

Treball final de grau

Estudi: Grau en Enginyeria Informàtica

Títol: Aplicació per al càlcul de propostes de trasplantament creuat de ronyó amb donant viu considerant fallades de darrera hora

Document: Memòria

Alumne: Eloi Soldevila Sala (u1905688)

Tutor: Mateu Villaret / Francesc Castro

Departament: IMAE

Àrea:LSI

Convocatòria (mes/any) Setembre de 2018

Índex

1. Introducció 5

- 1.1 Marc de treball 5
- 1.2 Història dels trasplantaments renals 5
- 1.3 Conceptes previs 7
 - 1.3.1 Trasplantament creuat 7
 - 1.3.2 Donant altruista i cadenes 9
 - 1.3.3 Programes d'intercanvi renal a Europa (KEP's) 11
 - 1.3.4 Intercanvis renals internacionals 12
- 1.4 Nomenclatura 13
- 1.5 Motivacions 15
- 1.6 Propòsit 15
- 1.7 Objectius 15
- 1.8 Punt de partida i abast del projecte 15

2. Estudi de viabilitat 18

3. Metodologia 19

4. Planificació 20

5. Requisits del sistema 24

- 5.1 Requisits 24
- 5.2 Funcionalitats opcionals 25

6. Estudis i decisions 26

- 6.1 Llenguatge 26
- 6.2 Llibreries 26
- 6.3 Compilador i entorn d'execució 26
- 6.4 Explicació dels *solvers* 26
- 6.5 Comparació de solvers 27

7. Anàlisi i disseny del sistema 30

- 7.1 Anàlisi 30
 - 7.1.1 Model de dades 30
 - 7.1.2 Model de processos 31
- 7.2 Disseny 33
 - 7.2.1 Model físic de dades 33
 - 7.2.2 Model físic de processos 34

8. Implementació i proves 38

- 8.1 Model de classes 38
 - 8.1.1 Package poolMembers 38
 - 8.1.2 Package pool 39
 - 8.1.3 Package poolOptimizer 39
 - 8.1.4 Interfície d'usuari (GUI) i classe principal executable (dss) 40
- 8.2 Algorisme de càlcul de cicles 41
- 8.3 Problemes i solucions apareguts 42
- 8.4 Proves 42
 - 8.4.1 Package pool 42
 - 8.4.2 Package poolOptimizer 43
 - 8.4.3 Interfície d'usuari (GUI) 43

9. Implantació i resultats 44

- 9.1 Implantació 44
- 9.2 Resultats 44

10. Conclusions 58

- 10.1 Assoliment d'objectius 58
- 10.2 Assoliment dels requisits del programa desenvolupat 58
- 10.3 Desviacions en la planificació original 59
- 10.4 Crítica dels resultats 59

11. Treball futur 61

12. Bibliografia 62

13. Annexos 64

14. Manual d'usuari i/o instal·lació 65

- 14.1 Manual d'instal·lació 65
- 14.2 Manual d'usuari 66
 - 14.2.1 Moure's per les pestanyes 67
 - 14.2.2 Pestanya «Home» 68
 - 14.2.3 Pestanya «Pools» 69
 - 14.2.4 Pestanya «Selections» 71
 - 14.2.5 Pestanya «Failures» 73
 - 14.2.6 Pestanya «Repairs» 75
 - 14.2.7 Pestanya «History» 77
 - 14.2.8 Guardat de fitxers 80
 - 14.2.9 Carrega de fitxers 81

1. Introducció

1.1 Marc de treball

Aquest projecte s'emmarca, d'una banda, dins la col·laboració d'alguns professors del Departament d'Informàtica, Matemàtica Aplicada i Estadística de la Universitat de Girona, amb l'Organització Nacional de Trasplantaments, en concret, en el seu Kidney Exchange Program, que gestiona els procediments de trasplantament renals creuats amb donant viu; d'altra banda, el projecte també s'emmarca dins una acció europea que pretén acostar els KEP's dels diferents països de la Unió Europea i cercar-ne vies d'estandardització dels seus criteris i procediments.

1.2 Història dels trasplantaments renals

El primer trasplantament renal, entre humans, realitzat amb èxit va ser a l'*Hospital Brigham* de Boston (Estats Units) l'any 1954. Aquest va ser un trasplantament renal amb donant viu (TRDV), entre bessons [1].

A Espanya, el primer trasplantament renal, entre humans, realitzat amb èxit va ser l'*Hospital Clínic* de Barcelona l'any 1965. Aquest va ser un trasplantament renal amb donant mort (TRDM) [1]. Aquest tipus de trasplantament es realitza quan hi ha disponible algun ronyó procedent d'algun donant mort, que té l'òrgan requerit en bon estat, i prèvia autorització feta pel finat quan estava viu o de la seva família un cop mort.

Els TRDV són un tractament establert a Espanya des dels anys seixanta, però que va mantenir un escàs nivell d'activitat fins a l'any 2000, data en què començà un increment progressiu en el nombre de procediments (Fig.1) i hospitals que realitzen aquesta teràpia [2].

