

Ús d'algoritmes genètics per a la gestió de la ubicació i assignació de pacients d'una unitat d'hospitalització

Daniel Miró Pettican
Grau d'Enginyeria Informàtica
Intel·ligència artificial

Consultor: Dr. David Isern Alarcón
Professor responsable de l'assignatura: Dr. Carles Ventura Royo

3 de Gener de 2019

Copyright © 2018 Daniel Miró Pettican

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

[A copy of the license is included in the section entitled "GNU Free Documentation License".](#)

Agraïments

Al meu consultor de TFG, David Isern Alarcón, pel temps, dedicació i bons consells.

A la meva dona Maria Eugènia per estar sempre al meu costat durant tota aquesta aventura que és la vida.

Als meus fills, Àngel i Neus, per ser llum i alegria... i aprenentatge.

Als meus companys, Sílvia, Xisco i Mar per les rialles i els bons consells.

I a tot persona que d'una manera o una altra m'ha ajudat a arribar fins aquí.

Moltes gràcies!

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Ús d'algoritmes genètics per a la gestió de la ubicació i assignació de pacients d'una unitat d'hospitalització</i>
Nom de l'autor:	<i>Daniel Miró Pettican</i>
Nom del consultor/a:	<i>Dr. David Isern Alarcón</i>
Nom del PRA:	<i>Dr. Carles Ventura Royo</i>
Data de lliurament (mm/aaaa):	<i>01/2019</i>
Titulació o programa:	<i>Grau d'Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Intel·ligència Artificial</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Algoritmes genètics, gestió de pacients, Infermeria</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>La gestió dels pacients de les unitats d'hospitalització és una de les tasques de la gestió hospitalària que, avui en dia, podria beneficiar-se de l'ús d'algoritmes d'intel·ligència artificial amb l'objectiu de millorar la seva eficiència.</p> <p>Aquest TFG explora la implementació i ús d'un algoritme genètic que, donada una distribució física dels pacients en una unitat i la determinació de la seva càrrega de cures, suggereix canvis d'ubicació per intentar millorar l'assignació dels pacients als professionals de la Unitat. Per dur-ho a terme, es basa en l'assignació suggerida per un altre algoritme genètic, el qual proporciona assignacions de pacients per als diferents professionals de manera que les càrregues queden equilibrades, intentat que les distàncies a recórrer per atendre a tots els pacients siguin les més reduïdes possibles.</p> <p>Els resultats de l'aplicació dels algoritmes mostren que l'algoritme d'assignació demostra un comportament correcte a l'hora d'assignar pacients. No obstant això, a l'hora de realitzar canvis d'ubicació l'algoritme no respon de forma favorable i, per tant, no és factible el seu ús.</p> <p>En conclusió, els algoritmes genètics poden ser una eina útil a l'hora de gestionar certes qüestions relacionades amb la gestió hospitalària, però cal avaluar altres opcions per trobar eines específiques a l'hora de tractar amb altres tasques.</p>	

Abstract (in English, 250 words or less):

The management of patients in hospitalization units is one of the tasks of hospital management that, nowadays, could benefit from the use of artificial intelligence algorithms with the aim of improving their efficiency.

This TFG explores the implementation and use of a genetic algorithm that, given a physical distribution of patients in a unit and the determination of their load of care tasks, suggests changes of location trying to improve patient allocation to the unit professionals. For that, it is based on the allocation suggested by another genetic algorithm, which provides patient assignments for the different professionals so that care tasks charges are balanced, trying that distances to cover to attend all the patients are as reduced as they can possibly be.

The results of the application of the algorithms show that the allocation algorithm demonstrates a correct behavior when assigning patients. However, when making location changes the algorithm does not respond favorably and, therefore, its use is not feasible.

In conclusion, genetic algorithms can be an useful tool when managing certain issues related to hospital management, but other options must be evaluated to find specific tools when it comes to manage certain tasks.

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball.....	1
1.2 Objectius del Treball.....	3
1.3 Enfocament i mètode seguit.....	4
1.4 Planificació del Treball.....	4
1.5 Breu sumari de productes obtinguts.....	10
1.6 Breu descripció dels altres capítols de la memòria.....	10
2. Elecció de l'entorn de desenvolupament.....	11
3. Impacte de la gestió dels pacients.....	13
3.1. Espai físic.....	15
3.2. Càrrega de cures dels pacients.....	20
4. Codificació de l'algoritme.....	21
4.1. Algoritme d'assignació.....	21
4.1.1 Codificació i generació dels cromosomes.....	21
4.1.2. Avaluació dels cromosomes.....	22
4.1.3. Selecció dels millors individus.....	24
4.1.4. Mutacions i encreuaments.....	24
4.2. Algoritme d'ubicació.....	26
4.2.1. Codificació i generació dels cromosomes.....	26
4.2.2. Avaluació dels cromosomes.....	26
4.2.3. Selecció dels millors individus.....	26
4.2.4. Mutacions i encreuaments.....	26
5. Resultats.....	28
5.1. Algoritme d'assignació.....	28
5.1.1. Validació dels paràmetres de la funció de fitting.....	28
5.1.2. Comportament de l'algoritme.....	31
5.1.3. Aplicació de l'algoritme en un cas real.....	43
5.2. Algoritme d'ubicació.....	48
5.2.1. Comportament de l'algoritme.....	48
5.2.2. Aplicació de l'algoritme en un cas real.....	55
6. Conclusions.....	58
6.1. Conclusions del treball.....	58
6.2. Assoliment dels objectius plantejats.....	58
6.3. Seguiment de la planificació.....	59
6.4. Futures línies de treball.....	59
7. Glossari.....	61
8. Bibliografia.....	62
9. Annexos.....	63
A. Manual d'instal·lació.....	63
B. Paràmetres d'ús.....	64

Llista de figures

Il·lustració 1: funció de fitting.....	2
Il·lustració 2: cicle evolutiu d'un algoritme genètic.....	3
Il·lustració 3: nou pacient a la unitat.....	13
Il·lustració 4: nou pacient i canvis d'ubicació.....	14
Il·lustració 5: unitat amb control central.....	15
Il·lustració 6: unitats d'hospitalització en forma d'L (Wards A, B & C).....	16
Il·lustració 7: unitat d'un únic passadís.....	16
Il·lustració 8: esquema d'una unitat d'hospitalització d'adults de l'HUSE.....	17
Il·lustració 9: distància euclidiana vs distància Manhattan.....	18
Il·lustració 10: hotmap de les distàncies de les habitacions.....	19
Il·lustració 11: evolució del valor mínim de l'algoritme d'assignació.....	43
Il·lustració 12: evolució de la mediana de l'algoritme d'assignació.....	43
Il·lustració 13: representació gràfica de les associacions del cas real.....	44
Il·lustració 14: hotmap de les càrregues.....	44
Il·lustració 15: evolució del valor mínim de l'algoritme d'assignació post-ajusts.....	45
Il·lustració 16: evolució de la mediana dels valors de l'algoritme d'assignació post-ajusts.....	45
Il·lustració 17: evolució de la mitja dels valors de l'algoritme d'assignació post-ajusts.....	45
Il·lustració 18: evolució de la suma dels valors de l'algoritme d'assignació post-ajusts.....	45
Il·lustració 19: evolució dels valors màxims de l'algoritme d'assignació post-ajusts.....	46
Il·lustració 20: representació gràfica de les associacions del cas real post-ajusts.....	47
Il·lustració 21: hotmap de les càrregues després de la nova ubicació.....	55
Il·lustració 22: assignació de pacients després de la nova ubicació.....	56

1. Introducció

1.1 Context i justificació del Treball

La gestió de la Sanitat és sempre un punt que apareix a tot programa electoral i, encara que sempre existeixen aspectes conflictius al seu voltant, és imprescindible considerar-la una part important de la gestió política. Dintre de les diferents estructures de la Sanitat Pública, es pot entendre als hospitals com les estructures amb més visibilitat i, per tant, els objectius derivats de la seva gestió poden ser indicadors de la "bona salut" de les diferents decisions i polítiques sanitàries.

Multitud de persones amb diferents perfils són responsables de la gestió hospitalària, entre les quals es poden trobar a les supervidores d'infermeria de les diferents unitats d'hospitalització. Aquest tipus de gestió no arriba a tenir massa visibilitat, però és una de les àrees fonamentals de la gestió hospitalària atès que una bona gestió de les unitats implica uns fluxos de treball més eficients, reduint costos i millorant la seguretat.

Entre altres funcions, les supervidores de les unitats es responsabilitzen de la gestió de la ubicació dels pacients dintre de la unitat una vegada es rep l'ordre d'ingrés d'un pacient a la unitat per part del Servei d'Admissió de l'Hospital. Així, la seva ubicació es decideix seguint criteris com l'estat general del pacient, les necessitats específiques de tractament, el sexe i la resta de pacients de la unitat.

Amb tot, la gestió habitual es limita a ubicar al pacient en un llit disponible sense avaluar quina mena d'impacte tindrà. En efecte, aquesta ubicació pot ser determinant a l'hora d'assignar pacients als diferents professionals i no és estrany que, amb l'objectiu d'obtenir una assignació equilibrada dels recursos humans, un professional tingui una assignació de pacients que impliqui més desplaçaments dels necessaris. Ara bé, encara que aquest tipus de gestió no es podria considerar com a desitjable, és cert que gestionar l'impacte d'aquest tipus de decisions no és senzill i necessita algun tipus de tractament informàtic.

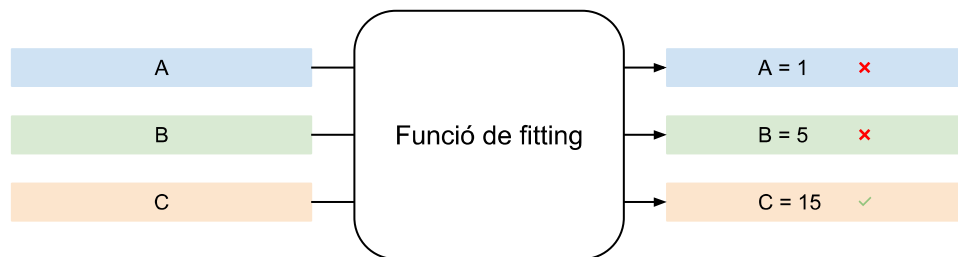
Amb referència a l'ús d'eines informàtiques per tractar problemes relacionats amb la gestió, es poden trobar diferents casos d'aplicació d'algoritmes d'intel·ligència artificial orientats a la gestió dels recursos humans, més específicament, a la gestió dels torns del personal d'infermeria. La dificultat d'aquesta tasca ha estat formulada com un problema NP-difícil [WikiNSP], generant articles al seu voltant orientats a la cerca de solucions mitjançant l'ús d'algoritmes genètics.[Chang09][Moz07][Kom14] Per altra banda, dintre de les tasques de gestió, la ubicació i programació dels pacients també ha estat la base per l'aplicació d'aquest tipus d'algoritmes.[Aza14][Vili97]

Cal fer una especial referència a aquest tipus d'algoritmes que destaquen per oferir una solució suficientment bona a un problema

d'optimització i recerca dintre d'una quantitat raonable de temps. En concret, un algoritme genètic és un tipus d'algoritme que s'inspira en el procés de selecció natural i utilitza mecanismes com la reproducció, mutació, recombinació genètica i selecció.

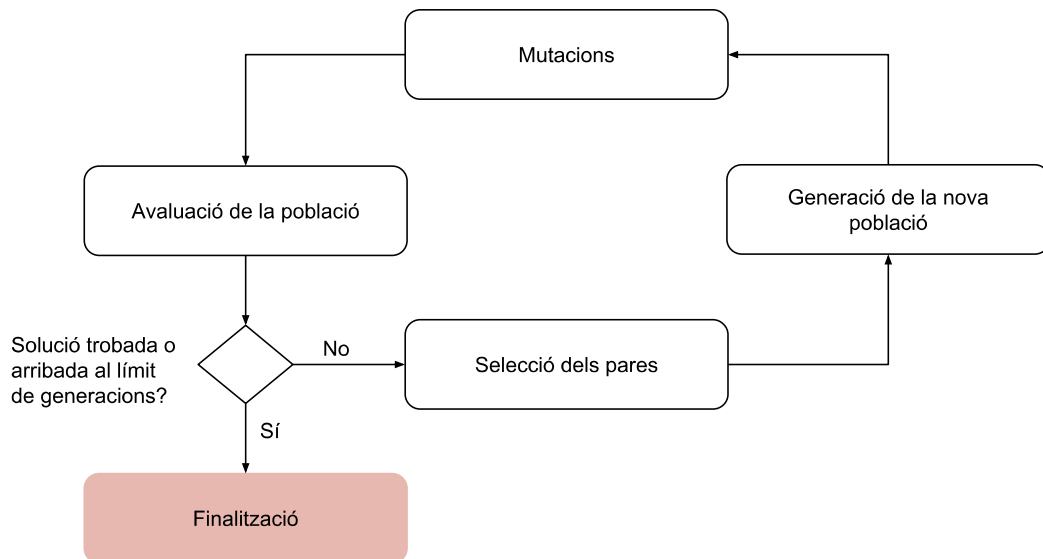
Els algoritmes genètics parteixen de la premissa que una possible solució al problema es pot representar i codificar per un conjunt de paràmetres o gens denominat cromosoma. D'aquesta manera, només cal cercar-la dintre d'un conjunt de solucions que es denomina població.

Per poder determinar la validesa de la solució, cada cromosoma té un valor definit per una funció d'aptitud o fitness. Així, com millor valor de fitness té un cromosoma, millor predictor com a possible solució al problema.



Il·lustració 1: funció de fitting

La base dels algoritmes genètics són els cicles evolutius pels quals els millors membres de la població actual, és a dir, els que tenen un millor valor per a la funció de fitness, tenen més possibilitats de ser escollits com a pares de la següent generació. Així, els seus gens es combinen per generar nous cromosomes per un procediment d'encreuament. A més a més, de manera semblant al procediment de reproducció natural, poden existir "errades" o mutacions que modifiquen alguns gens de la nova generació que poden traduir-se en un millor valor de fitness. Finalment, s'avalua aquesta nova generació i el cicle evolutiu torna a començar fins que es troba un individu que doni la millor solució o fins que s'arriba a un nombre determinat de generacions.[Man99]



Il·lustració 2: cicle evolutiu d'un algoritme genètic

Així i tot, la gestió de la ubicació dels pacients sembla no haver servit de base per a la investigació relacionada amb algoritmes genètics per intentar trobar solucions plausibles.

1.2 Objectius del Treball

Aquest TFG neix del plantejament de l'existència de formes més eficients d'ubicar als pacients d'una unitat per millorar els temps d'atenció d'aquests per part dels professionals de la unitat.

Per això es defineixen els següents objectius principals:

1. Creació d'un algoritme genètic per a la gestió de la ubicació dels pacients que, donada una configuració inicial, generi una nova ubicació d'aquests minimitzant la diferència de càrregues de cures segons l'assignació per a cada professional.
2. Creació d'un algoritme genètic per a l'assignació de pacients que, donada una configuració inicial de pacients, determini quina és l'assignació més equilibrada i compacte de pacients per als diferents professionals segons la càrrega de cures i la seva ubicació.

Com a objectius secundaris es defineixen:

3. Avaluar la codificació de restriccions addicionals als algoritmes, com ara el sexe dels pacients i els aïllaments
4. Avaluar l'ús de computació en paral·lel amb diferents fils d'execució per millorar els temps dels algoritmes.

1.3 Enfocament i mètode seguit

Atès que avui en dia no sembla existir cap mena d'algoritme orientat a la ubicació dels pacients des del punt de vista de la càrrega de cures, no és possible adaptar cap producte existent i, per tant, només queda desenvolupar l'algoritme com un producte nou.

Per intentar obtenir un model el més real possible, la font del set de dades d'inici s'obtindrà dels registres de la història clínica electrònica de Cerner (Millennium), que s'utilitza en l'actualitat a l'Hospital Universitari Son Espases de Palma de Mallorca.

1.4 Planificació del Treball

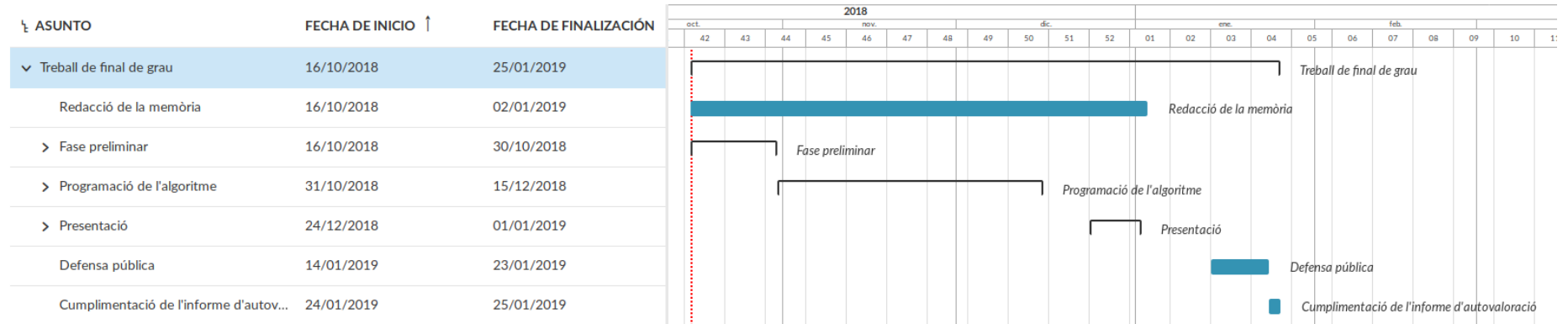
Recursos materials

Com a recursos materials, es té disponible un ordinador personal Intel Core i5-4430 de 4 cores amb 8 GiB de memòria RAM.

