill Climbing ja que fixem un nombre d'iteracions màxim (i per tant un temps d'execució màxim). També creiem que SA tindrà un menor benefici que HC, ja que processa menys temps. Això últim dependrà de si el nombre d'entregues és molt gran. Per a un nombre d'entregues petit encara SA obtindrà un bon benefici, però per al cas de moltes entregues el nombre d'iteracions de SA serà petit en comparació a l'espai de cerca, cosa que provocarà pitjors resultats que HC.



Es veu com el benefici que s'obté amb els dos algorismes és quasi idèntic per al cas de maximització del benefici. Cosa que no passa en la minimització del retard. Això és degut a dues coses: la primera i més òbvia és que el gràfic no mostra el retard (només el benefici) i, tot i que estan relacionats, no són directament proporcionals. La segona cosa és que la funció heurística 1 és millor que la dos per la raó que té en compte el retard a més del benefici.



Veient els gràfics del temps de procés confirmem la hipòtesi que el temps de SA és constant mentre que HC segueix una llei quadràtica (vegeu l'experiment sobre temporització més endavant).

Influència de la capacitat dels camions

Hem realitzat 3 proves tal com està indicat a l'apartat 7 de l'enunciat que consisteixen en 15 camions de cada tipus a excepció d'un tipus de camions que en tenim 30. D'aquesta manera la capacitat efectiva no és la mateixa però no hi ha una diferència molt gran. Les capacitats són 60.000, 67.500 i 82.500.

De fet entre el primer i segon escenari la diferència de capacitat és només de 7500kg, en canvi el benefici és de l'ordre de 25.000€. Sabent que en el millor dels casos aquests 7500kg arribaran a l'hora i seran paquets dels més cars no arribem al benefici que indiquen els experiments. Això ens ve a dir que el fet de tenir camions més grossos és molt beneficiós més enllà de la capacitat

de càrrega extra que obtenim, ja que ajuda molt a la repartició de les entregues i evita en la mesura del possible els espais buits.

Si l'espai buit més gran que podem tenir en aquest problema és 400 aquest representa un 80% d'espai malgastat, mentre que en un camió molt més gran (2000kg) representa només el 20% de la capacitat. La idea és molt intuïtiva: més val un camió gran que molts de petits, a la realitat moltes vegades és així.

Influència de la distribució del treball

La distribució del treball fa variar molt la qualitat de la solució. Això és degut a les restriccions que imposa el problema. En primer lloc la penalització per retard fa que si la feina es concentra a primeres hores el benefici sigui petit, ja que hauríem de repartir totes les entregues en molt poc temps, cosa impossible. En canvi si les entregues estan al final tenim molta més llibertat en l'hora d'entrega, pel que el benefici és més gran.

Es podria pensar que si les entregues són a les últimes hores el benefici serà molt gran ja que totes arribaran a temps, però el que fa que la diferencia no sigui gaire gran és el fet que la penalització és proporcional al retard i això és un gran avantatge. Si la penalització fos molt més gran per retard (per exemple penalització del 100% quan passa una hora) aleshores veuríem una diferència dramàtica entre les primeres hores i les últimes.

Això és precisament el que hem fet a la segona taula: hem experimentat canviant les condicions de l'enunciat dient que si una entrega arriba tard no té benefici. Així veiem que la primera solució és molt dolenta comparada amb la situació on les últimes hores tenen molta demanda.

I. Annexos

Temps de l'algorisme en funció del nombre d'entregues

	Temps (segons)
100	0.16
150	0.72
200	4.85
250	17.04
300	32.3
350	48.62
400	62.26
450	85.04
500	105.85

Temps i benefici final de S.A. segons el nombre d'iteracions

	Temps (segons)	Benefici (milers)
50.000	4	93.0
100.000	11	92.4
200.000	18	91.9
300.000	29	91.9
400.000	57	92.3

Temps i benefici final de S.A. segons el paràmetre stiter

	Temps (segons)	Benefici (milers)
50	6.3	85.2
75	9.5	84.1
100	9.4	85.5
125	10.9	85.5
150	9.4	85.9
175	9.4	85.2
200	9.7	83.3

Temps i benefici final de S.A. segons el paràmetre lambda

	Temps (segons)	Benefici (milers)
0.001	6.6	94.6
0.0025	4.2	94.3
0.005	4.6	94.0
0.01	5.2	95.3
0.02	5.3	94.2
0.03	5.4	94.8

Temps i benefici final de S.A. segons el paràmetre K

	Temps (segons)	Benefici (milers)
5	6.7	94.0
10	7.0	94.6
20	4.1	94.3
30	4.6	94.8
40	7.7	93.6
50	5.2	94.3

Temps, benefici i retard per al cas de 200 entregues

	Benefici final										
Max. benef.	90.0	90.1	84.3	58.7	90.6	94.6	93.5	85.0	90.6	90.9	87.2
	82.2	85.4	92.8	97.2	87.3	90.8	94.3	89.6	95.9	55.2	
	79.3	90.5	91.0	100.5	88.6	94.6	94.4	61.1	82.9	95.1	
Min. retard	79.1	90.2	72.0	23.0	88.1	94.4	93.7	78.9	88.0	91.6	85.1
	74.7	85.4	92.5	96.0	85.2	90.3	94.6	89.3	96.0	26.9	
	51.9	90.0	91.5	99.6	87.3	91.5	94.8	43.0	71.1	95.1	
	Hores de retard										
Max. benef.	233	20	281	1162	107	29	9	200	46	48	243
	248	83	80	88	91	64	54	31	73	1360	
	637	103	74	113	237	33	128	1316	267	82	
Min. retard	177	14	194	619	68	28	0	151	56	18	154
	192	77	80	78	86	62	24	16	43	537	
	369	78	44	115	188	65	72	581	248	82	
	Temps d'execució										
Max. benef.	5.3	0.4	3.0	9.0	2.9	0.5	0.4	6.4	6.1	1.2	4.87
	9.1	3.7	4.7	1.1	5.3	1.4	1.8	0.0	1.8	19.3	
	1.6	1.7	2.7	6.1	4.8	3.4	5.7	25.9	10.0	0.9	
Min. retard	6.5	0.5	4.2	10.2	3.1	0.7	0.4	3.5	2.1	1.0	5.35
	3.8	1.3	1.4	0.4	1.5	0.6	0.9	0.0	1.0	30.1	
	19.2	2.2	2.3	5.1	5.0	2.3	4.3	34.1	12.2	0.7	