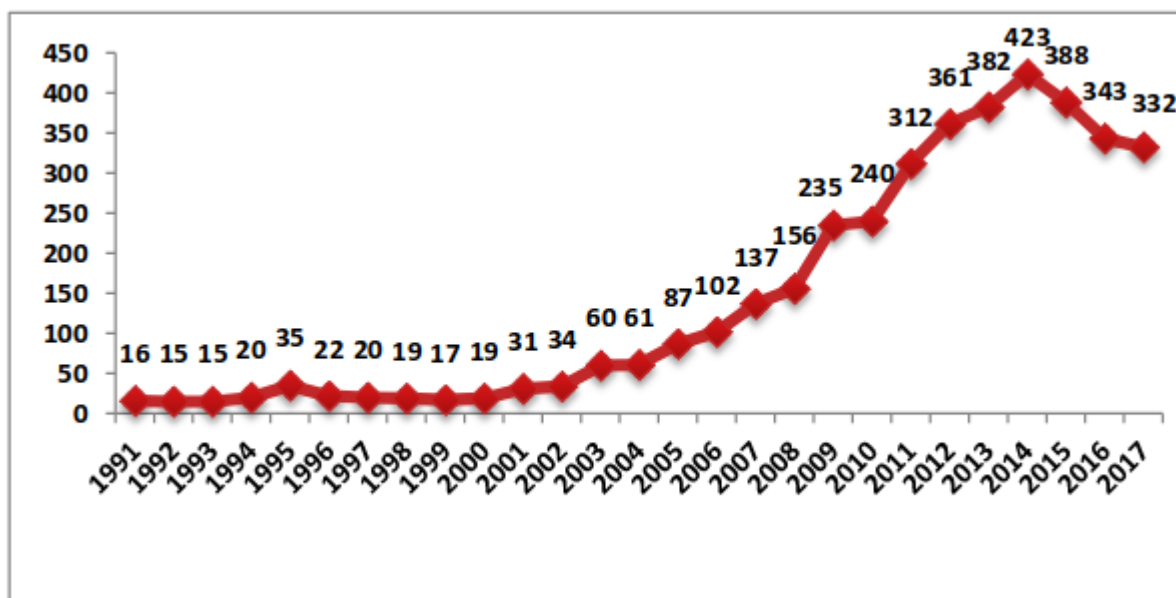


Fig. 1. Activitat de trasplantament renal de donant viu. Espanya 1991-2017 [3]

Aquest tipus de trasplantament es realitza quan hi ha disponible algun ronyó procedent d'algun donant viu. Per tant, necessiten una donació altruista o que el pacient tingui alguna persona (normalment un familiar) disposada a donar el seu ronyó, perquè ell/a en rebi un.

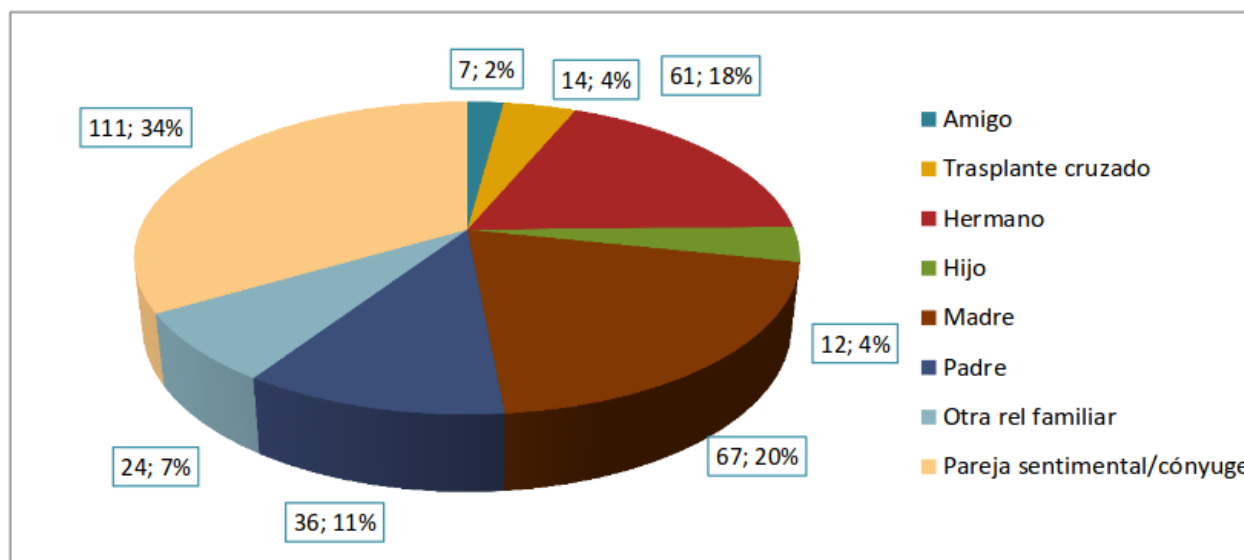


Fig. 2. Trasplantament renal de viu: relació dels pacients amb el seu/s donants vinculats. Espanya 2017. [3]

Els motius pels que el trasplantament renal de donant viu està emergint són diversos i poden englobar-se en 4 apartats: [2]

- **Millors resultats**, ja que s'obté en mitjana, una major supervivència de l'òrgan trasplantat i del pacient. El bon estat de salut del donant i l'absència de danys que es produeixen al ronyó amb la mort encefàlica són alguns dels motius d'aquesta millora en els resultats.
- **Necessitat de trasplantament de viu**, ja que la flexibilització dels criteris d'entrada a la llista d'espera comporta una major dificultat d'atendre tota la demanda de trasplantament, sobretot en pacients joves, en que les possibilitats d'obtenir un òrgan adequat a la seva edat són menors, degut a que el perfil de donant cadàver cada cop és de més edat, donat l'increment en l'esperança de vida de la població.
- **Millora en la seguretat del donant**, ja que l'excel·lent avaluació i seguiment dels donants (basats en estàndards internacionals), i l'ús de tècniques quirúrgiques menys invasives, comporta un baix índex de complicacions i una esperança de vida dels donants similar a la de la població general.
- **Possibilitat de trasplantament entre pacient i donant viu incompatibles**, degut a la major especialització en l'àmbit, per l'esforç en formació realitzat pels equips de trasplantament i coordinadors, junt

amb programes específics de donació renal creuada, i dessensibilització de pacients.

Actualment conviuen les dues tipologies de trasplantament TRDM i TRDV (Fig. 3), i així seguirà sent, alguns dels pacients estan inclosos alhora a la llista d'espera de donant cadàver i a la llista de donació de viu.