Recursos humans

En un principi, l'únic recurs humà del qual disposem és el mateix autor del treball de final de grau. Encara així, és possible que sigui necessari comptar amb recursos externs per poder extreure les dades inicials dels sistemes d'informació.

Planificació general



Planificació fase preliminar



Planificació de programació

ASUNTO	FECHA DE INICIO ↑	FECHA DE FINALIZACIÓN	2018												
			oct.			nov.					dic.			ene.	
			42	43	44	45	46	47	48	49	50	51	52	01	02
Programació de l'algoritme	31/10/2018	15/12/2018													
Programació de l'algoritme de mill...	31/10/2018	17/11/2018													
Estructura bàsica del programa	31/10/2018	01/11/2018													
Creador de generacions	02/11/2018	04/11/2018													
Funció de fitting	05/11/2018	07/11/2018													
Modificacions genètiques	08/11/2018	10/11/2018													
Finalització de la programació	11/11/2018	13/11/2018													
Testing	14/11/2018	17/11/2018													
Programació de l'algoritme de mill...	19/11/2018	06/12/2018													
Estructura bàsica del programa	19/11/2018	20/11/2018													
Creador de generacions	21/11/2018	23/11/2018													
Funció de fitting	24/11/2018	26/11/2018													
Modificacions genètiques	27/11/2018	29/11/2018													
Finalització de la programació	30/11/2018	02/12/2018													
Testing	03/12/2018	06/12/2018													
Programació concurrent	08/12/2018	15/12/2018													
Adaptació de l'algoritme	08/12/2018	12/12/2018													
Testing	13/12/2018	15/12/2018													

Planificació de la presentació



Planificació de fites

ASUNTO	FECHA DE INICIO ↑	2018																
		sep.			oct.			nov.				dic.				ene.		
		39	40	41	42	43	44	45	46	47	48	49	50	51	52	01	02	03
Informe comparatiu de plataformes de desenvolupament	26/10/2018					◆												
Set de dades d'entrada	29/10/2018						◆											
Plataforma de desenvolupament	31/10/2018							◆										
Algoritme genètic d'assignació	19/11/2018										◆							
Algoritme genètic d'ubicació	07/12/2018												◆					
Algoritme concurrent	16/12/2018														◆			
Presentació	02/01/2019																	◆

1.5 Breu sumari de productes obtinguts

Resultat del TFG s'obté la present memòria i un paquet de programari que es lliura com un fitxer comprimit. Dintre d'aquest arxiu es pot trobar:

- Codi font Python de la implementació dels algoritmes genètics
- Tots els sets de proves utilitzats

1.6 Breu descripció dels altres capítols de la memòria

La memòria del projecte presenta 9 capítols en total on, en primer lloc, s'exposa el marc contextual del projecte, així com els objectius, enfocament i planificació.

A continuació, el segon capítol explica quins han estat els criteris pels quals s'ha escollit l'entorn de desenvolupament del TFG.

El tercer capítol comenta tots els aspectes a tenir en compte a l'hora d'avaluar l'impacte de la ubicació dels pacients, és a dir, el càlcul de les distàncies entre les ubicacions físiques i la determinació de la càrrega de cures dels pacients.

Al quart capítol es parla del disseny dels diferents algoritmes codificats, així com les decisions de disseny i els mecanismes d'avaluació, reproducció, mutació i encreuament.

Al cinquè capítol es mostren les diferents proves executades per comprovar l'execució dels algoritmes, així com la seva viabilitat.

Seguidament, al sisè capítol, es mostren les conclusions del treball i es comenten possibles línies de treball.

El setè capítol inclou un petit glossari amb termes i conceptes que poden necessitar algun tipus d'aclariment.

El vuitè capítol mostra les diferents referències bibliogràfiques i, finalment, el novè capítol contempla com a annex un petit manual d'usuari del codi.

2. Elecció de l'entorn de desenvolupament

A l'hora de decidir el llenguatge de programació i l'entorn de desenvolupament, s'estableixen diferents criteris d'avaluació. Així, amb l'objectiu d'obtenir un codi amb la possibilitat d'ampliar les seves funcionalitats de forma senzilla, s'opta per un model de programació orientat a objectes. A més a més, per facilitar el seu ús a diferents plataformes, es dóna prioritat a llenguatges de programació que siguin multiplataforma. Així docs, es determina que el llenguatge de programació ha de complir:

- Ha de permetre la programació orientada a objectes
- Ha de ser un llenguatge conegut per l'usuari
- Ha de permetre la concurrència
- És desitjable que tingui compiladors i/o intèrprets multiplataforma
- És desitjable que disposi de paquets o llibreries de fàcil instal·lació i ús si aquestes són necessàries per a una correcta execució de l'algoritme

Seguint els criteris definits, els dos llenguatges que compleixen amb tots els criteris són Java i Python.

En el marc dels entorns de desenvolupament, tant per Java com per Python es troben diferents entorns. Així com s'han definit diferents criteris que els llenguatges de programació a considerar han de complir, també es formulen les següents característiques que van orientades a facilitar la tasca de codificació:

- Ha de ser un entorn conegut per l'usuari
- Ha de permetre dur a terme tasques de depuració
- Ha de permetre control de versions Git
- Compatible amb els llenguatges de programació d'elecció
- Gratuït o amb llicència gratuïta per estudiants
- Compatible amb Linux

Després de dur a terme una cerca dels entorns de desenvolupament per Java i Python, es duu a terme una comparativa seguint els criteris ja esmentats.

	NetBeans	Eclipse	Jedit	Visual Studio Code	PyCharm
Llenguatge	Java Python ¹	Java Python ²	Java Python ³	Java ⁴ Python ⁵	Python
Funcions de depuració	Sí	Sí	Sí ⁶	Sí	Sí
Git	Sí	Sí ⁷	Sí ⁸	Sí	Sí
Llicència	Gratuït	Gratuït	Gratuït	Gratuït	Community edition ⁹
S.O.	Linux Windows Mac OS	Linux Windows Mac OS	Linux Windows Mac OS	Linux Windows Mac Os	Linux Windows Mac Os
Conegut per l'usuari	No	Sí	No	No	Sí

Ateses les característiques dels diferents entorns, se selecciona Python com a llenguatge de programació en la seva versió 3.6 i PyCharm 2018.2.4 (Community Edition) per dur a terme les tasques de desenvolupament.

1 Només v 8.2 via plugin

2 Via PyDev

3 Via Jython

4 Via plugins

5 Via plugins

6 Python via JPyDebug

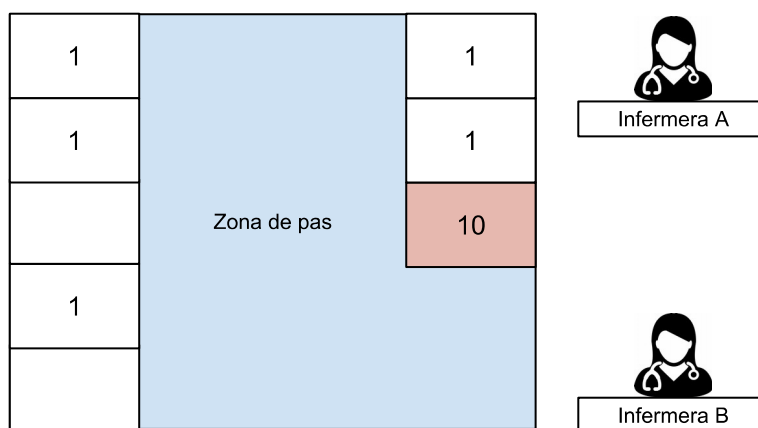
7 Via eGit

8 Via GitPlugin

9 Gratuïta per desenvolupament científic

3. Impacte de la gestió dels pacients

Tal com ja s'ha comentat, ubicar a un pacient dintre d'una unitat té un impacte no trivial dintre de la distribució de la càrrega de cures. Cal remarcar que no només augmenta la càrrega de treball total de la unitat, sinó que pot tenir com a conseqüència que alguns professionals tinguin més càrrega de treball que altres o que, en un intent d'equilibrar-la, hagin de desplaçar-se encara més que la resta. Com a exemple, el següent cas mostra l'impacte de l'ingrés d'un nou pacient amb una alta càrrega de cures (en vermell):



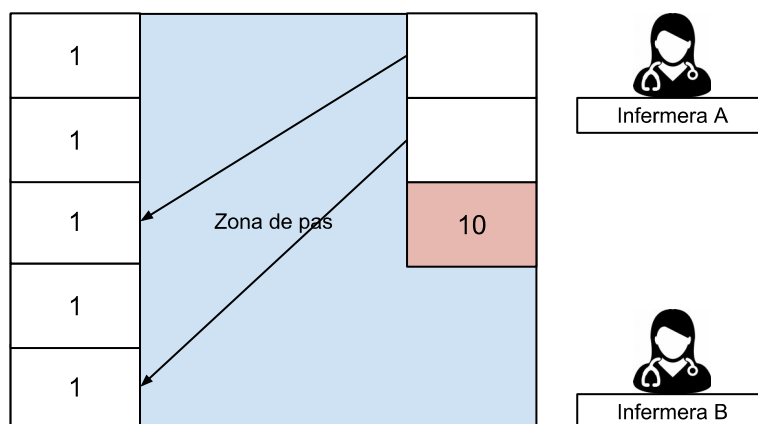
Il·lustració 3: nou pacient a la unitat

En funció de com es dugui a terme l'assignació de pacients per cada infermera, sorgeixen diferents aspectes a tenir en compte:

- Si es determina que cada infermera és responsable d'una zona de la unitat, es crea la situació que la Infermera B té el mateix nombre de pacients que la Infermera A, però la seva càrrega de cures augmenta molt en comparació ($12 \gg 3$).
- Com a conseqüència de l'elevada càrrega de cures del nou pacient, es pot decidir que s'assigni de forma exclusiva a una de les infermeres. Per tant, l'altra s'encarregaria de la resta de pacients, cosa que sembla ser més equitativa ($10 > 5$), però implica un gran nombre de desplaçaments i distància recorreguda per poder atendre a tots els pacients.

Per aquest fet, la ubicació no sempre és una qüestió fàcil a resoldre, sobretot si es té en compte que les unitats d'hospitalització poden ubicar un nombre de pacients força important, amb càrregues de cures heterogènies i un nombre variant de professionals.

Ara bé, també convé destacar que, a l'hora de gestionar la ubicació dels pacients, no només cal tenir en compte la ubicació dels nous pacients, sinó que hi ha la possibilitat de fer canvis amb la resta de pacients. Seguint amb l'exemple anterior, a més d'ingressar al pacient, s'executen canvis addicionals:



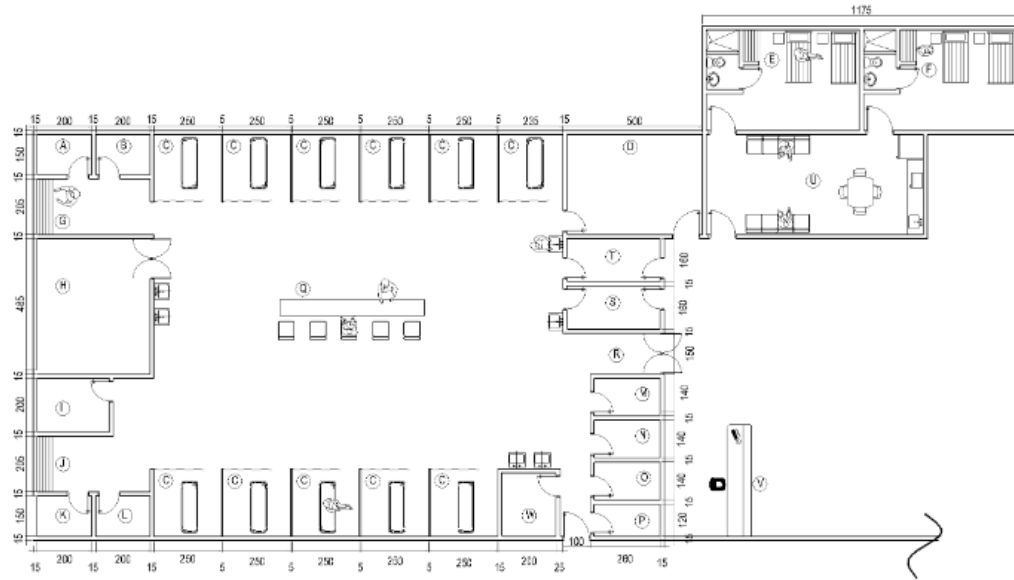
Il·lustració 4: nou pacient i canvis d'ubicació

Aquests canvis permeten fer una assignació més equitativa a la qual una de les infermeres es faria responsable del nou pacient i l'altra de la resta. Encara que aquesta situació és semblant al cas anterior, on la càrrega de cures és desigual ($10 > 5$), el fet que els pacients amb menor càrrega estiguin més propers, ajuda que la infermera responsable no hagi de recórrer una distància innecessària.

Així, pel que fa a com avaluar l'impacte de la ubicació del pacient caldrà tenir en compte dos aspectes: l'espai físic a on se situen els llits i la càrrega de cures.

3.1. Espai físic

Els dissenys arquitectònics dels hospitals i la seva eficiència són un problema en si mateix, discutits a diferents documents [Hol15] i guies. De la mateixa manera, les unitats d'hospitalització poden presentar diferents arquitectures en funció de les seves necessitats funcionals.



Il·lustració 5: unitat amb control central



Il·lustració 6: unitats d'hospitalització en forma d'L (Wards A, B & C)



Il·lustració 7: unitat d'un únic passadís

L'arquitectura de la unitat determinarà el cost en distàncies de les possibles assignacions de pacients als professionals. Encara que les distàncies reals entre les habitacions sí que es podrien mesurar de forma exacta fent servir mapes a escala, es considera més adient per l'objectiu del projecte l'ús de distàncies relatives aproximades i, per tant, cal definir una metodologia coherent per poder establir el "cost" de les assignacions.

Com a exemple, s'agafa el disseny d'una planta d'hospitalització d'adults de l'HUSE. El disseny general presenta una estructura en forma de "L" amb 27 habitacions: 10 dobles i 17 individuals per un total de 37 llits. Es pot observar l'esquema següent si se situen les habitacions dins una quadrícula, on les quadrícules de color gris són les zones de pas dels professionals, pacients i familiars i les zones de color vermell corresponen a les diferents àrees de servei (control d'infermeria, despatxos, sales de medicació, vestuaris...):

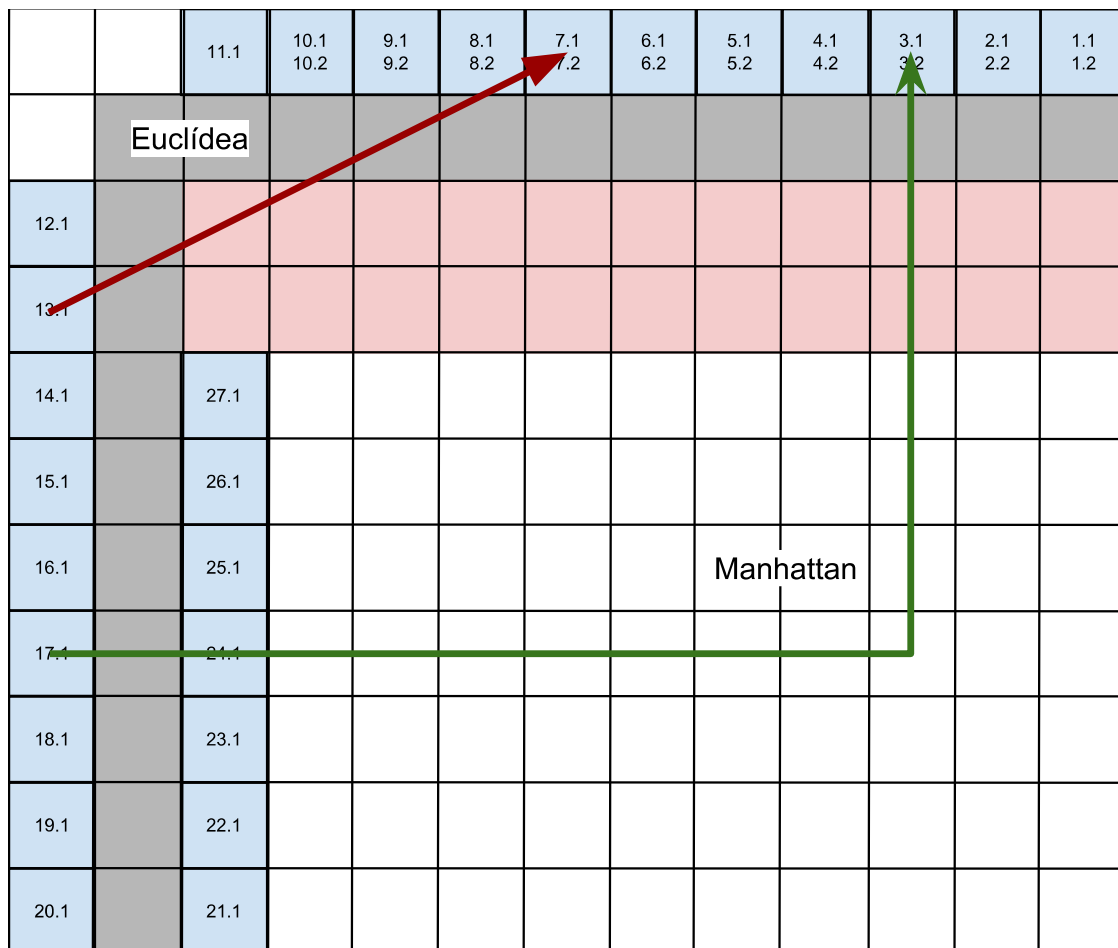
		11.1	10.1 10.2	9.1 9.2	8.1 8.2	7.1 7.2	6.1 6.2	5.1 5.2	4.1 4.2	3.1 3.2	2.1 2.2	1.1 1.2
12.1												
13.1												
14.1		27.1										
15.1		26.1										
16.1		25.1										
17.1		24.1										
18.1		23.1										
19.1		22.1										
20.1		21.1										

Il·lustració 8: esquema d'una unitat d'hospitalització d'adults de l'HUSE

D'aquesta manera es pot codificar la posició d'una habitació segons la columna i fila que ocupa. Si s'assumeix que l'habitació 1 té la posició (0,0) a la quadrícula, l'habitació 2 té la posició (0,1), l'habitació 3 la posició (0,2), etc. Ara bé, com que algunes habitacions tenen més d'un llit, es considera que la seva distància relativa és negligible, atès que, una vegada un professional entra a una habitació, pot atendre als dos pacients sense que això impliqui més desplaçaments addicionals. En conseqüència, algunes posicions queden duplicades per reflectir-ho.