Temps, benefici i retard per al cas de 300 entregues

	Benefici final										
Max. benef.	76.7	110.8	73.5	22.3	93.7	123.1	125.3	72.5	85.5	133.0	94.2
	70.0	105.3	93.1	120.0	91.7	118.2	115.6	137.3	122.6	22.0	
	44.5	104.3	124.2	118.0	77.6	120.5	117.2	14.1	61.2	132.7	
Min. retard	24.4	99.1	31.1	-32.4	40.5	114.8	99.4	36.2	35.4	122.2	63.1
	12.6	70.8	60.9	99.4	60.2	100.0	96.0	137.6	107.9	-35.3	
	-13.0	71.2	113.2	80.6	39.9	109.7	104.3	-35.6	22.8	119.4	
	Hores de retard										
Max. benef.	2329	1030	2466	3373	1924	493	769	2365	2226	251	1651
	2548	1456	1752	980	1556	927	952	132	924	3506	
	3128	1640	418	1329	2457	741	1113	3508	2779	478	
Min. retard	1078	538	1047	1655	950	348	448	1049	1032	226	785
	1162	739	822	514	662	397	433	97	439	1651	
	1421	825	365	569	978	451	466	1790	1094	309	
	Temps d'execució										
Max. benef.	21.2	9.0	37.6	38.7	47.0	17.4	26.1	46.5	45.6	10.2	28.7
	44.4	30.8	25.9	19.1	34.6	25.4	20.8	0.3	24.2	41.9	
	13.8	41.4	43.4	25.9	22.8	41.0	29.5	14.4	26.6	38.0	
Min. retard	51.0	31.7	91.8	92.2	69.8	22.7	27.9	77.9	74.8	13.1	54.7
	86.9	44.8	47.2	32.5	63.0	40.5	35.5	0.3	39.0	111.1	
	106.9	52.5	16.8	42.0	72.3	25.7	45.9	109.3	96.6	19.1	

Resultats de Sim. Ann. per a 200, 250 i 300 entregues

		Benefici final										
Max. benefici	200	91.4	90.4	85.0	57.6	91.9	94.7	93.5	83.7	90.8	92.2	87.1
	250	90.1	108	80.5	44.9	102	114	115	84.8	91.0	114	94.4
	300	73.4	110	74.4	20.1	94.9	120	126	68.9	76.9	131	89.5
Min. retard	200	92.0	90.0	81.2	57.7	88.9	94.3	93.3	81.8	88.3	91.3	85.8
	250	83.7	102	64.0	30.1	103	111	108	85.0	80.0	110	87.6
	300	73.1	107	48.4	116	86.4	118	112	66.6	63.1	130	92.0
		Temps d'execució										
Max. benefici	200	5.5	4.1	3.5	3.3	4.2	4.4	7.1	7.6	7.8	12.0	5.9
	250	9.0	3.9	4.0	4.2	3.6	3.6	9.8	9.3	9.6	10.0	6.7
	300	10.7	4.5	4.7	6.4	11.2	9.8	9.9	13.0	11.9	10.7	9.3
Min. retard	200	4.2	3.1	2.1	2.6	3.0	6.0	9.2	5.4	5.3	7.4	4.8
	250	5.2	2.6	3.3	2.9	2.4	2.9	6.2	6.5	6.6	7.3	4.6
	300	6.9	2.9	3.0	3.1	2.8	3.0	2.9	5.4	7.4	5.7	4.3

Influència de la capacitat dels camions (cas 50% 25% 25%)

	Benefici final					
	57.0	63.7	63.8	63.7	55.2	60.7

Influència de la capacitat dels camions (cas 25% 50% 25%)

	Benefici final					
	82.7	87.0	87.0	88.2	82.0	85.4

Influència de la capacitat dels camions (cas 25% 25% 50%)

	Benefici final					
	113.6	105.9	110.0	109.5	117.3	111.3

Influència de la distribució del treball

	Temps d'execució										
8..11h	49.0	102.5	38.0	2.1	77.9	109.3	107.8	58.2	62.4	113.3	70.1
11..14h	57.6	109.3	50.0	15.8	86.2	112.3	115.5	64.2	70.0	115.3	79.6
14..17h	69.5	111.9	55.4	24.4	90.9	114.5	118.2	69.5	72.1	116.6	84.3

Influència de la distribució del treball (modificació enunciat)

	Temps d'execució										
8..11h	101.3	45.6	44.3	82.1	91.2	64.9	-3.9	27.4	85.2	35.4	57.4
11..14h	50.8	99.2	41.3	1.5	78.5	101.5	108.4	54.7	57.3	108.4	70.1
14..17h	61.1	107.7	40.2	16.8	85.8	111.8	116.3	66.2	63.8	115.2	78.5