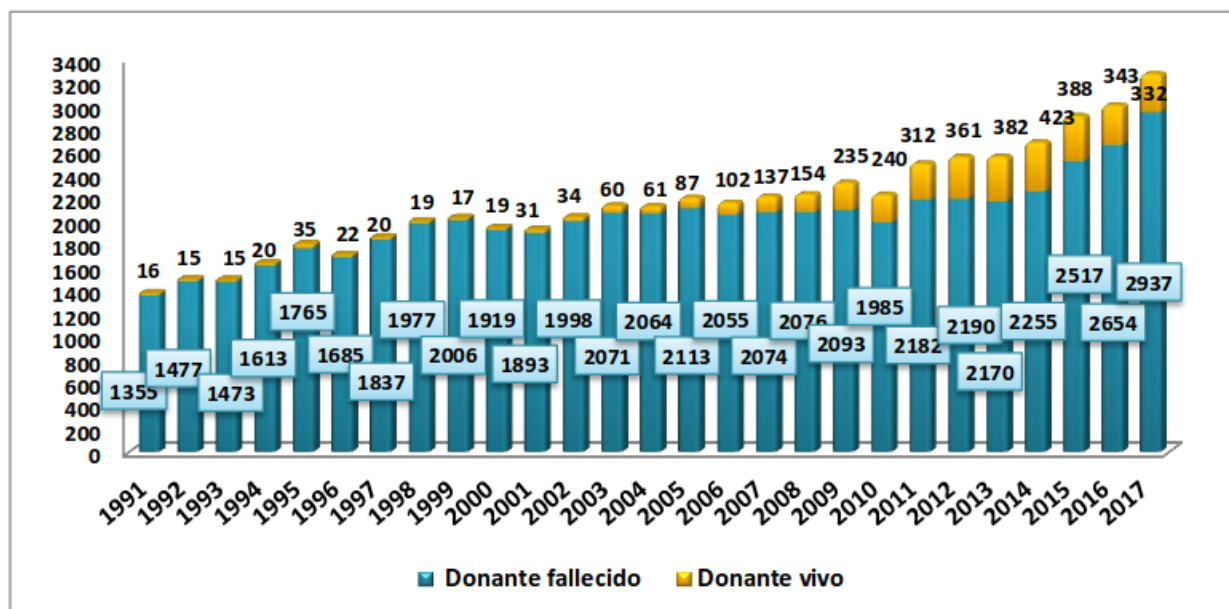


Fig. 3. Activitat de trasplantament renal de donant mort i donant viu. Espanya 1991-2017 [3]

1.3 Conceptes previs

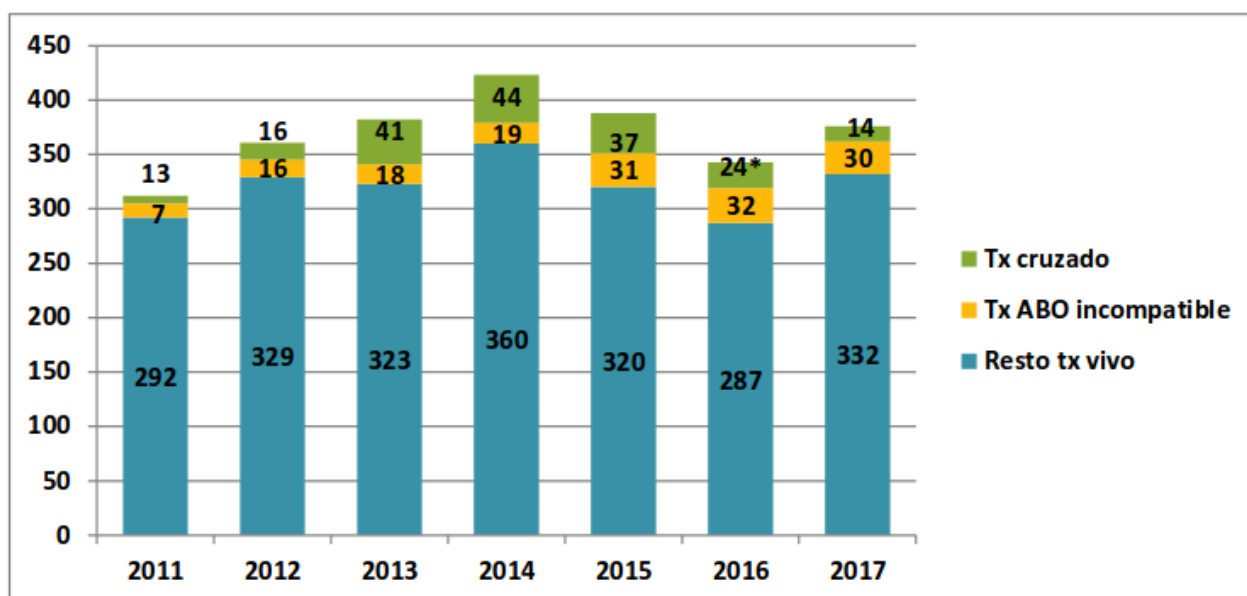
Aquest punt, així com l'1.1 Marc de treball i l'1.4 Nomenclatura, segons la guia de redacció de la memòria del TFG, aniria dins del capítol 5. He decidit suprimir el capítol 5, posant tot el seu contingut a la introducció atès que estem en un àmbit complex, en que es treballa amb molts conceptes diferents, per tal de facilitar la comprensió del lector en els capítols següents a aquest. Sense aquests conceptes no es podria fer una descripció detallada de manera clara i entenedora.

1.3.1 Trasplantament creuat

A Espanya es va dur a terme el primer entre una parella de l'*Hospital Clínic* de Barcelona i una altra de l'*Hospital Virgen de las Nieves* de Granada l'any 2010. [1]

Són una part dels TRDV, l'any 2017 van suposar un 4% aproximadament (Fig. 4), pensats pels casos que el pacient i el seu/s donant/s vinculat/s (d'ara endavant parella), són incompatibles, però que en alguns casos (segons el KEP del país) també inclouen parelles compatibles. Aquestes solen tenir màxima

prioritat i es beneficien de la possibilitats de trobar-los un donant amb un perfil més adequat al seu perfil (menor diferència d'edat per exemple).



**Incluye un trasplante cruzado que además fue ABO incompatible*

Fig. 4. Trasplantaments renals de viu en diferents modalitats. Espanya 2011-2017 [3]

Es tracta de crear una recopilació de parelles (en endavant *pool*) per tal de fer intercanvis de ronyó entre elles (en endavant cicles). Aquests cicles poden ser de diferents mides segons el nombre de donacions que el formen (Fig. 5). A Espanya, així com a la majoria de països europeus, es limita la mida a dos i tres per temes logístics, i perquè en general són més fàcils de dur a terme com més petits siguin, és a dir, millor fer tres cicles de mida dos que dos cicles de mida tres.