Tenint en compte tot el que s'ha comentat, la unitat d'exemple es pot codificar de la següent manera: ((0,0) (0,0) (0,1) (0,1) (0,2) (0,2) (0,3) (0,3) (0,4) (0,4) (0,5) (0,5) (0,6) (0,6) (0,7) (0,7) (0,8) (0,8) (0,9) (0,9) (0,10) (2,12) (3,12) (4,12) (5,12) (6,12) (7,12) (8,12) (9,12) (10,12) (10,10) (9,10) (8,10) (7,10) (6,10) (5,10) (4,10)).

A l'hora de calcular la distància entre els diferents llits, atès tenim les seves coordenades a un espai de dues dimensions, existeixen diferents mesures disponibles. Les més habituals a l'hora de computar les distàncies són la distància euclidiana i la Manhattan, encara que n'hi ha d'altres.[Lott15]



Il·lustració 9: distància euclidiana vs distància Manhattan

Atès que el valor de la distància només és útil a escala comparativa entre les diferents assignacions, es poden utilitzar les dues, però s'opta per la distància Manhattan gràcies a la seva facilitat de càlcul. La fórmula general per la distància Manhattan és

$$d(M, P) = |M_x - P_x| + |M_y - P_y|$$

Com a exemples:

- Llit 1.1 a 1.2: els dos llits es troben a la mateixa habitació al punt (0,0) i, per tant la distància és 0.

- Llit 1.1 a 2.2: el llit 1.1 es troba al punt (0,12) i el llit 2.2 al punt (0,11). La distància és 1.
- Llit 1.1 a 20.1: el llit 1.1 es troba al punt (0,12) i el llit 20.1 al (10,0). La distància és 22.

Així, es pot generar una matriu de distàncies de mida 37x37 que formarà part de les dades d'entrada de l'algorisme (per simplicitat, només es mostren les distàncies entre habitacions):

Distància	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	0	1	2	3	4	5	6	7	8	9	10	14	15	16	17	18	19	20	21	22	20	19	18	17	16	15	14
2	1	0	1	2	3	4	5	6	7	8	9	13	14	15	16	17	18	19	20	21	19	18	17	16	15	14	13
3	2	1	0	1	2	3	4	5	6	7	8	12	13	14	15	16	17	18	19	20	18	17	16	15	14	13	12
4	3	2	1	0	1	2	3	4	5	6	7	11	12	13	14	15	16	17	18	19	17	16	15	14	13	12	11
5	4	3	2	1	0	1	2	3	4	5	6	10	11	12	13	14	15	16	17	18	16	15	14	13	12	11	10
6	5	4	3	2	1	0	1	2	3	4	5	9	10	11	12	13	14	15	16	17	15	14	13	12	11	10	9
7	6	5	4	3	2	1	0	1	2	3	4	8	9	10	11	12	13	14	15	16	14	13	12	11	10	9	8
8	7	6	5	4	3	2	1	0	1	2	3	7	8	9	10	11	12	13	14	15	13	12	11	10	9	8	7
9	8	7	6	5	4	3	2	1	0	1	2	6	7	8	9	10	11	12	13	14	12	11	10	9	8	7	6
10	9	8	7	6	5	4	3	2	1	0	1	5	6	7	8	9	10	11	12	13	11	10	9	8	7	6	5
11	10	9	8	7	6	5	4	3	2	1	0	4	5	6	7	8	9	10	11	12	10	9	8	7	6	5	4
12	14	13	12	11	10	9	8	7	6	5	4	0	1	2	3	4	5	6	7	8	10	9	8	7	6	5	4
13	15	14	13	12	11	10	9	8	7	6	5	1	0	1	2	3	4	5	6	7	9	8	7	6	5	4	3
14	16	15	14	13	12	11	10	9	8	7	6	2	1	0	1	2	3	4	5	6	8	7	6	5	4	3	2
15	17	16	15	14	13	12	11	10	9	8	7	3	2	1	0	1	2	3	4	5	7	6	5	4	3	2	3
16	18	17	16	15	14	13	12	11	10	9	8	4	3	2	1	0	1	2	3	4	6	5	4	3	2	3	4
17	19	18	17	16	15	14	13	12	11	10	9	5	4	3	2	1	0	1	2	3	5	4	3	2	3	4	5
18	20	19	18	17	16	15	14	13	12	11	10	6	5	4	3	2	1	0	1	2	4	3	2	3	4	5	6
19	21	20	19	18	17	16	15	14	13	12	11	7	6	5	4	3	2	1	0	1	3	2	3	4	5	6	7
20	22	21	20	19	18	17	16	15	14	13	12	8	7	6	5	4	3	2	1	0	2	3	4	5	6	7	8
21	20	19	18	17	16	15	14	13	12	11	10	10	9	8	7	6	5	4	3	2	0	1	2	3	4	5	6
22	19	18	17	16	15	14	13	12	11	10	9	9	8	7	6	5	4	3	2	3	1	0	1	2	3	4	5
23	18	17	16	15	14	13	12	11	10	9	8	8	7	6	5	4	3	2	3	4	2	1	0	1	2	3	4
24	17	16	15	14	13	12	11	10	9	8	7	7	6	5	4	3	2	3	4	5	3	2	1	0	1	2	3
25	16	15	14	13	12	11	10	9	8	7	6	6	5	4	3	2	3	4	5	6	4	3	2	1	0	1	2
26	15	14	13	12	11	10	9	8	7	6	5	5	4	3	2	3	4	5	6	7	5	4	3	2	1	0	1
27	14	13	12	11	10	9	8	7	6	5	4	4	3	2	3	4	5	6	7	8	6	5	4	3	2	1	0

Il·lustració 10: hotmap de les distàncies de les habitacions

3.2. Càrrega de cures dels pacients

Tot pacient hospitalitzat té el que s'anomena una càrrega de cures, és a dir, una certa quantitat de cures, proves i tractaments que impliquen diferents activitats a dur a terme per part dels diferents professionals. La quantificació d'aquest "cost" ha format part de projectes com el projecte SIGNO II [Sig96], però, malauradament, aquests costos se solen determinar a l'alta del pacient. Una altra opció és determinar-ho a partir de resultats a escales d'avaluació [Car13] o, fins i tot, deixar-ho a criteri dels diferents professionals.

En qualsevol cas, sigui quin sigui el mètode per determinar el cost del pacient, l'algoritme presentarà el mateix comportament i, per tant, s'opta per una estimació simple que parteix de la següent suposició: a més tasques pautades per als professionals d'infermeria, més càrrega de cures cal assumir.

A tall d'exemple, es duu a terme una extracció de dades de pacients d'una unitat d'hospitalització i es fa un recompte del nombre de tasques pautades. Aquest recompte genera una codificació on a cada pacient s'assigna el nombre total de tasques (un "0" implica que el llit no està ocupat):

0, 38, 43, 47, 46, 39, 39, 29, 28, 33, 35, 35, 45, 35, 67, 44, 46, 29, 53, 35, 64, 46, 41, 44, 38, 52, 55, 61, 57, 46, 44, 44, 51, 75, 62, 33, 23

4. Codificació de l'algoritme

L'esquema general d'un algoritme genètic és el següent:

```
inicialització de la població
avaluació de la població
selecció del millor cromosoma
mentre no se satisfà el criteri d'acabament fer
    crear una nova població
    avaluació de la població
    selecció del millor cromosoma
fmentre
retorna el millor cromosoma
```

En el cas que ens ocupa, es codifiquen dos algoritmes genètics:

- Un algoritme genètic d'assignació que retorna la millor distribució dels pacients per als diferents professionals
- Un algoritme genètic d'ubicació que retorna una millor distribució dels pacients tenint en compte el valor retornat per l'algoritme anterior

4.1. Algoritme d'assignació

A l'hora de programar l'algoritme s'han tingut en compte les següents modificacions i opcions:

- L'avaluació dels individus de la població s'executa en el mateix moment de la seva inserció a la nova població
- A l'hora de crear la nova població, es contempla la mutació com una possibilitat després del seu encreuament
- Els encreuaments s'executen entre el candidat i el millor cromosoma de la generació
- S'habilita l'opció de l'elitisme: el millor cromosoma de la generació sempre arriba a la nova població sense cap mena de modificació
- S'habilita l'opció de l'endogàmia: el millor cromosoma de la generació sempre és un dels pares per generar els cromosomes de la següent generació

4.1.1 Codificació i generació dels cromosomes

Els diferents cromosomes d'aquest algoritme codifiquen assignacions de llits per als diferents professionals on cada gen amb valor positiu representa un dels llits de la planta. A més a més, existeixen gens especials amb valor negatiu per separar les assignacions. Així, un exemple de cromosoma podria ser

[0, 1, 2, 3, 4, 5, 6, 7, 8, -1, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, -2, 20, 21, 22, 23, 24, 25, 26, 27, -3, 28, 29, 30, 31, 32, 33, 34, 35, 36]

on una de les infermeres té els pacients del llit 0 fins al 8, altra del 9 al 19, altra del 20 al 27 i la darrera, del 28 al 36.

A l'hora de generar els cromosomes inicials, s'agafa el conjunt de llits i s'afegeixen com a gens inicials, on el primer llit val 0 i es va augmentant el valor com més llits existeixin. Després s'afegeixen els gens de separació començant per -1 i augmentant segons el nombre d'infermeres. És a dir, el nombre total de gens és

$$G=L+(P-1)$$

on

G = nombre de gens

L = nombre de llits

P = nombre de professionals

Finalment, es duu a terme una permutació dels gens, atès que dues infermeres no poden atendre al mateix pacient. En funció del nombre de llits i professionals involucrats, es pot parlar d'un nombre de possibles permutacions que pot arribar a valors força elevats.

Combinacions		Professionals		
		2	3	4
Pacients	35	3,72e+41	1,38e+43	5,23e+44
	36	1,38e+43	5,23e+44	2,04e+46
	37	5,23e+44	2,04e+46	8,16e+47
	38	2,04e+46	8,16e+47	3,34e+49

4.1.2. Avaluació dels cromosomes

La funció de fitting de l'algoritme depèn de dues variables: per una banda, de la suma dels costos de la càrrega de cures; per l'altra, de les distàncies que necessita cobrir un professional per atendre a tots els pacients assignats.

Per dur a terme la seva avaluació, cada cromosoma se separa segons les seves assignacions i s'executen els següents càlculs:

- a) Suma de les càrregues i càlcul de la desviació estàndard
Atès que cada assignació és un subconjunt dels llits de la unitat, només és necessari trobar a la codificació de la càrrega de cures quin és el valor del llit assignat. Després es duu a terme la suma de les càrregues individuals per establir el total de la càrrega assignada al professional.

A tall d'exemple, si se suposa una codificació de càrrega de cures [1,2,3,4]

	Assignació A	Càrrega total
Infermera A	(0,1)	3
Infermera B	(2,3)	7

No obstant això, és necessari poder determinar si una assignació de pacients és millor que una altra des del punt de vista de la càrrega de

cures. És a dir, és necessari poder calcular fins a quin punt les assignacions són equilibrades. Per fer-ho, es defineix que una assignació és més equilibrada com més petita sigui la diferència entre els costos totals.

Com a exemple, seguint amb la codificació de càrrega anterior, es considera una distribució diferent

	Assignació B	Càrrega total
Infermera A	(0,3)	5
Infermera B	(1,2)	5

Es pot observar que, per a l'assignació A, hi ha una diferència de 4 i, per a l'assignació B, hi ha una diferència de 0 i, així, es considera aquesta com una assignació més equilibrada.

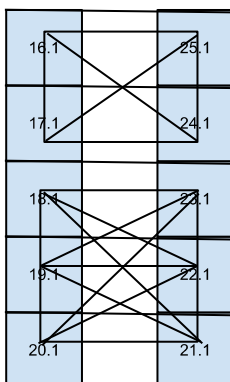
Encara que aquest càlcul sembla senzill, a mesura que augmenta el nombre de professionals implicats, es complica el seu càlcul i la seva interpretació. Afortunadament, existeix un estadístic que, precisament, calcula la dispersió dels diferents elements sobre la mitja aritmètica: la desviació estàndard. Així, es calcula aquest sobre el conjunt de valors de càrregues totals de les assignacions i com més proper a 0 sigui aquest, més equilibrada és l'assignació.

Així, per a l'assignació A, la desviació estàndard (σ) = 2 i per a l'assignació B, σ = 0.

b) Càlcul dels arbres generadors minimal i suma dels costos

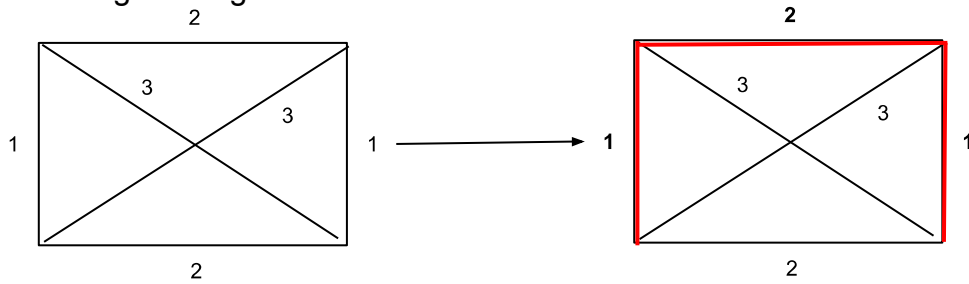
Tal com ja s'ha establert abans, cada un dels llits d'una assignació pertany a un punt de la quadrícula. Si es considera cada un dels punts com el vèrtex d'un graf, es pot crear un graf complet d' n vèrtex on el valor de cada aresta del graf ve determinada per la distància d'anar d'un llit a l'altra.

Per exemple, tenint les assignacions (16,17,24,25) i (18,19,20,21,22,23), es poden establir els grafes



Per poder determinar si les assignacions són equitatives en funció de les distàncies, és necessari establir algun tipus de mesura comparativa. Tot graf complet presenta el que s'anomena un arbre generador minimal que

és un arbre que connecta tots els vèrtexs d'un graf seguint el camí més curt seguint l'algoritme de Prim.



Així, per cada una de les assignacions, es pot calcular el cost d'aquest arbre i fer la suma. El valor d'aquesta suma representa el cost total de les distàncies que necessiten recórrer els professionals per atendre a tots els pacients de la unitat. Com més petit sigui aquest valor, més petits són els arbres generadors i, en conseqüència, més "compactes" són les assignacions dels pacients.

Aquests dos càlculs se sumen, generant un valor determinat de fitting. Encara així, per a cada una d'aquestes variables s'habilita la possibilitat d'utilitzar un coeficient d'ajust ("w_loads" i "w_dist") per donar més pes a alguna de les variables si es desitja.

Aquesta avaluació provoca que els millors individus de la generació, contràriament al que és habitual, siguin els que presenten els valors més petits. Això implica fer una transformació posterior a l'hora de determinar quins són els individus seleccionats a reproduir-se.

4.1.3. Selecció dels millors individus

El mètode de selecció dels millors individus que aportaran els seus gens a la nova generació és el de la ruleta, on cada un dels individus tenen una probabilitat determinada en funció del seu valor de fitting tal que, com més petit sigui aquest, més probabilitats tenen de ser escollits.