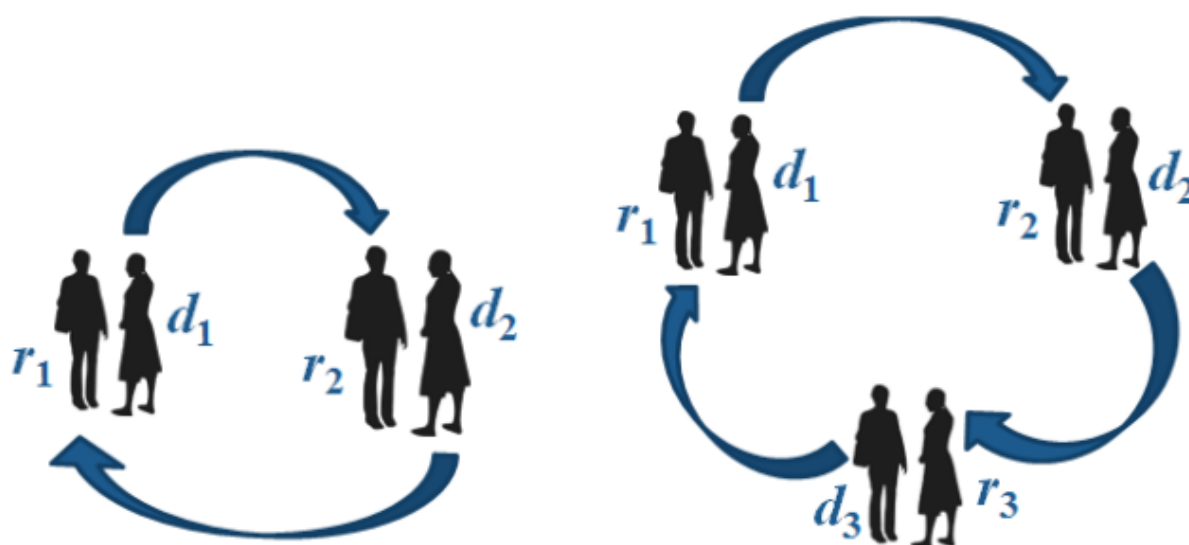


Fig. 5. Cicle de 2 donacions i cicle de 3 donacions [4]

A grans trets, el procés que se segueix per a la realització de trasplantaments creuats, al KEP espanyol (cada 4 mesos aproximadament) és:

- i. A partir del *pool* de pacients i donants vius vinculats, es calculen totes les possibles compatibilitats entre elles.
- ii. A partir d'aquestes compatibilitats es calcula el conjunt de tots els possibles cicles (fins a la mida màxima que es contempla, 3 actualment).
- iii. A partir d'aquest conjunt de cicles teòricament viables se selecciona (planifica) segons uns criteris establerts, un subconjunt de cicles de intercanvi disjunts entre ells (és a dir, sense cap pacient comú) de cares a ser implementats si és possible.
- iv. S'intenta fer tots els cicles (trasplantaments) planificats.
- v. Si hi ha hagut fallades (donants o pacients no disponibles en aquell moment o bé incompatibilitats de darrera hora entre donant i pacient) i encara hi ha possibles cicles a realitzar es torna al punt iii, però ara el conjunt de cicles sobre el que es selecciona (seleccionables), són els cicles que no tenen fallades ni pacients ja trasplantats, és a dir els cicles que després de les fallades ocorregudes i els trasplantaments fets segueixen sent teòricament viables, donant prioritat màxima a tots els pacients seleccionats en la planificació anterior, per exemple, si un pacient havia estat seleccionat per fer-li el trasplantament i el seu cicle ha fallat, però no ell, si existeix algun altre cicle teòricament viable (tenint en compte les fallades ocorregudes i els pacients ja trasplantats) que el contingui, aquest serà segur triat en la replanificació.

Els KEP's europeus segueixen uns processos molt similars a aquest, però canvien, entre d'altres qüestions, en els criteris i en el mètode utilitzat per calcular les seleccions segons aquests criteris.

1.3.2 Donant altruista i cadenes

Existeixen persones que, per altruisme, volen donar un dels seus ronyons al *pool* de trasplantament de ronyó provinent de donant viu. En aquests casos, el ronyó donat s'utilitza per engegar una *cadena de trasplantaments*, encadenant un cert nombre de trasplantaments; el donant vinculat al darrer pacient de la cadena queda disponible per engegar una nova cadena en un altre moment (com si es tractés d'un nou donant altruista), si bé, en altres casos, es prefereix passar el seu ronyó al *pool* de pacients que esperen un ronyó d'un donant cadàver.

A Espanya es va dur a terme la primera cadena l'any 2011, entre la *Fundació Puigvert* de Barcelona i un hospital de Granada (Fig. 6). Tanmateix, el nombre de donants altruistes (també anomenats samaritans) a Espanya és reduït en comparació al d'altres països. També cal dir que no tots els KEP's contemplen les cadenes amb donant altruista.

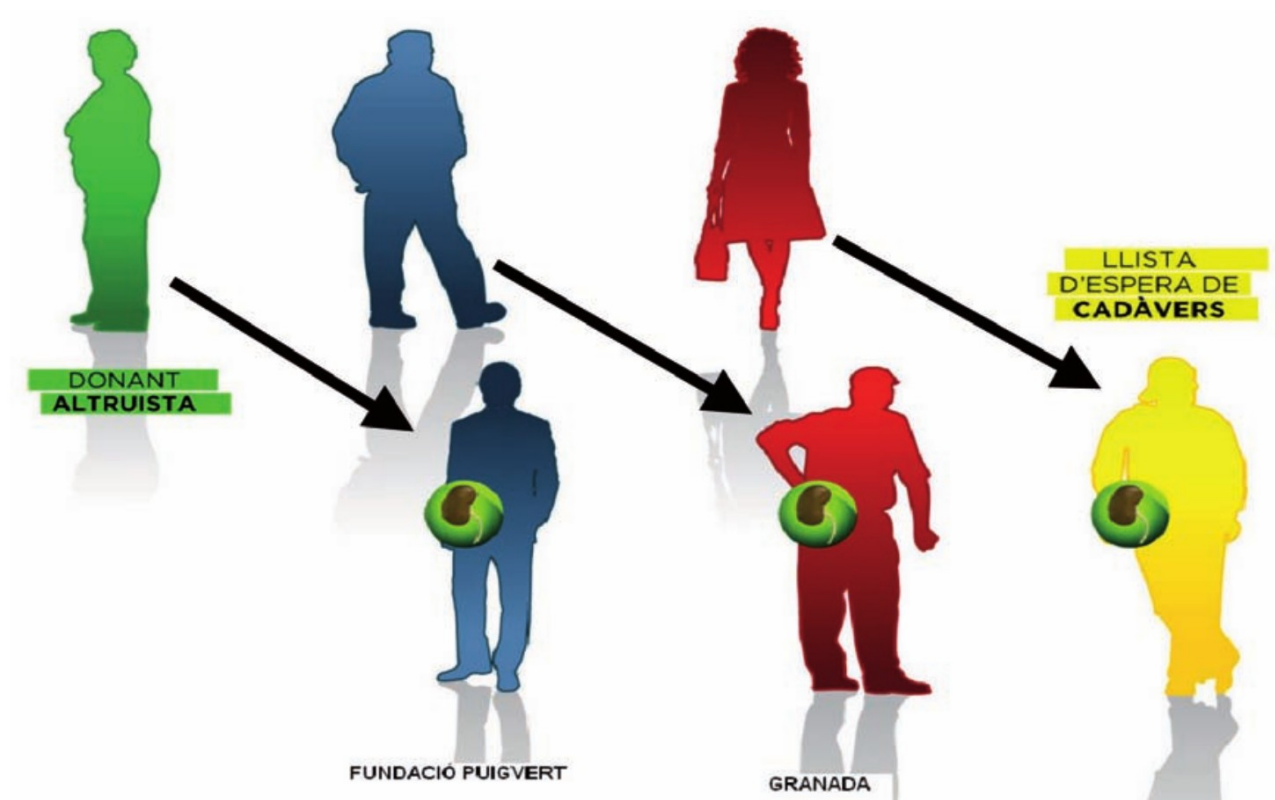


Fig. 6. Primera cadena de trasplantaments a l'Estat espanyol [1]

En aquest projecte, m'he centrat en els cicles, que és el denominador comú de tots els KEP's europeus, però contemplant la possibilitat d'incorporar en el futur les cadenes.

1.3.3 Programes d'intercanvi renal a Europa (KEP's)

El primer intercanvi renal dut a terme a Europa via KEP va tenir lloc al KEP Belga-Holandès l'any 2004. [4]

Molts països europeus ja tenen implementat, des de fa anys o recentment, el seu propi KEP, o bé es troba en fase de preparació. Hi ha alguns conformats per més d'un estat (República Txeca i Àustria per exemple). Al mapa següent es mostra l'estat dels KEP's en els diferents països europeus:

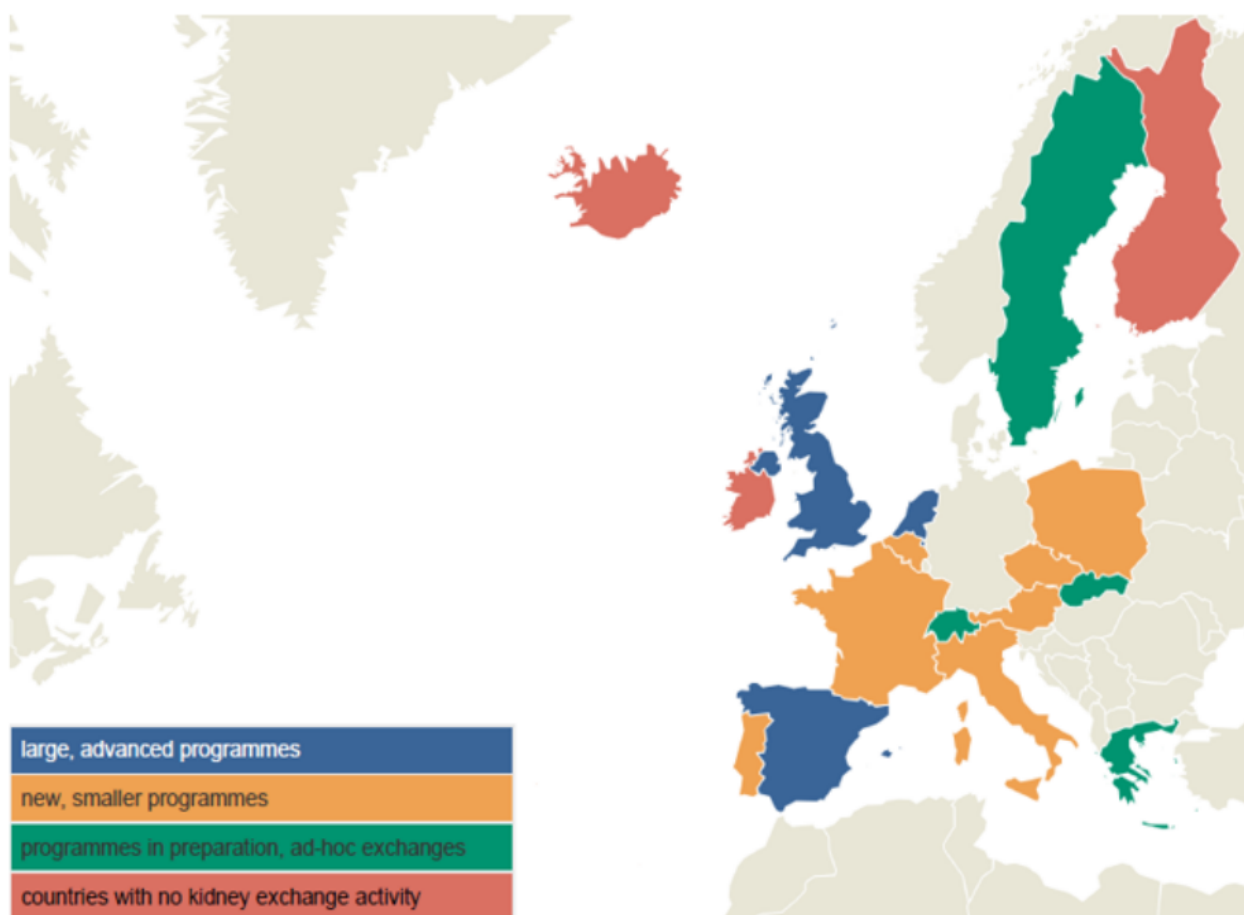


Fig. 7. Programes d'intercanvi renal a Europa [4]