4.1.4. Mutacions i encreuaments

A l'hora de realitzar la mutació dels cromosomes és important tenir en compte que aquesta no es pot dur a terme de la manera habitual, atès que qualsevol modificació d'un gen, pot implicar la repetició d'aquest dins del mateix cromosoma. Així doncs, es considera la mutació com l'intercanvi de posició de dos gens del mateix cromosoma amb l'objectiu de:

- Assignar un pacient a una altra infermera
- Assignar diferents pacients d'una infermera a una altra

Com a exemple, la mutació del cromosoma

[0,1,2,-1,3,4,-2,5]

pot generar el cromosoma

[0,1,-1,2,3,4,-2,5]

De la mateixa manera, els encreuaments també poden provocar la repetició de gens i, per tant, és necessari dur a terme mecanismes que permetin dur un control d'aquesta situació. Per evitar-ho, s'executa el següent algoritme:

- Determinar una part fixa del cromosoma de partida: [0,1,2,-1,3,4,-2,5]
- Determinar la part que creuarem del segon progenitor: [1,3,5,-1,0,-2,2,4]
- Determinar els gens conflictius del segon progenitor: [1,3,5,-1,0,-2,2,4]
- Determinar els gens que mancaran: [0,1,2,-1,3,4,-2,5]
- Substituir els gens conflictius pels quals mancaran: [1,3,5,-1,3,-2,5,4]
- Dur a terme l'encreuament: [0,1,2,-1,3,-2,5,4]

Cal destacar que aquest encreuament es planteja de forma simple o doble, és a dir, es determinen una o dues part fixes dintre del cromosoma de partida. A més a més, es contempla l'opció de dur a terme encreuaments on es manté una assignació per a un professional de forma fixa, mantenint-la tal qual al nou cromosoma.

A tall d'exemple, partint dels cromosomes [0,1,2,-1,3,4,-2,5] i [1,3,5,-1,0,-2,2,4]:

- Es determina l'assignació al professional que quedarà fixa del primer progenitor: [0,1,2,-1,3,4,-2,5]
- Determinar la part que creuarem del segon progenitor: [1,3,5,-1,0,-2,2,4]
- Determinar els gens conflictius del segon progenitor: [1,3,5,-1,0,-2,2,4]
- Determinar els gens que mancaran: [0,1,2,-1,3,4,-2,5]
- Substituir els gens conflictius pels quals mancaran: [1,3,5,-1,3,-2,5,4]
- Dur a terme l'encreuament: [0,1,2,-1,3,-2,5,4]

4.2. Algoritme d'ubicació

La segona part de l'algoritme presenta el mateix disseny general que l'algoritme d'assignació amb una sèrie de particularitats que s'expliquen als diferents apartats.

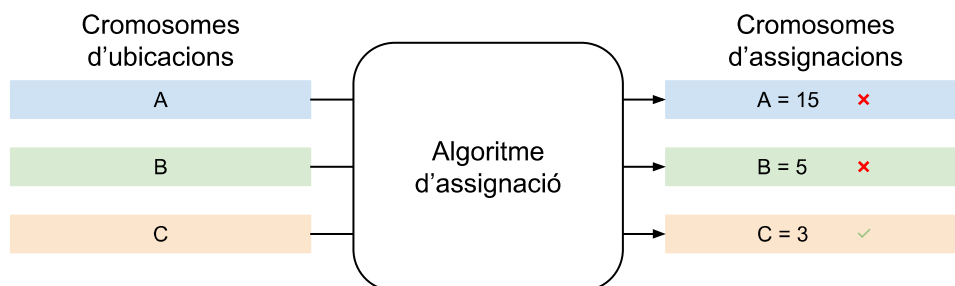
4.2.1. Codificació i generació dels cromosomes

Els cromosomes de l'algoritme d'ubicació codifiquen una nova distribució de les càrregues de cures dels pacients, presentant tants gens com llits hi hagi disponibles. A tall d'exemple, un cromosoma d'aquest algoritme pot presentar la forma [0,2,4,2,6,0] on la primera i darrera habitació de la unitat estan lliures.

Habitualment, la generació dels cromosomes inicials es duu a terme de forma aleatòria, generant permutacions dels seus elements com en el cas de l'algoritme d'assignació o bé amb altres mètodes aleatoris. Ara bé, cal destacar el fet que qualsevol modificació dels gens d'aquests cromosomes implica un canvi d'habitació dels pacients, la qual cosa té el risc potencial d'empitjorar l'estat del pacient. Així doncs, sense entrar en altre tipus de restriccions, es decideix una limitació pel que fa a les modificacions dels cromosomes, només permetent les mutacions.

4.2.2. Avaluació dels cromosomes

Els cromosomes d'aquest algoritme s'avaluen fent servir l'algoritme d'assignació. És a dir, cada cromosoma serveix d'entrada a una execució de l'algoritme d'assignació, el qual retornarà una assignació de pacients que sigui prou factible i, per tant, es considera aquest resultat com el valor de fitting de l'algoritme d'ubicació. Així, es considerarà que un cromosoma és millor que altra si el valor de la seva assignació de pacients és inferior a un altre.



4.2.3. Selecció dels millors individus

Igual que en el cas de l'algoritme d'assignació, el mètode de selecció per aquest algoritme és el de la ruleta, seguint el mateix criteri.

4.2.4. Mutacions i encreuaments

Les mutacions dels cromosomes d'aquest algoritme han de seguir el mateix principi d'evitar possibles repeticions d'un gen dintre del cromosoma, atès que implicaria que un mateix pacient s'ubicaria a dos llits de la unitat. Així, es torna a considerar una mutació com l'intercanvi de posició de dos gens del mateix cromosoma amb l'objectiu de:

- Intercanviar dos pacients de llit
- Assignar un pacient a un llit disponible

Així doncs, la mutació del cromosoma

[0,2,4,2,6,0]

pot generar el nou cromosoma

[0,**0**,4,2,6,**2**]

Això no obstant, tal com ja s'ha comentat, aquest algoritme no admet encreuaments dels cromosomes, atès que implicaria multitud de canvis de pacients dintre de la unitat, generant un alt risc per aquests. És cert que aquest fet limita molt l'espai de cerca de l'algoritme, però s'accepta aquest límit per tal d'aconseguir resultats que puguin ser aplicats en el món real.

5. Resultats

5.1. Algoritme d'assignació

Per avaluar si l'algoritme funciona correctament, es plantegen diferents situacions per veure si el resultat final és factible.

Els paràmetres inicials que es faran servir són:

- Mida de la població: 100 individus
- Probabilitat d'encreuament: 80%
- Probabilitat de les mutacions: 5%
- Màxim de generacions: 5000
- Màxim de generacions sense canvis: 1500
- Amb elitisme i endogàmia

5.1.1. Validació dels paràmetres de la funció de fitting

Atesa la complexitat d'aquesta funció, es vol comprovar que els valors independents d'aquesta es calculen correctament. Així, es duu a terme una avaluació bàsica amb una mostra de 10 habitacions amb 2 infermeres.

En primer lloc s'avalua si l'algoritme pot trobar una assignació amb una desviació estàndard mínima sense tenir en compte les distàncies entre els llits. Per dur-ho a terme, es considera que la càrrega de pacients és la següent:

9,1,1,1,1,1,1,1,1,1

Si l'algoritme és correcte, assignarà el primer pacient a una infermera i la resta a l'altra per una desviació estàndard de 0. L'algoritme d'assignació s'executa assignant un valor de 0 al paràmetre "w_dist" amb l'objectiu d'anul·lar el valor del cost dels arbres mínims. Així, l'algoritme troba el següent cromosoma

[6, 4, 2, 3, 8, 9, 5, 1, 7, -1, 0]

que produeix una assignació tal que

	Assignació	Càrrega total
Infermera A	(6,4,2,3,8,9,5,1,7)	9
Infermera B	(0)	9

la qual té un valor $\sigma = 0$, tal com s'esperava.

Per tal d'observar els tipus de distàncies que l'algoritme pot establir, es decideix executar l'algoritme assignant un valor de 0 al paràmetre "w_loads", per tal d'anul·lar el valor del cost de les càrregues de cures. A més a més, es presentaran tres tipus de distribucions per comprovar que el comportament és coherent. Com a primer cas es considera una distribució de la planta en forma de passadís.

0 (0,0)	1 (0,1)	2 (0,2)	3 (0,3)	4 (0,4)
------------	------------	------------	------------	------------

5 (2,0)	6 (2,1)	7 (2,2)	8 (2,3)	9 (2,4)
------------	------------	------------	------------	------------

L'algorithm troba el següent cromosoma que fa una assignació equitativa dels pacients

[4, 1, 3, 2, 0, -1, 8, 5, 9, 7, 6]

0 (0,0)	1 (0,1)	2 (0,2)	3 (0,3)	4 (0,4)
------------	------------	------------	------------	------------

5 (2,0)	6 (2,1)	7 (2,2)	8 (2,3)	9 (2,4)
------------	------------	------------	------------	------------

	Assignació	Cost de l'arbre mínim
Infermera A	(4,1,3,2,0)	4
Infermera B	(8,5,9,7,6)	4

Ara es prova una distribució de pacients amb l'existència d'un control central d'infermeria.

	0 (0,1)	1 (0,2)	2 (0,3)	
9 (1,0)				3 (1,4)
8 (3,0)				4 (3,4)
	7 (4,1)	6 (4,2)	5 (4,3)	

S'executa l'algorithm i es troba un cromosoma que assigna als pacients de la següent forma

[0, 1, 2, 7, 6, 4, 3, 5, -1, 8, 9]

	0 (0,1)	1 (0,2)	2 (0,3)	
9 (1,0)				3 (1,4)
8 (3,0)				4 (3,4)
	7 (4,1)	6 (4,2)	5 (4,3)	

	Assignació	Cost de l'arbre mínim
Infermera A	(0,1,2,7,6,4,3,5)	10
Infermera B	(8,9)	2

Encara que l'assignació sembla no ser correcta, fent el càlcul dels arbres es comprova que no hi ha cap altra assignació que millori la suma dels costos dels arbres mínims. Així, aquesta assignació compleix amb els requisits, atès que en cap moment es demana que l'assignació sigui equitativa.

Finalment, es planteja una distribució en forma d'L de la planta

0 (0,0)	1 (0,1)	2 (0,2)	3 (0,3)	
9 (2,0)	8 (2,1)	7 (2,2)		4 (2,4)
		6 (3,2)		5 (3,4)

L'algoritme troba una configuració que fa l'assignació

[4, 5, -1, 6, 7, 3, 2, 0, 9, 1, 8]

0 (0,0)	1 (0,1)	2 (0,2)	3 (0,3)	
9 (2,0)	8 (2,1)	7 (2,2)		4 (2,4)
		6 (3,2)		5 (3,4)

	Assignació	Cost de l'arbre mínim
Infermera A	(6,7,3,2,0,9,1,8)	8
Infermera B	(4,5)	1

Igual que el cas anterior, sembla que les assignacions no compleixen amb els requisits, però no existeixen altres arbres mínims que tinguin una millor suma.

Així, es pot afirmar que, en un principi, els paràmetres de la funció de fitting es calculen de forma vàlida.

5.1.2. Comportament de l'algoritme

Per comprovar com treballa l'algoritme, es plantegen casos bàsics per calcular l'assignació a dues infermeres amb diferents distribucions de les unitats i dels pacients.

Concretament, es plantegen els casos següents:

- Cas A: distribució de la planta en forma de passadís:
 - Cas A.1: càrrega de cures equitativa
 - Cas A.2: 2 pacients amb elevada càrrega de cures als extrems del passadís
 - Cas A.3: 2 pacients amb elevada càrrega situats en habitacions conjuntes
 - Cas A.4: pacients amb diferents càrregues de treball
- Cas B: distribució de la planta amb control central:
 - Cas B.1: càrrega de cures equitativa
 - Cas B.2: 2 pacients amb càrrega elevada en els extrems
 - Cas B.3: 2 pacients amb càrrega elevada en habitacions conjuntes
 - Cas B.4: pacients amb diferents càrregues de treball
- Cas C: distribució de la planta amb forma d'L:
 - Cas C.1: càrrega de cures equitativa
 - Cas C.2: 2 pacients amb càrrega elevada als extrems
 - Cas C.3: 2 pacients amb càrrega elevada en habitacions conjuntes
 - Cas C.4: pacients amb diferents càrregues de treball

Cas A.1

Amb aquest exemple, s'espera que l'algoritme dugui a terme una separació de la unitat en dues parts que siguin completament equitatives, atès que les càrregues de cures són totalment homogènies i, per tant, l'únic determinant seran els abres mínims.

0 (0,0) 1	1 (0,1) 1	2 (0,2) 1	3 (0,3) 1	4 (0,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 1	6 (2,1) 1	7 (2,2) 1	8 (2,3) 1	9 (2,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

L'algoritme s'executa i troba l'assignació

Chromosome with score 8.0

Assignment: **5, 6, 7, 8, 9**. Load score: 5. Distance score: 4

Assignment: **0, 1, 2, 3, 4**. Load score: 5. Distance score: 4

0 (0,0) 1	1 (0,1) 1	2 (0,2) 1	3 (0,3) 1	4 (0,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 1	6 (2,1) 1	7 (2,2) 1	8 (2,3) 1	9 (2,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

Amb aquesta assignació es pot observar que es duu a terme una distribució equitativa de pacients, on les càrregues de cures es troben perfectament equilibrades, amb les millors distàncies possibles.

Cas A.2

Aquest exemple introdueix dos pacients que, en comparació tenen una càrrega de cures elevada. En qualsevol cas, considerant que els dos es troben separats, s'espera que l'algoritme respongui com en el cas anterior, assignant la mateixa càrrega de cures als professionals.

0 (0,0) 4	1 (0,1) 1	2 (0,2) 1	3 (0,3) 1	4 (0,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 1	6 (2,1) 1	7 (2,2) 1	8 (2,3) 1	9 (2,4) 4
-----------------	-----------------	-----------------	-----------------	-----------------

L'execució de l'algoritme genera el cromosoma

Chromosome with score 8.0

Assignment: **0, 1, 2, 3, 4**. Load score: 8. Distance score: 4

Assignment: **5, 6, 7, 8, 9**. Load score: 8. Distance score: 4

0 (0,0) 4	1 (0,1) 1	2 (0,2) 1	3 (0,3) 1	4 (0,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 1	6 (2,1) 1	7 (2,2) 1	8 (2,3) 1	9 (2,4) 4
-----------------	-----------------	-----------------	-----------------	-----------------

Igual que en el cas anterior, aquesta assignació de pacients distribueix als pacients amb una càrrega perfectament equilibrada ($\sigma = 0$) i els millors camins mínims possibles.

Cas A.3

Aquest exemple intenta forçar un desequilibri a l'hora d'assignar els pacients. El fet de tenir dos pacients amb una càrrega elevada de cures situats molt propers implica que l'algoritme ha de decidir quina mena d'assignació és la més adient.

0 (0,0) 1	1 (0,1) 4	2 (0,2) 4	3 (0,3) 1	4 (0,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 1	6 (2,1) 1	7 (2,2) 1	8 (2,3) 1	9 (2,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

S'executa l'algoritma i es genera el cromosoma

Chromosome with score 10.0

Assignment: **0, 1, 2**. Load score: 9. Distance score: 2

Assignment: **3, 4, 5, 6, 7, 8, 9**. Load score: 7. Distance score: 7

0 (0,0) 1	1 (0,1) 4	2 (0,2) 4	3 (0,3) 1	4 (0,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 1	6 (2,1) 1	7 (2,2) 1	8 (2,3) 1	9 (2,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

S'observa que aquesta assignació genera un valor $\sigma = 1$ que no es pot millorar i dos arbres mínims que proporcionen la suma més petita possible.

Cas A.4:

Finalment, aquest exemple pretén observar quina serà l'execució de l'algoritme en un entorn heterogeni.

0 (0,0) 6	1 (0,1) 3	2 (0,2) 4	3 (0,3) 4	4 (0,4) 4
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 3	6 (2,1) 5	7 (2,2) 2	8 (2,3) 5	9 (2,4) 3
-----------------	-----------------	-----------------	-----------------	-----------------

La seva execució genera un cromosoma tal que

Chromosome with score 9.5

Assignment: **0, 1, 2, 3, 4**. Load score: 21. Distance score: 4

Assignment: **5, 6, 7, 8, 9**. Load score: 18. Distance score: 4

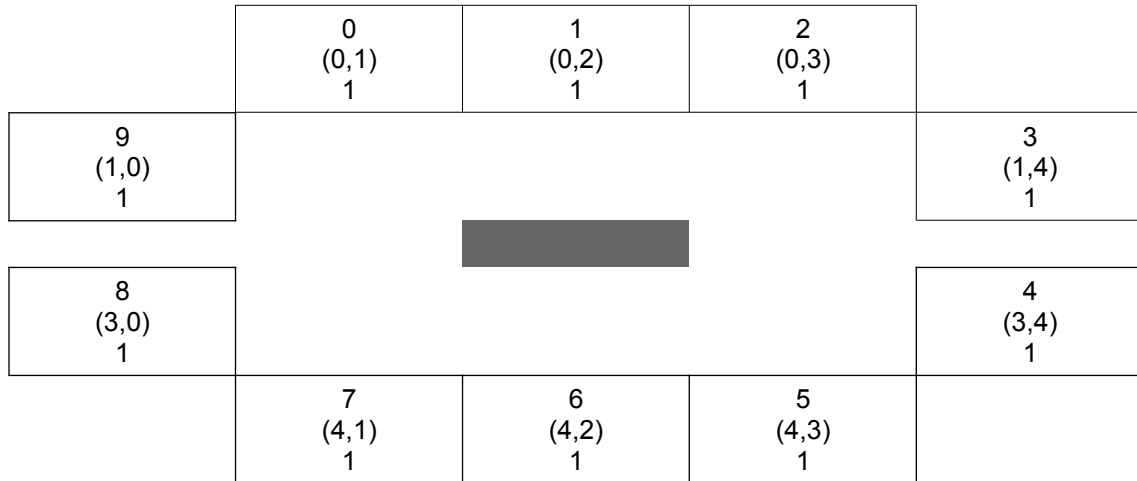
0 (0,0) 6	1 (0,1) 3	2 (0,2) 4	3 (0,3) 4	4 (0,4) 4
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 3	6 (2,1) 5	7 (2,2) 2	8 (2,3) 5	9 (2,4) 3
-----------------	-----------------	-----------------	-----------------	-----------------

Aquesta assignació genera dues assignacions amb $\sigma = 1.5$ i uns arbres mínims que no es poden millorar.