Donat que cada país té implementat el seu propi KEP, les característiques i criteris que s'hi utilitzen, com ara la mida màxima dels intercanvis planificats, varien en cada cas. A molts països, d'altra banda, es realitzen investigacions per tal d'intentar millor l'efectivitat dels procediments d'intercanvi. A continuació mostro una taula resum de l'ús i característiques dels principals criteris utilitzats per cada país:

| | Austria | Belgium | Czech Republic | France | Italy | Netherlands | Poland | Portugal | Spain | UK | Sweden | Switzerland |
|---------------------------------------------------------------|---------|---------|----------------|--------|-------|-------------|--------|----------|-------|----|--------|-------------|
| First exchange in KEP: 20XX | 13 | 14 | 11 | 14 | 07 | 04 | 15 | 13 | 09 | 07 | | |
| Altruistic donor chains possible? | ✓ | x | ✓ | x | ✓ | ✓ | x | ✓ | ✓ | ✓ | x | x |
| Compatible pairs/ couples participate? | x | x | ✓ | x | ✓ | ✓ | ✓ | x | ✓ | ✓ | ✓ | ✓ |
| Multiple donors register for one patient? | ✓ | x | ✓ | x | ✓ | x | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Incompatible transplants allowed within KEP? | ✓ | x | ✓ | x | x | x | x | x | ✓ | ✓ | ✓ | ✓ |
| Single lab carries out cross matching after virtual matching? | ✓ | ✓ | ✓ | x | x | ✓ | x | x | x | x | x | x |
| Simultaneous surgery required for an exchange in KEP? | ✓ | ✓ | x | ✓ | ✓ | ✓ | x | ✓ | ✓ | ✓ | ✓ | ✓ |
| Organs usually travel (O) or donors (D)? | D | O | - | O | O | D | O | O | O | O | O | D |
| Matching process every x months (NR=not regular) | NR | NR | 3 | 3 | NR | 3 | 1 | 3 | 4 | 3 | na | 3 |
| Longest exchange already conducted | 3 | 3 | 7 | 2 | 2 | 4 | 3 | 3 | 3 | 3 | na | na |
| Longest chain already conducted | na | na | 6 | na | 6 | 2 | na | na | 6 | 3 | na | na |

Fig. 8. Taula resum de les principals característiques dels KEP's europeus [4]

Aquestes discrepàncies en les característiques poden suposar una problemàtica afegida en la planificació d'intercanvis internacionals, que s'estan començant a realitzar. Seria per tant molt positiu crear una estandardització en els procediments i criteris usats en els KEP's europeus, objectiu que persegueix l'acció europea COST en què s'emmarca aquest projecte.

1.3.4 Intercanvis renals internacionals

Aquest mateix estiu de 2018 s'ha realitzat el primer intercanvi renal internacional del sud d'Europa, dins un *pool* amb un total de 113 parelles, 79 espanyoles, 19 portugueses i 15 italianes. L'intercanvi, àmpliament difós pels mitjans, s'ha realitzat entre una parella espanyola i una d'italiana [5].

El trasplantament renal creuat internacional s'havia realitzat en dues ocasions abans d'aquest. «Els *Estats Units i el Canadà*, per una banda, i *Àustria i la República Txeca*, per una altra, van ser els primers a aplicar aquesta modalitat terapèutica més enllà de les seves fronteres» [6].

1.4 Nomenclatura

- **Cicle viable**: és un cicle que no té cap donant, pacient ni compatibilitat que prèviament hagi fallat, ni cap pacient prèviament trasplantat.
- **Cicles alternatius**: dos cicles són alternatius si, són de la mateixa mida i contenen exactament els mateixos pacients.
- **Cicles embedded**: un cicle A, és embedded a un cicle B, si la mida del cicle A és més petita que la del B, i el cicle B conté tots els pacients de l'A.
- **Estimació del nombre de trasplantament d'un cicle**: valor entre 0 i la mida del cicle, calculat en funció de la mida del cicle, la probabilitat de fallada de les donacions que el conformen i les probabilitats de fallada dels seus cicles alternatius i embeddeds. És una mesura de la robustesa del cicle.
- **Cicles compatibles**: si els conjunts formats pels seus pacients són disjunts.
- **Cicles incompatibles**: si no són compatibles.
- **Pool**: recopilació de donants, pacients, compatibilitats i el conjunt de tots els seus cicles possibles (fins a la mida contemplada).
- **Cicles inicials**: conjunt dels cicles possibles d'un pool.
- **Cicles seleccionables**: subconjunt dels cicles inicials que són viables.
- **Cicles seleccionats, selecció o solució**: subconjunt dels cicles seleccionables, compatibles entre ells.
- **Cicles no seleccionats**: la diferència entre el conjunt de cicles seleccionables i el conjunt de cicles seleccionats.
- **Solució no incrementable**: si per cada un dels cicles no seleccionats, n'hi ha almenys un de seleccionat incompatible amb ell.
- **Solució incrementable**: qualsevol solució que no és no incrementable.
- **Criteri**: és un valor inherent a les seleccions que quantifica numèricament la bondat d'aquesta en un determinat aspecte. Permeten realitzar judicis de valor de les seleccions, és a dir, determinar si una selecció és millor que una altre. N'hi ha infinitat, però els que es contemplen en aquest projecte són:
 - «**Trasplantaments**»: nombre de pacients que conté.
 - «**Pacients difícils**»: nombre de pacients difícils que conté.