Cas B.1:

A continuació es defineix una unitat amb control central que presenta una càrrega homogènia de cures. Igual que en el cas A.1, s'espera que l'algoritme divideixi la unitat en dues parts iguals, definida pel cost dels arbres mínims.

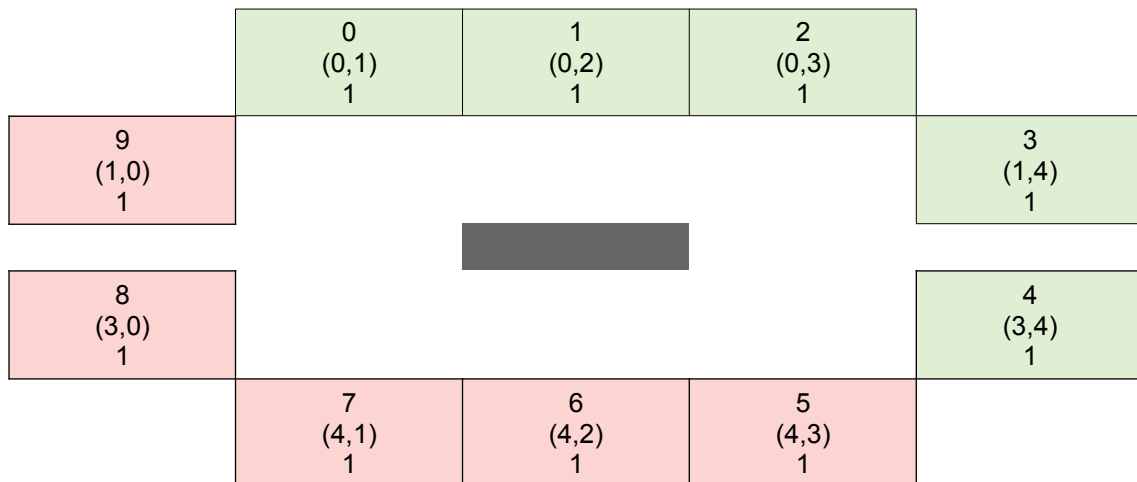


L'algoritme retorna la següent assignació

Chromosome with score 12.0

Assignment: **5, 6, 7, 8, 9**. Load score: 5. Distance score: 6

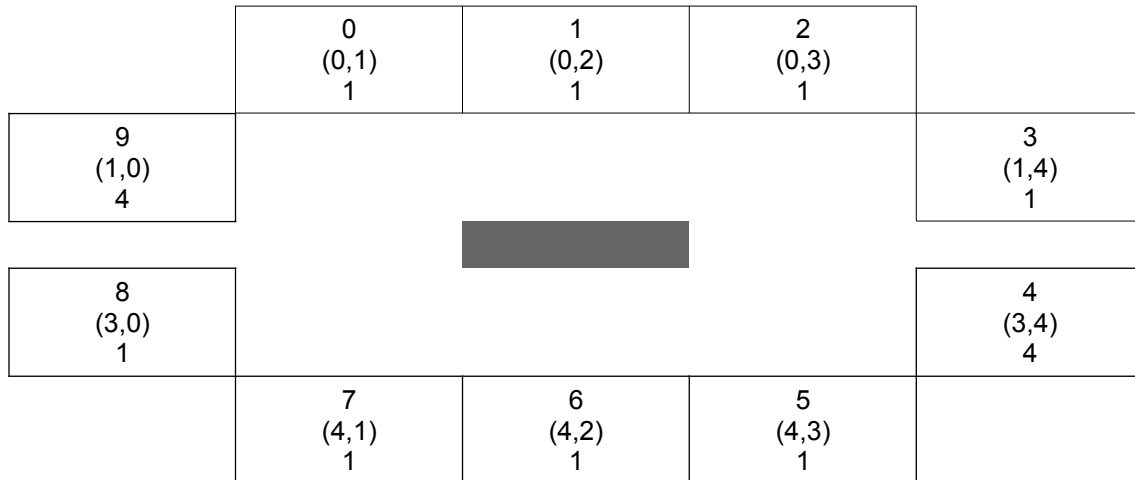
Assignment: **0, 1, 2, 3, 4**. Load score: 5. Distance score: 6



On s'observa una assignació de pacients totalment equitativa, així com uns camins mínims òptims que, en efecte, divideixen la unitat de forma totalment regular.

Cas B.2

Igual que en el cas A.2, la ubicació de pacients amb elevada càrrega de cures als extrems vol assegurar que l'algoritme és capaç de dividir la unitat en dues parts iguals, assignant els pacients de forma equitativa, on els arbres mínims decidiran la divisió òptima.

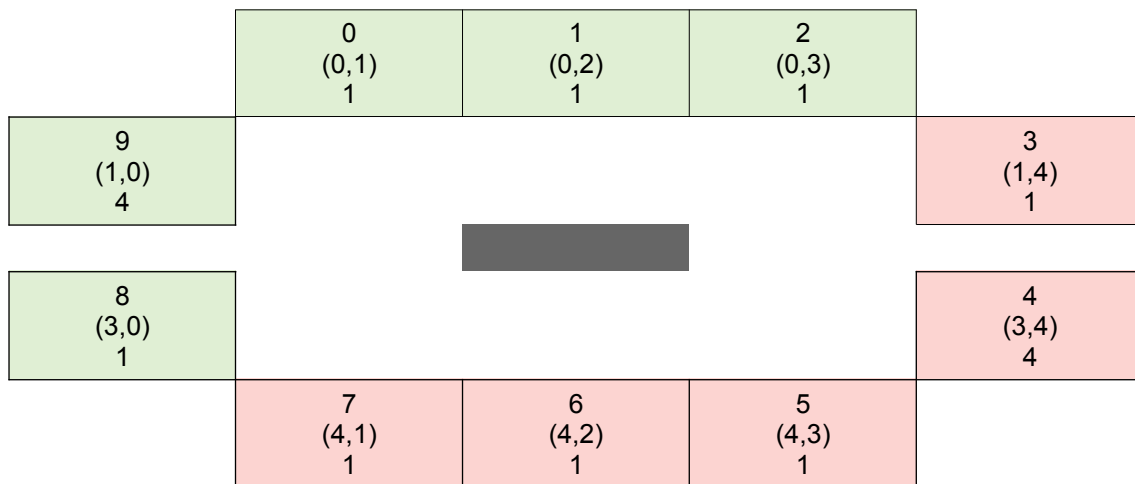


La seva execució genera un cromosoma d'assignació tal que

Chromosome with score 12.0

Assignment: **3, 4, 5, 6, 7**. Load score: 8. Distance score: 6

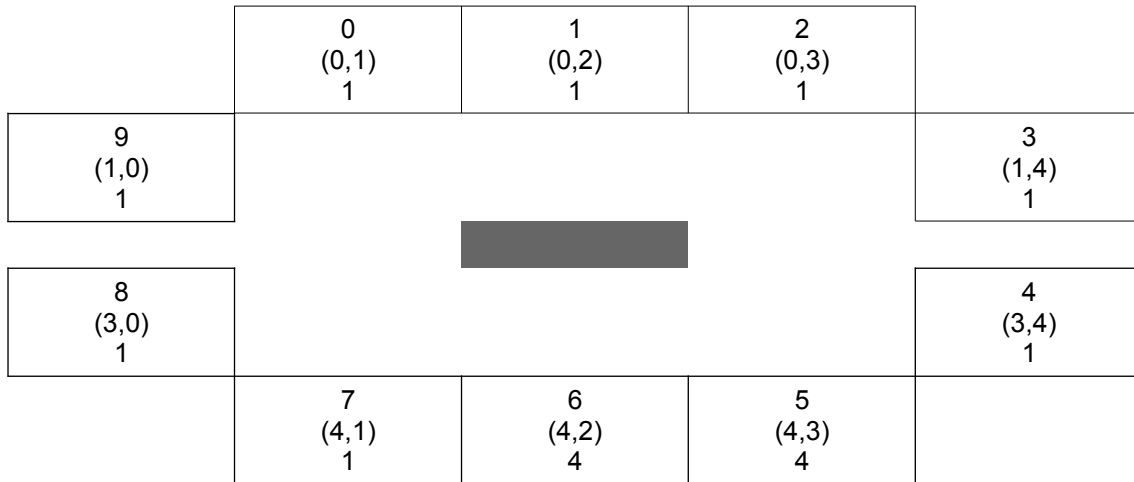
Assignment: **0, 1, 2, 8, 9**. Load score: 8. Distance score: 6



Igual que en el cas anterior, aquesta assignació proporciona una assignació totalment equilibrada i camins mínims òptims, dividint la unitat en dues parts.

Cas B.3

Així com el cas A.3 volia comprova com assignava l'algoritme tenint en compte la ubicació propera dels pacients amb càrrega elevada de cures, aquest cas també vol dur a terme la mateixa comprovació, però tenint en compte la nova arquitectura.

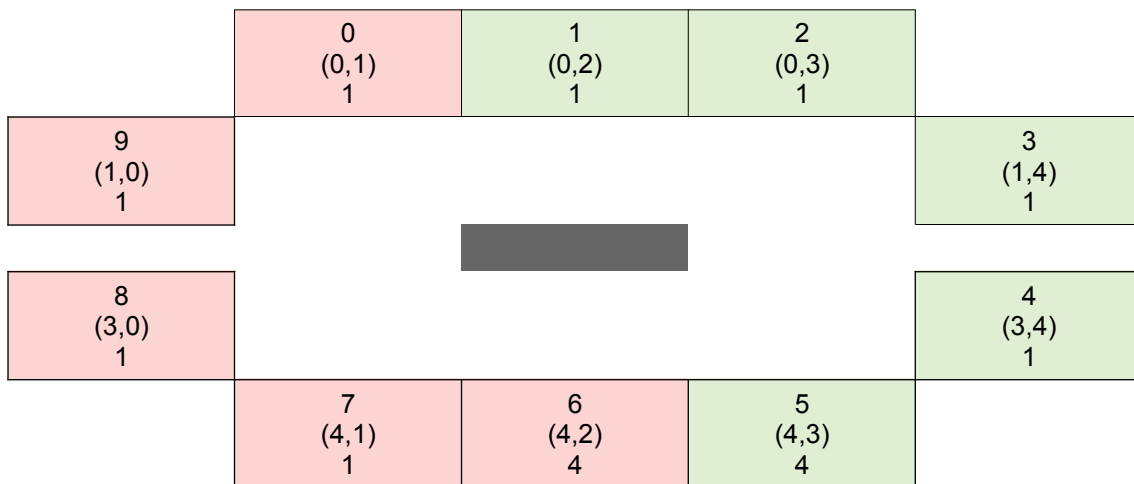


L'execució genera el cromosoma

Chromosome with score 14.0

Assignment: **0, 6, 7, 8, 9**. Load score: 8. Distance score: 7

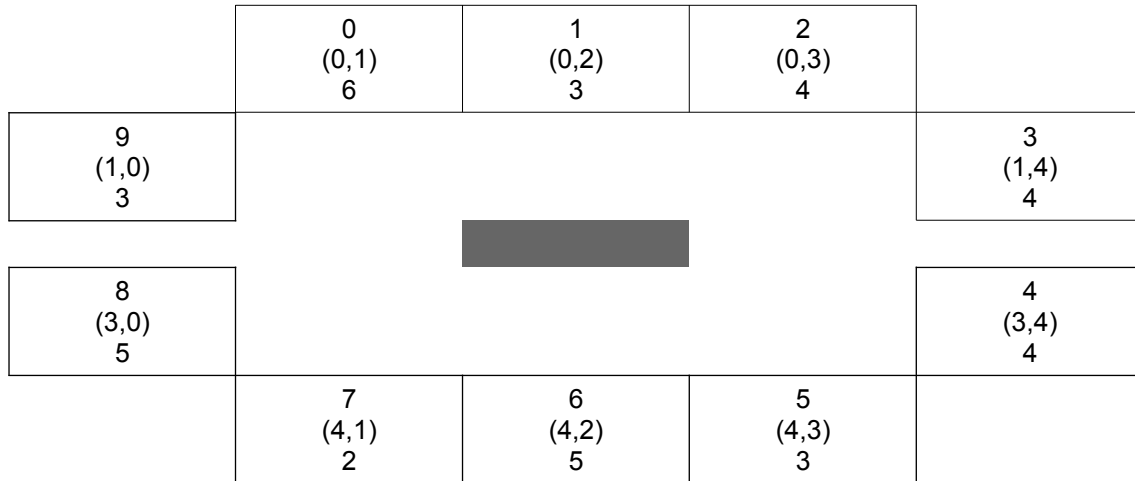
Assignment: **1, 2, 3, 4, 5**. Load score: 8. Distance score: 7



S'observa que aquesta assignació proporciona unes càrregues totalment equilibrades amb arbres mínims adequats amb la particularitat de que fa una divisió exacta de la unitat.

Cas B.4

El darrer exemple assignat a la unitat amb control central presenta una distribució heterogènia de pacients, per tal d'observar el comportament de l'algoritme.

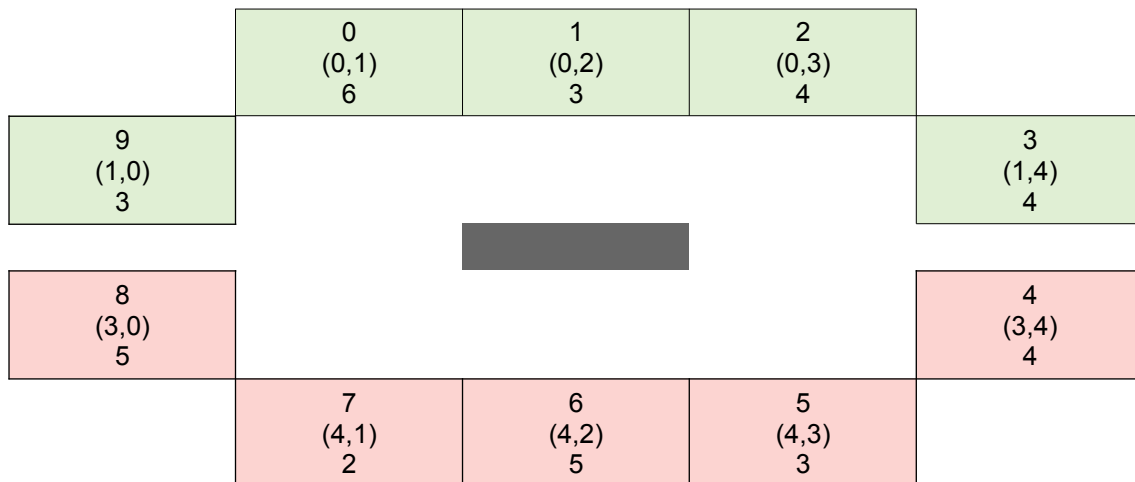


La seva execució genera el cromosoma

Chromosome with score 12.5

Assignment: **0, 1, 2, 3, 9**. Load score: 20. Distance score: 6

Assignment: **4, 5, 6, 7, 8**. Load score: 19. Distance score: 6



Aquesta assignació genera assignacions de càrregues de cures molt equilibrades ($\sigma = 0.5$) i uns camins mínims òptims.

Cas C.1

Finalment, es consideren les unitats amb forma d'L. Habitualment, l'assignació de pacients en aquest tipus d'unitats es duen a terme per "zones" o conjunts d'habitacions. En el cas que ens ocupa, l'assignació habitual podria considerar assignar les habitacions 0, 1, 2 i 3 a una infermera i la resta a l'altra, però es vol comprovar si l'algoritme pot trobar una distribució més equitativa.

0 (0,0) 1	1 (0,1) 1	2 (0,2) 1	3 (0,3) 1	
9 (2,0) 1	8 (2,1) 1	7 (2,2) 1		4 (2,4) 1
		6 (3,2) 1		5 (3,4) 1

L'algoritme ens retorna l'assignació

Chromosome with score 10.0

Assignment: **4, 5, 6, 7, 8**. Load score: 5. Distance score: 5

Assignment: **0, 1, 2, 3, 9**. Load score: 5. Distance score: 5

0 (0,0) 1	1 (0,1) 1	2 (0,2) 1	3 (0,3) 1	
9 (2,0) 1	8 (2,1) 1	7 (2,2) 1		4 (2,4) 1
		6 (3,2) 1		5 (3,4) 1

Aquesta assignació genera una càrrega equilibrada, així com camins mínims òptims. És interessant destacar que fa una assignació de pacients que no es consideraria en el món real.

Cas C.2

En aquest exemple, es vol observar si l'algoritme és capaç d'assignar els pacients seguint els mateixos principis d'equilibri en la càrrega de cures i assignacions compactes, atès que l'arquitectura i ubicació dels pacients pot fer-ho força complicat.

0 (0,0) 4	1 (0,1) 1	2 (0,2) 1	3 (0,3) 1	
9 (2,0) 1	8 (2,1) 1	7 (2,2) 1		4 (2,4) 1
		6 (3,2) 1		5 (3,4) 4

L'execució de l'algoritme genera el cromosoma

Chromosome with score 10.0

Assignment: **0, 1, 2, 3, 9**. Load score: 8. Distance score: 5

Assignment: **4, 5, 6, 7, 8**. Load score: 8. Distance score: 5

0 (0,0) 4	1 (0,1) 1	2 (0,2) 1	3 (0,3) 1	
9 (2,0) 1	8 (2,1) 1	7 (2,2) 1		4 (2,4) 1
		6 (3,2) 1		5 (3,4) 4

Com en el cas anterior, l'assignació proporciona una distribució equilibrada de la càrrega de cures i camins òptims.

Cas C.3

Aquest cas vol comprovar si, de la mateixa manera que en els casos A.3 i B.3 ha trobat assignacions equitatives, també ho pot dur a terme en aquest tipus d'arquitectura.

0 (0,0) 4	1 (0,1) 4	2 (0,2) 1	3 (0,3) 1	
9 (2,0) 1	8 (2,1) 1	7 (2,2) 1		4 (2,4) 1
		6 (3,2) 1		5 (3,4) 1

S'executa l'algoritme i es genera el cromosoma

Chromosome with score 11.0

Assignment: **1, 2, 3, 4, 5**. Load score: 8. Distance score: 6

Assignment: **0, 6, 7, 8, 9**. Load score: 8. Distance score: 5

0 (0,0) 4	1 (0,1) 4	2 (0,2) 1	3 (0,3) 1	
9 (2,0) 1	8 (2,1) 1	7 (2,2) 1		4 (2,4) 1
		6 (3,2) 1		5 (3,4) 1

Amb aquesta distribució s'observa que la càrrega de cures queda perfectament equilibrada amb $\sigma = 0$ i uns arbres força bons que no es poden millorar. A més a més, igual que en els casos esmentats, es fa una divisió de la unitat força regular.

Cas C.4

El darrer cas del set de proves per l'algoritme d'assignació vol observar el comportament d'aquest en una unitat que presenta pacients amb una distribució heterogènia.

0 (0,0) 6	1 (0,1) 3	2 (0,2) 4	3 (0,3) 4	
9 (2,0) 3	8 (2,1) 5	7 (2,2) 2		4 (2,4) 4
		6 (3,2) 5		5 (3,4) 3

La seva execució retorna el cromosoma

Chromosome with score 10.5

Assignment: **0, 1, 2, 3, 9**. Load score: 20. Distance score: 5

Assignment: **4, 5, 6, 7, 8**. Load score: 19. Distance score: 5

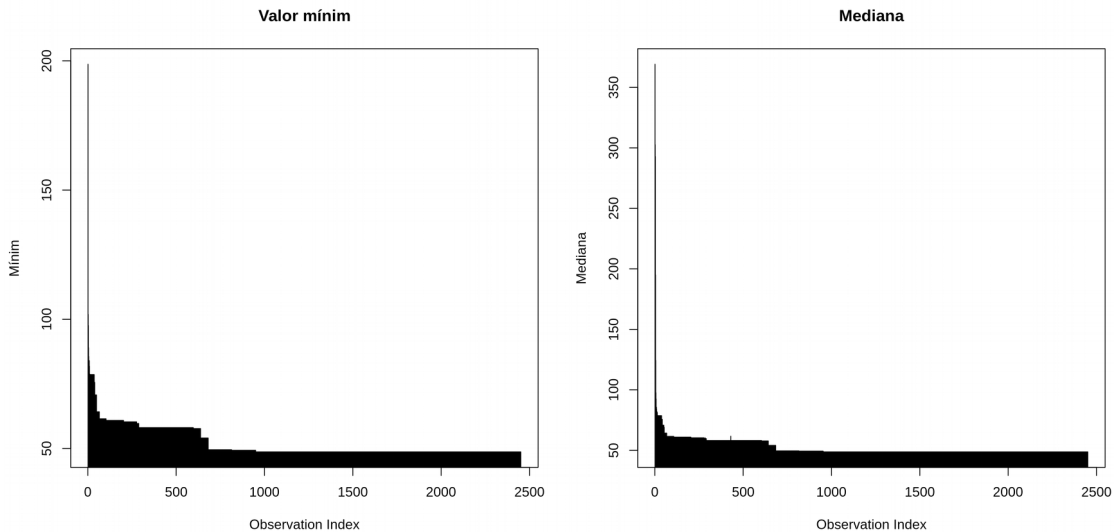
0 (0,0) 6	1 (0,1) 3	2 (0,2) 4	3 (0,3) 4	
9 (2,0) 3	8 (2,1) 5	7 (2,2) 2		4 (2,4) 4
		6 (3,2) 5		5 (3,4) 3

S'observa que aquesta darrera assignació proporciona una càrrega equilibrada ($\sigma = 0.5$) i uns camins mínims força bons.

En conclusió, s'observa que l'algoritme respon molt bé als diferents escenaris proposats, generant una assignació de pacients equilibrada i uns camins mínims correctes.

5.1.3. Aplicació de l'algoritme en un cas real

Amb els paràmetres inicials, es fa una execució estàndard de l'algoritme per obtenir una primera determinació. Després de 45 segons i 2450 generacions, troba una assignació amb valor 48.69 a la generació 949.



Il·lustració 11: evolució del valor mínim *Il·lustració 12: evolució de la mediana de l'algoritme d'assignació*

Es pot apreciar com el valor mínim convergeix de forma molt ràpida a les primeres generacions, apreciand-se una "desacceleració" gradual fins a arribar al valor final. De la mateixa manera, la mediana també presenta un comportament similar, mostrant com el millor cromosoma ocupa el 50% de la població ràpidament, eliminant la variabilitat genètica i bloquejant l'exploració d'altres solucions.

El cromosoma d'assignació resultat d'aquesta execució presenta la següent definició

Chromosome with score 48.69258240356725

Assignment: 16, 21, 22, 23, 24, 31, 32, 33.

Load score: 405. Distance score: 15

Assignment: 5, 6, 7, 8, 9, 10, 11, 13, 15, 17, 19, 36.

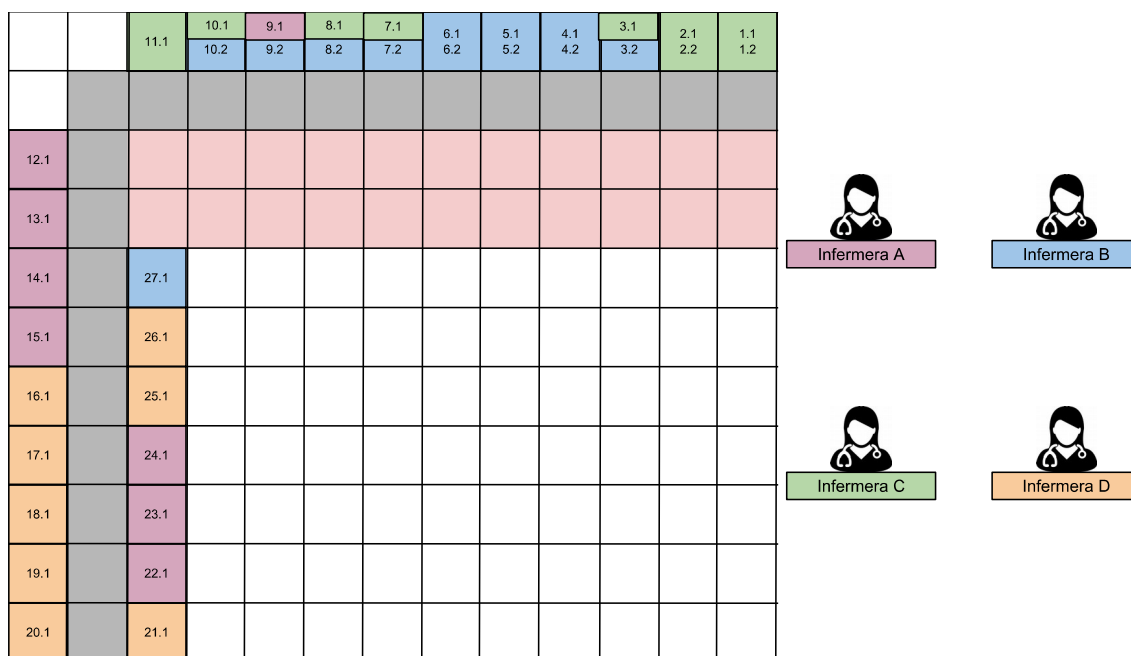
Load score: 404. Distance score: 12

Assignment: 0, 1, 2, 3, 4, 12, 14, 18, 20.

Load score: 403. Distance score: 10

Assignment: 25, 26, 27, 28, 29, 30, 34, 35.

Load score: 410. Distance score: 9



Il·lustració 13: representació gràfica de les associacions del cas real

Càrrega

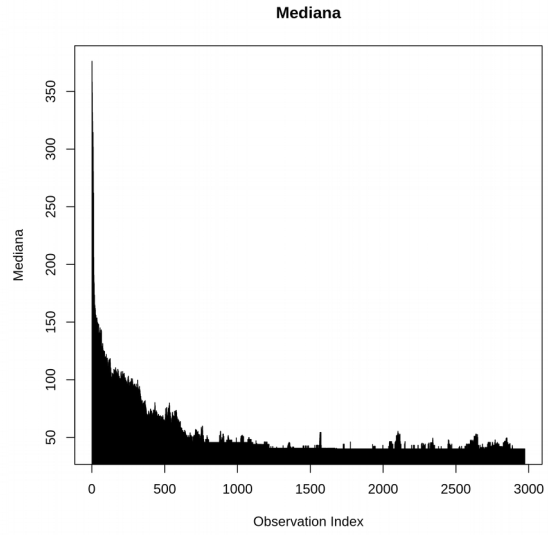
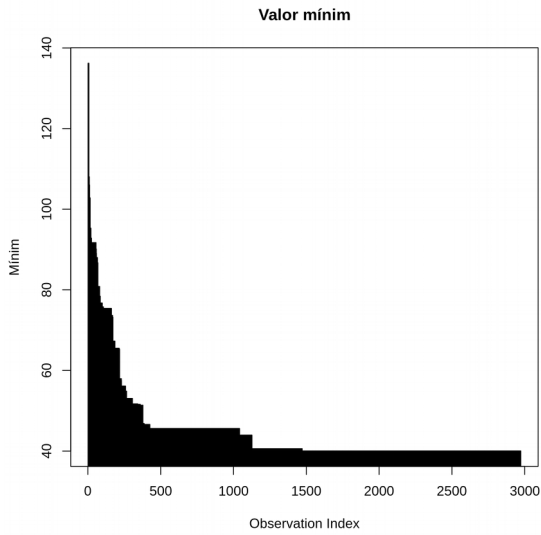
11.1	10.1	9.1	8.1	7.1	6.1	5.1	4.1	3.1	2.1	1.1
64	53	46	67	45	35	28	39	46	43	0
	10.2	9.2	8.2	7.2	6.2	5.2	4.2	3.2	2.2	1.2
	35	29	44	35	35	33	29	39	47	38

12.1	46
13.1	61
14.1	44
15.1	38
16.1	52
17.1	55
18.1	61
19.1	57
20.1	46

27.1	23
26.1	33
25.1	62
24.1	75
23.1	51
22.1	44
21.1	44

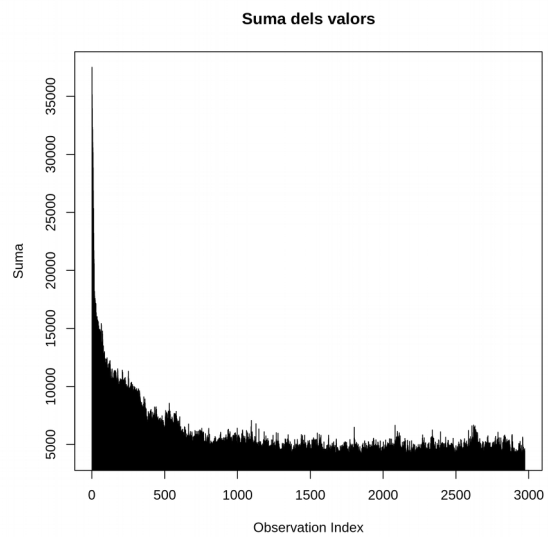
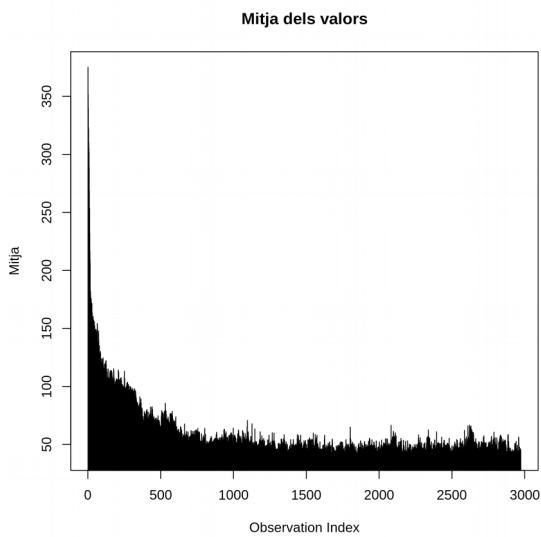
Il·lustració 14: hotmap de les càrregues

Encara que l'assignació sembla ser força correcta, atesa la ràpida convergència de la funció de fitting i la poca variabilitat de la cerca, es decideix implementar l'opció d'evitar l'endogàmia, és a dir, permetre a altres cromosomes de participar en les tasques de reproducció amb l'objectiu de mantenir "viu" l'espai de cerca. Després de 2972 generacions i 56 segons, s'obté un individu amb valor 40.04 a la generació 1471.



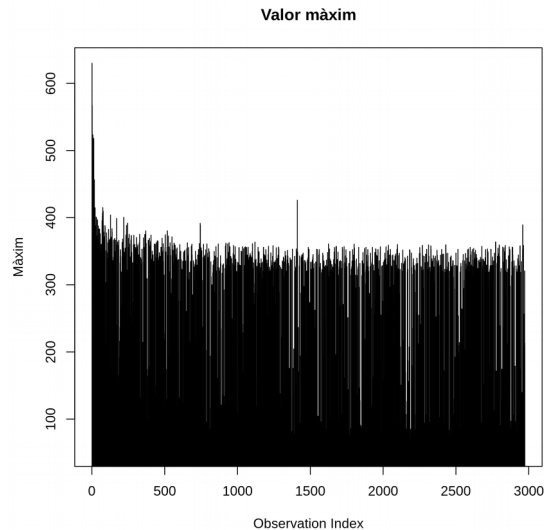
Il·lustració 15: evolució del valor mínim de l'algoritme d'assignació post-ajusts

Il·lustració 16: evolució de la mediana dels valors de l'algoritme d'assignació post-ajusts



Il·lustració 17: evolució de la mitja dels valors de l'algoritme d'assignació post-ajusts

Il·lustració 18: evolució de la suma dels valors de l'algoritme d'assignació post-ajusts



Il·lustració 19: evolució dels valors màxims de l'algoritme d'assignació post-ajusts

A les gràfiques es pot apreciar com ara es convergeix d'una manera més suau i com la mediana, encara que arriba a la convergència amb el valor mínim, presenta pics que mostren com s'ha intentat explorar noves opcions. També es pot observar com la mitja dels valors es manté estable, mostrant com, encara que hi hagi més exploració (com mostra la gràfica de màxims), la cerca de la solució es duu a terme al voltant dels millors cromosomes. Ara bé, encara que el temps que triga a arribar a una solució augmenta lleugerament, és important no ignorar-ho, atès que aquest augment en el temps d'execució derivarà en un pitjor comportament de la segona part de l'algoritme i, per tant, caldria explorar altres opcions com ara la computació en paral·lel per intentar pal·liar el seu efecte.

Finalment, es mostra l'assignació de pacients definida per aquest cromosoma:

Assignment: 25, 26, 27, 28, 29, 33, 34.

Load score: 408. Distance score: 7

Assignment: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 14.

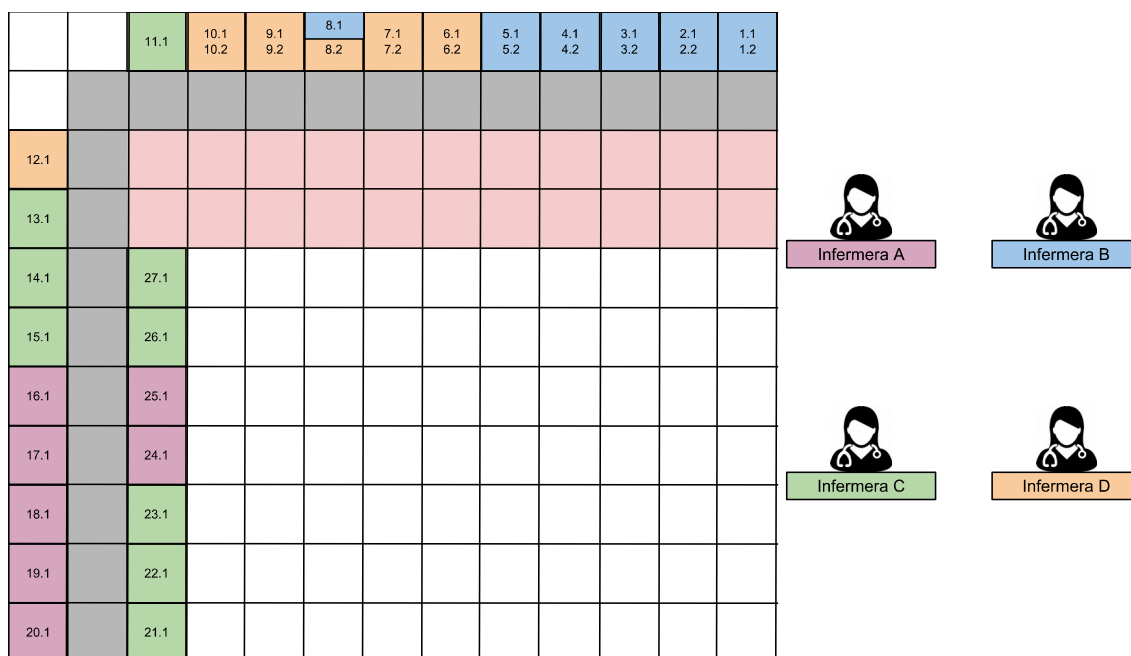
Load score: 409. Distance score: 7

Assignment: 20, 22, 23, 24, 30, 31, 32, 35, 36.