- «**Selfcompatibles**»: nombre de parelles compatibles que conté.
 - «**Cicles de 2**»: nombre de cicles de dos trasplantaments que conté.
 - «**Score**»: Suma de les puntuacions de les donacions que conté.
 - «**Embeddeds**»: nombre de cicles seleccionables que són embedded a algun dels cicles d'aquesta.
 - «**Alternatius**»: nombre de cicles seleccionable que són alternatius a algun dels cicles d'aquesta.
 - «**Dispersió d'embeddeds**»: nombre de cicles d'aquesta que tenen almenys un cicle embedded entre els seleccionables.
 - «**Dispersió d'alternatius**»: nombre de cicles d'aquesta que tenen almenys un cicle alternatiu entre els seleccionables.
 - «**Trasplantaments esperats**»: suma de l'estimació del nombre de trasplantaments de cada cicle.
 - «**Pacients preseleccionats**»: suma dels nombre de pacients que formaven part de cicles fallats en la selecció prèvia a aquesta, i que conté aquesta. Només s'utilitzen en replanificacions sobre fallades.
 - «**Reparacions internes**»: suma del nombre de cicles d'aquesta que són alternatius o embeddeds a algun cicle fallat de la selecció prèvia a aquesta. Només s'utilitzen en replanificacions sobre fallades.
- **Optimització**: conjunt de seleccions que, partint del mateix conjunt de cicles seleccionables, permeten obtenir els valors més elevats d'un conjunt de criteris en ordre lexicogràfic.
 - **Reparació**: planificació a posteriori, d'un o més cicles viables, en lloc d'alguns planificats i fallats prèviament. En essència és una optimització.
 - **Reparació intracicle o interna**: si quan el cicle B, proposat a reparar el cicle fallat A, no té fallades i A i B són alternatius o el cicle B és embedded a A.
 - **Reparació intercycle o externa**: si no és una reparació intracicle
 - **Optimitzador**: component software que partint d'un conjunt de cicles seleccionables, s'encarrega de buscar seleccions òptimes, processar fallades i buscar reparacions òptimes.
 - **SAT**: «In computer science, the **Boolean satisfiability problem** (sometimes called **propositional satisfiability problem** and abbreviated as **SATISFIABILITY** or **SAT**) is the problem of determining if

there exists an interpretation that satisfies a given Boolean formula. In other words, it asks whether the variables of a given Boolean formula can be consistently replaced by the values TRUE or FALSE in such a way that the formula evaluates to TRUE.» [7]

- **MAX-SAT:** «In computational complexity theory, the **maximum satisfiability problem (MAX-SAT)** is the problem of determining the maximum number of clauses, of a given Boolean formula in conjunctive normal form, that can be made true by an assignment of truth values to the variables of the formula. It is a generalization of the Boolean satisfiability problem, which asks whether there exists a truth assignment that makes all clauses true.» [8]

1.5 Motivacions

Per als equips mèdics que realitzen la planificació dels intercanvis és útil disposar d'una aplicació informàtica que els permeti fer planificacions d'intercanvis i reparacions sota diferents configuracions de criteris per tal de comparar entre les solucions obtingudes amb uns i altres criteris i així poder decidir amb més varietat d'opcions.

D'altra banda, donat que aquest és un camp en constant investigació, una aplicació d'aquestes característiques pot ser també útil per als investigadors per tal de poder realitzar simulacions sobre diferents *pools* de dades de pacients i donants.

A partir d'aquestes motivacions, he procurat desenvolupar un programa altament configurable però sense perdre de vista els requisits establerts per l'ONT.

1.6 Propòsit

El propòsit d'aquest projecte és el de desenvolupar un aplicació adreçada al personal mèdic però també als investigadors d'aquest camp; aquesta aplicació està pensada com a eina de suport i seguiment de la planificació i implementació dels intercanvis, però també pot servir als investigadors per fer simulacions de procediments d'intercanvi de ronyó amb donant viu a partir de dades reals o bé sintètiques.

1.7 Objectius

Els principals objectius d'aquest projecte són:

- Creació d'un programa informàtic d'ajuda a la presa de decisions al personal mèdic que gestiona els intercanvis de ronyó; aquest programa els ha de permetre comparar possibles planificacions, triar-ne una, gestionar les fallades de darrera hora i replanificar els intercanvis a partir de les fallades sofertes, i els ha d'oferir informació detallada dels trasplantaments realitzats efectivament, de les fallades experimentades, etc.

- Explorar i comparar el comportament de diferents metodologies per a la resolució de problemes combinatoris, per a resoldre el problema de selecció dels cicles de trasplantaments a realitzar. Les tecnologies considerades són Programació amb restriccions (per exemple mitjançant MiniZinc i algun dels seus resoladors) i MaxSAT (per exemple el solver Open-WBO).
- Revisar i incorporar al programa la possibilitat d'ús de nous criteris (no usat en KEP's), entre ells l'estimació probabilística del nombre de trasplantaments que es realitzaran, així com, la distribució entre cicles de les possibles reparacions (fins ara s'utilitza el nombre de reparacions totals sense tenir en compte la distribució d'aquestes).

1.8 Punt de partida i abast del projecte

Per la realització d'aquest projecte he partit d'una bona base de coneixement en aquest àmbit, ja que, el curs passat (2016/17), vaig realitzar l'estada en l'entorn laboral dins d'aquest mateix entorn de recerca, tutoritzat pel Dr. Francesc Castro Villegas. Els aspectes treballats que m'han servit de punt de partida per aquest projecte són:

- Coneixement bàsic del context mèdic: tipologies de trasplantaments, funcionament del KEP espanyol i els seus criteris, fallades de darrera hora i els seus tipus i possibles mecanismes de reparació.
- Un criteri nou (no usat en KEP's), resultat final de l'EEL, que permet fer una estimació del *nombre esperat* de trasplantaments dins un cicle de intercanvi determinat (un valor real entre 0 i la mida del cicle) a partir de la probabilitat de fallada de les donacions que el formen i de les possibles reparacions (adjunto la formulació i el procés de raonament seguit per obtenir-lo a l'annex).