Load score: 402. Distance score: 14

Assignment: 10, 11, 12, 13, 15, 16, 17, 18, 19, 21.

Load score: 403. Distance score: 9



Il·lustració 20: representació gràfica de les associacions del cas real post-ajusts

A la il·lustració es pot observar com l'algoritme ha anat assignant pacients a les infermeres intentant agrupar els pacients assignats el màxim possible. L'assignació demostra un valor $\sigma = 3.04$, que és força ajustat si es té en compte els valors de càrregues dels pacients. Pel que fa als arbres, caldria valorar si aquests són òptims o no, però, visualment, es pot veure que, en general, les assignacions són bastant compactes. És interessant destacar el fet que, visualment, les habitacions 11 i 12 es podrien intercanviar entre sí per a les seves assignacions, la qual cosa obre la possibilitat de fer canvis d'ubicació de pacients per intentar equilibrar les assignacions.

5.2. Algoritme d'ubicació

Per avaluar el seu correcte funcionament, es plantegen diferents situacions per comprovar la seva viabilitat.

Els paràmetres que es faran servir per a l'avaluació de l'algoritme d'associació seran:

- Mida de la població: 100 individus
- Probabilitat d'encreuament: 80%
- Probabilitat de les mutacions: 5%
- Màxim de generacions: 5000
- Màxim de generacions sense canvis: 1500
- Amb elitisme i sense endogàmia

Per altra banda, ateses les necessitats especials de l'algoritme d'ubicació es defineixen els següents paràmetres:

- Mida de la població: 30 individus
- Sense encreuament
- Probabilitat de mutacions del 100%
- Màxim de generacions: 5
- Màxim de generacions sense canvis: 2
- Amb elitisme. L'endogàmia no té cap mena de paper

5.2.1. Comportament de l'algoritme

Per comprovar com treballa l'algoritme, es plantegen una sèrie de casos bàsics per observar si a una planta amb dues infermeres se suggereixen canvis d'ubicació dels pacients amb l'objectiu de millorar l'assignació.

Així, es plantegen els casos següents:

- Cas A: distribució de la planta en forma de passadís:
 - Cas A.1: dos pacients amb elevada càrrega de cures ubicats a un dels costats del passadís
 - Cas A.2: pacients amb diferents càrregues de treball
- Cas B: distribució de la planta amb control central:
 - Cas B.1: dos pacients amb elevada càrrega de cures ubicats a un dels costats de la unitat
 - Cas B.2: pacients amb diferents càrregues de treball
- Cas C: distribució de la planta en forma d'L:
 - Cas C.1: dos pacientss amb elevada càrrega de cures ubicats a un dels costats de la unitat
 - Cas C.2: pacients amb diferents càrregues de treball

Cas A.1

Aquest exemple es defineix per tal de comprovar si l'algoritme és capaç de suggerir un canvi d'ubicació que permeti una millor assignació. Concretament, sabent com es comporta l'algoritme d'assignació, s'espera que l'algoritme d'ubicació suggereixi canvia a un dels pacients de més càrrega de la part superior per un dels pacients de la part inferior, per equilibrar les seves ubicacions i permetre una divisió perfecta de la unitat.

0 (0,0) 1	1 (0,1) 1	2 (0,2) 4	3 (0,3) 4	4 (0,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 1	6 (2,1) 1	7 (2,2) 1	8 (2,3) 1	9 (2,4) 1
-----------------	-----------------	-----------------	-----------------	-----------------

L'algoritme troba la següent ubicació de pacients

Chromosome with score 8.0

[0, 1, 2, **7**, 4, 5, 6, **3**, 8, 9]

amb un cromosoma d'assignació

[**7**, **5**, **8**, **6**, **9**, -1, **2**, **0**, **3**, **1**, **4**]

0 (0,0) 1	1 (0,1) 1	2 (0,2) 4	3 (0,3) 1	4 (0,4) 1
-----------------	-----------------	-----------------	------------------------	-----------------

5 (2,0) 1	6 (2,1) 1	7 (2,2) 4	8 (2,3) 1	9 (2,4) 1
-----------------	-----------------	------------------------	-----------------	-----------------

S'observa que l'algoritme efectua un intercanvi de pacients tal que es pot dur a terme una assignació equitativa de les càrregues de cures amb camins mínims.

Cas A.2

El segon exemple per la unitat en forma de passadís es defineix per observar el comportament de l'algoritme quan troba una ubicació heterogènia de pacients.

0 (0,0) 6	1 (0,1) 3	2 (0,2) 4	3 (0,3) 4	4 (0,4) 4
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 3	6 (2,1) 5	7 (2,2) 2	8 (2,3) 5	9 (2,4) 3
-----------------	-----------------	-----------------	-----------------	-----------------

L'algoritme troba la següent ubicació de pacients

Chromosome with score 8.5

[**6**, 1, 2, 3, 4, 5, **0**, 7, 8, 9]

amb un cromosoma d'assignació

[7, 8, 5, 9, 6, -1, 4, 1, 3, 2, 0]

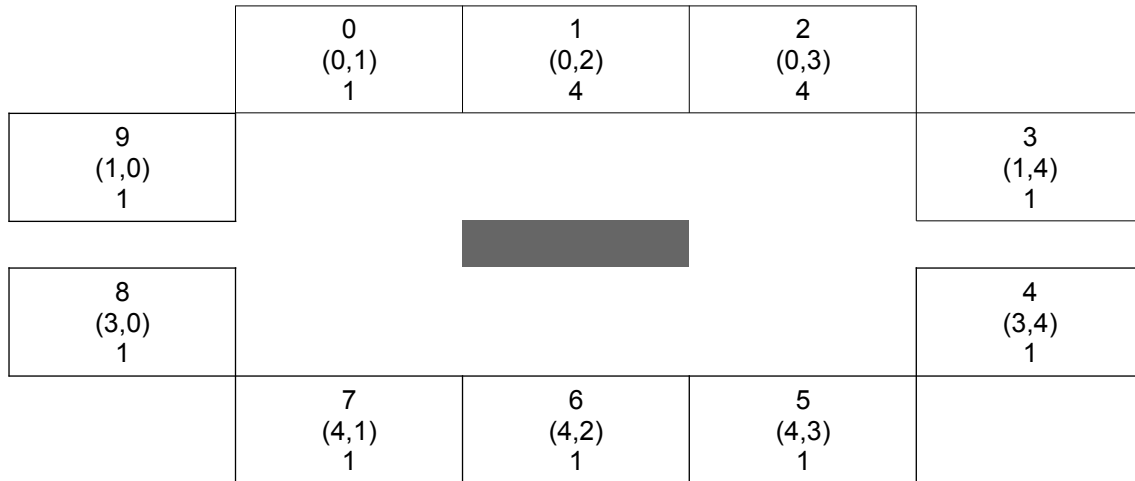
0 (0,0) 5	1 (0,1) 3	2 (0,2) 4	3 (0,3) 4	4 (0,4) 4
-----------------	-----------------	-----------------	-----------------	-----------------

5 (2,0) 3	6 (2,1) 6	7 (2,2) 2	8 (2,3) 5	9 (2,4) 3
-----------------	-----------------	-----------------	-----------------	-----------------

Aquesta nova distribució dels pacients permet a l'algoritme d'assignació una millor puntuació general (amb l'estructura original, la puntuació era de 9.5), derivada d'una millor distribució de la càrrega de cures ($\sigma = 0.5$).

Cas B.1

Aquest exemple es basa en el cas B.3 de l'algoritme d'assignació, del qual ja es coneix el seu comportament. Si no s'executa cap mena de canvi, l'algoritme d'assignació divideix la unitat en dues parts iguals, però s'espera que un dels pacients d'alta càrrega de la part superior sigui traslladat a la part inferior per generar uns arbres mínims més compactes.



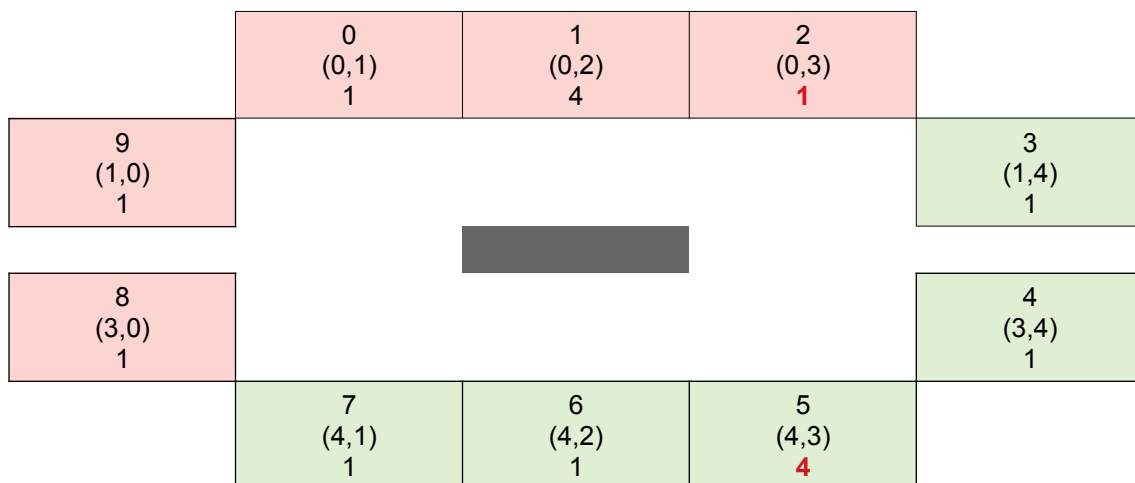
L'algoritme troba la següent ubicació de pacients

Chromosome with score 12.0

[0, 1, **5**, 3, 4, **2**, 6, 7, 8, 9]

amb un cromosoma d'assignació

[**1**, **2**, **8**, **0**, **9**, -1, **4**, **5**, **7**, **6**, **3**]



En aquest cas, l'algoritme d'ubicació torna a realitzar els canvis adients per permetre a l'algoritme d'assignació definir una càrrega de cures totalment equilibrada, amb uns arbres mínims òptims.

Cas B.2

Tal i com ja s'observa al cas A.2, l'exemple intenta observar i analitzar com treballa l'algoritme d'ubicació amb una ubicació heterogènia de pacients.

	0 (0,1) 6	1 (0,2) 3	2 (0,3) 4	
9 (1,0) 3				3 (1,4) 4
8 (3,0) 5				4 (3,4) 4
	7 (4,1) 2	6 (4,2) 5	5 (4,3) 3	

L'algoritme troba la següent ubicació de pacients

Chromosome with score 12.5

[0, 1, 2, 3, 4, 5, 6, **8**, **7**, 9]

amb un cromosoma d'assignació

[**0**, **9**, **3**, **1**, **2**, -1, **8**, **5**, **7**, **6**, **4**]

	0 (0,1) 6	1 (0,2) 3	2 (0,3) 4	
9 (1,0) 3				3 (1,4) 4
8 (3,0) 2				4 (3,4) 4
	7 (4,1) 5	6 (4,2) 5	5 (4,3) 3	

Es pot observar que, encara que s'ha executat un canvi d'ubicació, no existeix cap mena de millora real perquè els pacients segueixen assignats al mateix professional. En qualsevol cas, la distribució dels pacients original ja presentava una assignació òptima que l'algoritme d'assignació ja contemplava.

Cas C.1

El primer dels casos de la unitat en forma d'L intenta observar quina serà l'estratègia suggerida per l'algoritme d'ubicació.

0 (0,0) 1	1 (0,1) 4	2 (0,2) 4	3 (0,3) 1	
9 (2,0) 1	8 (2,1) 1	7 (2,2) 1		4 (2,4) 1
		6 (3,2) 1		5 (3,4) 1

L'algoritme troba la següent ubicació de pacients

Chromosome with score 10.0

[0, **5**, 2, 3, 4, **1**, 6, 7, 8, 9]

amb el cromosoma d'assignació

[**4**, **6**, **5**, **9**, **8**, **7**, -1, **2**, **0**, **3**, **1**]

0 (0,0) 1	1 (0,1) 1	2 (0,2) 4	3 (0,3) 1	
9 (2,0) 1	8 (2,1) 1	7 (2,2) 1		4 (2,4) 1
		6 (3,2) 1		5 (3,4) 4

Aquesta nova ubicació dels pacients permet assignar als pacients amb una càrrega de cures força equilibrada amb $\sigma = 1$ i dos arbres de cost 3 i 6 per un cost total de 9.

Cas C.2

Finalment, el darrer cas de proves de l'algoritme d'ubicació presenta una distribució heterogènia de pacients, per posar a prova la capacitat d'aquell per trobar una nova ubicació de pacients.

0 (0,0) 6	1 (0,1) 3	2 (0,2) 4	3 (0,3) 4	
9 (2,0) 3	8 (2,1) 5	7 (2,2) 2		4 (2,4) 4
		6 (3,2) 5		5 (3,4) 3

L'algoritme troba la següent ubicació de pacients

Chromosome with score 10.5

[0, 1, **4**, 3, **2**, 5, 6, 7, 8, 9]

amb el cromosoma d'assignació

[**4**, **5**, **6**, **7**, **8**, -1, **0**, **2**, **1**, **3**, **9**]

0 (0,0) 6	1 (0,1) 3	2 (0,2) 4	3 (0,3) 4	
9 (2,0) 3	8 (2,1) 5	7 (2,2) 2		4 (2,4) 4
		6 (3,2) 5		5 (3,4) 3

En aquest cas, el canvi de pacients no produeix cap mena de millora, generant una assignació molt equilibrada amb $\sigma = 0.5$ i dos arbres mínims de cost 5 cadascun per un total de 10.

En definitiva, l'algoritme d'ubicació sembla funcionar força bé per casos senzills, encara que no hauria de retornar cap mena de modificació dels pacients si aquesta no es tradueix en una millora real. Aquest fenomen es pot explicar pel fet d'utilitzar l'algoritme d'assignació com a funció de fitting. Cal tenir en compte que aquest no sempre proporciona el millor valor absolut sinó un que sigui prou bo fent que un cromosoma d'ubicació sigui avaluat per sota de la seva probabilitat de ser una solució acurada al problema.

5.2.2. Aplicació de l'algoritme en un cas real

Amb els paràmetres inicials, es fa una execució estàndard de l'algoritme per obtenir una determinació per una unitat amb 4 infermeres. Com que l'algoritme no s'executa massa generacions, no es pot contemplar cap mena d'evolució pel que fa als valors de mínim, màxim, mitja i mitjana dels valors de la funció de fitting i només val la pena analitzar si la proposta d'ubicació és factible. Val la pena comentar que l'execució d'aquest algoritme es duu a terme amb tècniques de concurrència per intentar millorar el temps global d'execució.

Així, després de 2 generacions i més de 24 minuts, s'obté el següent cromosoma d'ubicació

Chromosome with score 37.33012701892219

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, **24**, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, **10**, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36]

amb un cromosoma d'assignació

Chromosome with score 37.33012701892219

Assignment: 17, 29, 30, 31, 32, 33, 34, 35, 36. Load score: 407. Distance score: 14

Assignment: 8, 9, 13, 14, 15, 16, 18, 19, 20. Load score: 405. Distance score: 6

Assignment: 0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12. Load score: 399. Distance score: 6

Assignment: 21, 22, 23, 24, 25, 26, 27, 28. Load score: 411. Distance score: 7

Càrrega


11.1	10.1	9.1	8.1	7.1	6.1	5.1	4.1	3.1	2.1	1.1
64	53	46	67	45	38	28	39	46	43	0
	10.2	9.2	8.2	7.2	6.2	5.2	4.2	3.2	2.2	1.2
	35	29	44	35	35	33	29	39	47	38


12.1	46
13.1	61
14.1	44
15.1	35
16.1	52
17.1	55
18.1	61
19.1	57
20.1	46


27.1	23
26.1	33
25.1	62
24.1	75
23.1	51
22.1	44
21.1	44


Il·lustració 21: hotmap de les càrregues després de la nova ubicació

		11.1	10.1 10.2	9.1 9.2	8.1 8.2	7.1 7.2	6.1 6.2	5.1 5.2	4.1 4.2	3.1 3.2	2.1 2.2	1.1 1.2
12.1												
13.1												
14.1		27.1										
15.1		26.1										
16.1		25.1										
17.1		24.1										
18.1		23.1										
19.1		22.1										
20.1		21.1										


Infermera A


Infermera B


Infermera C


Infermera D

Il·lustració 22: assignació de pacients després de la nova ubicació

Visualment, es pot comprovar que l'algoritme assigna els pacients intentant fer els grups d'assignació els més compactes possibles, intentant mantenir una càrrega de cures equilibrada ($\sigma = 4.33$). Ara bé, per poder comprovar si l'algoritme d'ubicació s'ha executat de forma correcta, cal avaluar el resultat de l'assignació fent servir els valors de la càrrega de cures abans de dur a terme el canvi d'ubicació del pacient.

	Assignació	Càrrega total pre	Càrrega total post
Infermera A	17, 29, 30, 31, 32, 33, 34, 35, 36	407	407
Infermera B	8, 9, 13, 14, 15, 16, 18, 19, 20	405	405
Infermera C	0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12	396	399
Infermera D	21, 22, 23, 24, 25, 26, 27, 28	414	411
Desviació (σ)		6.42	4.33
Cost dels arbres		33	33
Total evaluació		39.42	37.33

Així, es pot veure que, efectivament, la nova ubicació permet obtenir un millor resultat a una assignació determinada. Ara bé, cal tenir present que quan es va executar l'algoritme d'assignació sobre el mateix exemple, el resultat de l'avaluació fou 40.04, que és pitjor resultat que el que s'ha obtingut amb l'assignació abans de dur a terme el canvi d'habitació (39.42). Així doncs, el

canvi d'ubicació realment no era necessari, atès que ja es tenia una millor assignació de pacients sense dur-ho a terme.

En conseqüència, l'algoritme d'ubicació sí que ofereix canvis coherents, però no sempre són pràctics o factibles, limitant molt la seva utilitat pràctica. Si, a més a més, s'afegeix el fet que el seu temps d'execució és força important, el seu potencial d'oferir una solució prou bona en un temps adequat queda totalment anul·lada i fa necessari replantejar el seu ús o execució.

6. Conclusions

6.1. Conclusions del treball

Per començar, el treball ha servit per plasmar una de les inquietuds de l'autor que és l'ús de tècniques d'intel·ligència artificial dintre de la gestió hospitalària i, més concretament, les tasques de gestió realitzades pels diferents professionals d'infermeria.

Pel que fa a l'aplicació dels algoritmes genètics dintre del camp de la gestió, aquest treball ha permès explorar les tendències actuals, generant noves idees pel futur, així com noves qüestions a resoldre. Gràcies als resultats obtinguts, s'ha pogut comprovar que hi ha aspectes de la gestió dels pacients, com, per exemple, l'assignació de professionals, que es poden gestionar perfectament amb l'ús d'aquest tipus d'algoritmes, proporcionant una alternativa interessant a la gestió habitual. Ara bé, també es pot apreciar que hi ha limitacions quant a la seva utilitat per altres tasques i és necessari replantejar-se el seu ús al món real o bé plantejar el problema des d'un altre punt de vista.

En qualsevol cas, de la gestió del projecte com a tal, es poden extreure les següents lliçons apreses:

- Planificar les tasques afegint sempre un o dos dies extres sobre l'estimació original.
- Descriure les tasques a executar amb la granularitat suficient per poder tenir una planificació el més realista possible.
- Intentar que no existeixin massa tasques que depenguin unes de les altres i permetre un alt grau de paral·lelització per tal de tenir més flexibilitat en cas de sorgir alguna classe d'imprevist.
- No confiar-se en el fet que es té el temps suficient. És millor anar treballant una mica dia a dia que haver de forçar la programació amb pocs dies.
- La gestió de la comunicació amb els interessats és molt important atès que permet un millor control de la planificació i manté un bon clima de treball.
- La política de control de gestions ha de quedar ben definida des del començament, on la branca master ha de reflectir tots els canvis una vegada han estat testejats i s'ha comprovat que no tenen cap mena d'errada.
- L'execució de proves ha de ser exhaustiva i s'ha de comprovar cada passa sense assumir que les passes trivials s'han executat de forma correcta.

6.2. Assoliment dels objectius plantejats

Dels quatre objectius plantejats, després d'analitzar els resultats obtinguts, es pot concloure que un dels objectius principals ha quedat totalment assolit, mentre que l'altra s'ha assolit de forma parcial.

Primerament, s'ha comprovat que l'algoritme genètic d'assignació proporciona assignacions de pacients molt equilibrades amb grups de pacients molt compactes, sempre tenint en compte la distribució dels pacients, oferint noves possibilitats a l'hora de gestionar les unitats d'hospitalització.

En segon lloc, l'algoritme genètic d'ubicació, encara que proporciona suggeriments de noves ubicacions correctes, té limitada la seva utilitat en tant ho fa en un temps d'execució força elevat i que no sempre tenen una utilitat real. És per aquest fet que cal dur a terme una revisió de l'algoritme com part d'una línia de treball futura. Encara així, cal recordar que l'algoritme treballa força bé quan tracta amb grups de pacients petits, encara que també cal revisar que els canvis suggerits tinguin una aplicació realista. En qualsevol cas, el motiu de no haver assolit l'objectiu, no es pot associar a una manca de planificació, sinó més aviat a unes expectatives massa optimistes sobre la seva viabilitat.

El següent punt a considerar és l'assoliment dels objectius secundaris on, malauradament, només s'ha pogut tractar l'opció de la computació en paral·lel. Concretament, s'ha utilitzat la llibreria integrada de python de multiprocessament, fent servir el que s'anomena un pool de treballadors on, a cada un dels membres d'aquest pool se'ls hi assigna un procés a executar. En el cas d'aquest TFG, a cada membre del pool se l'hi ha assignat la tasca d'avaluar un cromosoma de l'algoritme d'ubicació, ocupant el 100% del temps d'un processador en aquesta tasca. En altres paraules, cada membre del pool ha executat un algoritme d'assignació complet.

Per contra, l'objectiu d'avaluar la codificació de restriccions addicionals ha quedat sense resoldre per una manca de temps de desenvolupament, quedant pendent com una línia futura de treball.

6.3. Seguiment de la planificació

D'entrada és necessari comentar que, en general, s'han executat les diferents tasques seguint la planificació establerta. Encara així, hi ha certs punts a comentar que han generat modificacions del calendari:

- A petició del consultor, es va afegir una fita de revisió de la temporalització entre les fases I i II de codificació de l'algoritme per assegurar un bon seguiment de la programació.
- Dintre de la segona fase d'implementació del codi, va ser necessari dur a terme un petit canvi de planificació, on es va avançar tota la part d'implementació en concurrència, per intentar millorar els temps d'execució prevists a la part de l'algoritme d'ubicació.
- A l'hora de redactar la memòria, no es varen complir amb les expectatives de treball planificades per un excés de confiança. A més a més, durant la redacció de la memòria es varen trobar diferents errors d'execució del codi i es va necessitar dur a terme una nova revisió d'aquest, afegint encara més retard a la programació.

En resum, es pot considerar que la metodologia i planificació han estat adequades, però és necessari respectar la programació per evitar problemes derivats per imprevists.

6.4. Futures línies de treball

En un principi és necessari fer una distinció sobre el tipus de línies de treball a considerar a partir del treball executat en aquest TFG. Per una banda es poden considerar totes les línies destinades a millorar l'execució com a tal i, per l'altra, totes les línies orientades a ampliar la seva aplicabilitat.

Dintre del primer grup de línies de treball, es poden considerar:

- Milllores dels resultats
 - Comportament de l'algoritme d'assignació amb llits sense ocupació: actualment, l'algoritme considera els llits sense ocupació com un llit sense càrrega de treball. Ara bé, encara que no té cap càrrega de treball, atès que l'assigna a un professional, l'inclou dintre del graf per calcular el seu arbre mínim, augmentant el cost sense existir una necessitat real de fer-ho (realment, el professional no té cap mena de treball). Caldria modificar l'algoritme per tenir en compte aquest fet.
 - Comportament de l'algoritme d'ubicació quan no existeixen millores: ja s'ha comentat en un apartat anterior el fet que l'algoritme d'ubicació ha de considerar no aportar canvis si aquests no milloren l'assignació.
- Millores d'execució
 - Millores del temps d'execució: encara que l'ús de la concurrència permet reduir els temps d'execució, pot ser interessant avaluar l'impacte en l'àmbit d'execució amb l'ús de maquinari més potent (més velocitat o més nuclis).

Pel que fa al segon grup de línies de treball, cal destacar:

- Implementació de limitacions: ara per ara, l'algoritme només contempla càrregues de cures a l'hora d'assignar sense tenir en compte altres aspectes que són important per a la seva aplicabilitat en el món real.
 - Limitacions de la ubicació per sexe: és necessari evitar fer un canvi d'ubicació tal que dues persones del sexe contrari comparteixin habitació.
 - Limitacions per aïllament: alguns pacients necessiten estar ubicats tots sols a una habitació (per motius de salut o seguretat) i, per tant, no poden compartir habitació amb altres pacients o, en funció de la unitat, no poden deixar l'habitació que ocupen.
 - Altres tipus de limitacions: cal implementar opcions que tinguin en compte altres aspectes com la incompatibilitat entre pacients, pacients VIP...
- Ús en unitats petites o subunitats: atès que l'algoritme d'ubicació sembla treballar bé quan té conjunts petits de pacients, una línia de treball pot consistir en l'avaluació i viabilitat d'ús en unitats de pocs pacients o, fins i tot, plantejar dividir les unitats per zones o subunitats i aplicar l'algoritme d'ubicació a cada una d'elles.
- Avaluació amb tridimensionalitat: encara que l'algoritme està dissenyat per establir assignacions de pacients a una unitat, l'organització d'un hospital pot contemplar la gestió de professionals assignats a diferents unitats situades en plantes diferents. Una possible extensió d'aquest algoritme podria avaluar la verticalitat per tal d'incloure aquest fet a l'hora d'assignar pacients.

7. Glossari

Problema NP-difícil: un problema NP-díficil és un problema que pertany a la classe de complexitat NP-díficil, que és el conjunt dels problemes de decisió tals que tot problema de la classe NP es pot transformar en un problema NP-díficil en temps polinòmic. Es diu que aquests problemes són, com a mínim, tan difícils com els NP.

Java: llenguatge de programació pensat com un llenguatge orientat a objectes. És un dels llenguatges de programació més utilitzats, molt flexible i potent. Una de les seves característiques és que es tracta d'un llenguatge multi-plataforma

Python: llenguatge de programació d'alt nivell i propòsit general. La seva filosofia de disseny busca llegibilitat en el codi. Suporta paradigmes com la programació orientada a objectes, la programació imperativa i també funcional o procedimental.

Git: programari de sistema de control de versions pensat en l'eficiència i confiabilitat de manteniment de versions d'aplicacions.

Linux: família de sistemes operatius formats pel kernel Linux juntament amb les utilitats GNU.

Windows: sèrie de sistemes operatius produïts per Microsoft.

Mac OS: versió del sistema operatiu que utilitzen els ordinadors Macintosh, basada en un nucli Unix.

Distància euclidiana: és la distància ordinària entre dos punts que es mesuraria amb un regle i ve donada pel teorema de Pitàgores.

Distància Manhattan: també coneguda com a geometria del taxista o mètrica del taxista. Part de la premissa que, per anar d'un punt a un altre només es pot seguir la direcció vertical o horitzontal en la qual la funció de distància entre dos punts és la suma de les diferències absolutes de les seves coordenades.

Permutacions: una permutació és una mostra d'elements que es diferencien les unes de les altres per l'ordre de col·locació dels elements.

Desviació estàndard: és una mesura de diversitat que s'usa en estadística. Mostra quanta variació hi ha respecte a la mitjana. Una desviació baixa indica que els punts de dades tendeixen a ser proper a la mitjana.

Graf complet: és un graf simple on una aresta connecta tots els parells de vèrtex.

Arbre: és un graf en el qual dos vèrtex estan connectats per exactament un camí.

8. Bibliografía

WikiNSP: NA, Nurse scheduling problem, 2018,
https://en.wikipedia.org/wiki/Nurse_scheduling_problem

Chang09: Chang-Chun Tsai, Sherman H.A. Li, A two-stage modeling with genetic algorithms for the nurse scheduling problem, 2009

Moz07: Margarida Moz, Margarida Vaz Pato, A genetic algorithm approach to a nurse rostering problem, 2007

Kom14: Komgrit Leksakul, Nurse scheduling using genetic algorithm, 2014

Aza14: A. Azadeh, M. Hosseinabada Farahani, S. Torabzadeh, M. Baghersad, Scheduling prioritized patients in emergency department laboratories, 2014

Vili97: Vili Podgorelec, Peter Kokol, Genetic Algorithm Based System for Patient Scheduling in Highly Constrained Situations, 1997

Man99: KF Man, KS Tang, S Kwong, Genetic Algorithms Concepts and Designs, 1999

Hol15: M. K. Holst. Optimal Hospital Layout Design. Ph.d.-serien for Det Teknisk-Naturvidenskabelige Fakultet, Aalborg Universitet. 2015. <https://doi.org/10.5278/vbn.phd.engsci.00064>

Lott15: Steven Lott, Functional Python Programming, 2015

Sig96: Instituto Nacional de la Salud, Gestión analítica hospitalaria (Proyecto Signo), 1996

Car13: Carmona-Monge FJ, Rollán Rodríguez GM et al, Evaluation of the nursing workload through the Nine Equivalents for Nursing Manpower Use Scale and the Nursing Activities Score: a prospective correlation study, 2013

9. Annexos

A. Manual d'instal·lació

A l'hora d'executar l'algoritme és necessari tenir instal·lada la versió 3.6 o superior de Python. No és necessari cap altra llibreria addicional. Encara que es pot executar des de qualsevol sistema operatiu, es recomana utilitzar qualsevol versió de Linux.

Descomprimir l'arxiu a qualsevol carpeta. Es poden veure dues carpetes:

- src: conté el codi font del TFG en dues carpetes addicionals
 - assign: només executa l'algoritme d'assignació
 - ubic: executa tot l'algoritme
- test: conté tots els sets de proves

Abans d'executar el codi, és necessari copiar els arxius de pacients i habitacions corresponents a un dels sets de proves a la carpeta del codi que s'executarà.

A tall d'exemple, si es vol executar l'algoritme d'assignació sobre el set de proves del cas C.4, cal copiar els fitxers que es troben a la carpeta "Cas C4 assign" a la carpeta "assign" de la carpeta del codi font.

Per executar el codi, cal accedir a una consola de comandament i anar a la carpeta on es troba el codi font a executar.

L'ordre de comandament segueix l'estructura "python3 TFG.py rooms patients" on "rooms" és el nom de l'arxiu de l'estructura física de la unitat i "patients" el nom de l'arxiu que conté les càrregues de pacients.

Per exemple, per executar el set de proves anterior, cal executar el següent codi a la carpeta TFG/src/assign: python3 TFG.py rooms.txt patients.txt

```
daniel@daniel-VirtualBox:~/TFG/src/assign$ python3 TFG.py rooms.txt patients.txt
```

Si es té habilitada l'opció "verbose", apareix a la pantalla informació sobre l'execució de l'algoritme. A més a més, una vegada finalitza la seva execució es generen diferents arxius de log que es poden consultar per veure com ha treballat l'algoritme:

- Execució de l'algoritme d'assignació:
 - Arxius amb format "_param": guarda els paràmetres que s'han utilitzat per a l'execució de l'algoritme.
 - Arxius amb format "_log": guarden els valors mínims, màxims, sumatori, mitja i mitjana de la funció de fitting per cada generació.
 - Arxius amb format "_chrom": guarden els millors cromosomes que es van generant.
- Execució de l'algoritme d'ubicació (es genera una gran quantitat d'arxius en funció dels paràmetres seleccionats):

- Arxius amb format "*_param_asig" i "*_param_ubi": guarden els paràmetres que s'han utilitzat pe a l'execució dels algoritmes d'assignació i ubicació respectivament.
- Arxius amb format "*_log_asig" i "*_log_ubi": guarden els valors mínims, màxims, sumatoris, mitges i mitjanes de la funció de fitting per cada generació dels algoritmes d'assignació i ubicació.
- Arxius amb format "*_chrom_asig" i "*_chrom_ubi": guarden els millors cromosomes que es van generant pels algoritmes d'assignació i ubicació.

B. Paràmetres d'ús

Es poden modificar els diferents paràmetres disponibles dintre de l'arxiu ga.py per tal de modificar el comportament dels algoritmes. També es poden modificar a l'arxiu TFG.py, però només modificaran l'algoritme principal que s'executi.

- pool_size (enter): determina la mida del pool d'individus.
- p_cross (enter): probabilitat que un cromosoma es reproduïxi amb un altre. No té cap efecte sobre l'algoritme d'ubicació.
- p_muta (decimal): probabilitat que un cromosoma fill muti abans de ser inclòs al nou pool. No té cap efecte sobre l'algoritme d'ubicació
- elitism (boolean): habilita l'elitisme, és a dir, el millor cromosoma serà inclòs al nou pool sense cap mena d'alteració.
- endogamy (boolean): habilita l'endogàmia, és a dir, el millor cromosoma sempre és un dels pares a l'hora de realitzar encreuaments. No té cap efecte sobre l'algoritme d'ubicació.
- max_gen (enter): determina el nombre màxim de generacions d'un algoritme.
- max_change (enter): determina al nombre màxim de generacions sense una millora.
- nurses (enter): determina el nombre d'infermeres de la unitat. El seu valor ha de ser superior a 1.
- w_loads (decimal): coeficient d'ajust per les càrregues de cures. Pot augmentar o disminuir el grau que aquest valor té sobre la funció de fitting.
- w_dist (decimal): coeficient d'ajuts pels arbres mínims. Pot augmentar o disminuir el grau que aquest valor té sobre la funció de fitting.