Aquest projecte no abasta:

- El càlcul de les compatibilitats entre les parelles ni la puntuació d'aquestes. És un preprocés de les dades que queda fora del programa.
- El càlcul de les probabilitats de fallada d'aquestes compatibilitats. És un preprocés de les dades que queda fora del programa. Un possible camí d'investigació, no contemplat en aquest projecte, és com determinar la probabilitat de fallada de les donacions segons criteris mèdics.
- Cicles de mida més gran a 3. Aquests cicles de mida 4 o superior no es fan servir, ara per ara, a la majoria de KEP's europeus. Seria molt fàcil incorporar-los al programa, afegint el càlcul d'aquests quan es calculen tots els possibles cicles, no afecta el funcionament de les altres parts del programa. El cost computacional augmentaria molt poc, ja que es pot incorporar dins del càlcul dels altres cicles.

- Cadenes de trasplantaments a partir d'un donant altruista. Seria molt fàcil incorporar-les al programa, afegint el càlcul d'aquestes quan es calculen tots els possibles cicles, no afecta el funcionament de les altres parts del programa. S'hauria d'estudiar més en detall, però molt probablement el cost computacional augmentaria poc.

2. Estudi de viabilitat

La viabilitat d'aquest projecte depenia de tenir accés als següents recursos gratuïtament:

- **Descripció del *KEP* espanyol i d'altres *KEP's* europeus** per conèixer els processos i criteris que s'hi utilitzen. Els he obtingut d'informes de l'ONT [9] i d'articles i informes d'altres organismes semblants i investigadors de diferents països europeus com el Regne Unit [10], Portugal [11] i Suècia [12], que em facilitaren els tutors d'aquest projecte.
- **Dades reals** per conèixer el volum de dades a tractar normalment i poder adequar i testejar bé la solució proposada i la interfície d'usuari a aquests volums. Facilitades pel *Departament d'Informàtica, Matemàtica Aplicada i Estadística* de la *Universitat de Girona*, més concretament per un dels tutors d'aquest projecte, gràcies a la col·laboració amb l'ONT. En consta un exemple a l'annex d'aquesta memòria.
- **Un ordinador amb editor de text i terminal** per desenvolupar el programa. Tenia accés al meu ordinador personal i a un ordinador del laboratori *Alan Turing* de l'edifici *Politécnica. IV* de la *Universitat de Girona*, tots dos amb sistemes operatius *Linux*.
- **Recursos humans** per tal de dissenyar, implementar i testejar el programa: correspon a la meua feina en aquest projecte, amb la supervisió dels meus tutors.

Com que es tenia accés a tots aquests recursos (tecnològics, d'informació, de dades i de recursos humans) necessaris per al seu desenvolupament del projecte de forma gratuïta, aquest era viable.

3. Metodologia

La metodologia triada i utilitzada per al desenvolupament d'aquest projecte ha sigut la metodologia *Scrum* (melé) [13]. Aquesta metodologia està basada en *sprints* (esprints o iteracions) setmanals, que serveixen a la vegada per a la retrospectiva (objectius complerts i no complerts, problemes trobats, possibles solucions i solució a adoptar) de l'anterior esprint, la posada en comú i debat de noves funcionalitats opcionals i la planificació del següent esprint (nous objectius i les seves prioritats)

Considerem que aquesta metodologia és de les més adequades a les característiques d'aquest projecte, ja que es tracta d'una metodologia àgil, vàlida per a equips de desenvolupament petits i que es basa en els principis d'inspecció contínua, adaptació, priorització, autogestió i innovació. L'objectiu final d'aquesta és proveir als projectes d'uns processos iteratius que facilitin una bona flexibilitat en els requisits, una alta productivitat i que un alt grau de qualitat del producte final.

Les característiques d'aquesta metodologia que he trobat més interessants són:

- Desenvolupament iteratiu i incremental.
- Replantejament i replanificació d'objectius a l'inici de cada iteració.
- Flexibilitat i adaptació dels requisits i funcionalitats del programa i la seva priorització per valor.
- Comunicació molt freqüent entre l'equip de desenvolupament i el *Product owner* (client).
- Autogestió de l'equip de desenvolupament.
- Prova i ús de mòduls del programa en paral·lel a la implementació d'altres.

Les característiques del projecte que han motivat l'elecció d'aquesta són:

- Projecte i àmbit complex.
- Funcionalitats opcionals poc definides i canviants (algunes d'elles aparegudes durant el desenvolupament del projecte).
- Data de finalització no fixada (convocatòria de juny o setembre).
- Possibilitat de reunions setmanals de coordinació amb el client (exercien com a tal els tutors).

- Interès d'incorporar aspectes i funcionalitats innovadores, sense sacrificar les funcionalitats opcionals de més valor per a l'usuari final.

4. Planificació

La data prevista d'inici del projecte era 19 de març de 2018 i la data prevista de finalització 27 de maig de 2018, un total de 10 setmanes. La duració de cada sprint és d'una setmana. Per tant un total de 10 sprints.

- 1a. Fase: anàlisi de requeriments i sistema de classes (1 *sprint*)
 - Anàlisi de requeriments.
 - Disseny del sistema de classes del programa.
 - Disseny, implementació i test de les classes representatives dels elements bàsics, donants, pacients, etc.
 - Elaboració de la documentació pertinent.
- 2a. Fase: recopilacions de classes base i càlcul de cicles (1 *sprints*)
 - Disseny, implementació i test de les recopilacions de donants, pacients i les seves compatibilitats.
 - Familiarització amb la llibreria `scala.util.parsing.json.JSON`.
 - Implementació i test de la lectura de l'input del programa, fitxers en format *json*. Adjunt un exemple a l'annex.
 - Disseny, implementació i test de l'algorisme de càlcul de cicles.
 - Disseny, implementació i test de la sortida de dades.
 - Elaboració de la documentació pertinent.
- 3a. Fase: comparació de *solvers* en aquest àmbit (2 *sprints*)
 - Disseny, implementació i test (amb dades creades ad-hoc), del programa per *Minizinc* (input, model i restriccions).
 - Disseny, implementació i test de l'algorisme que a partir d'un conjunt de cicles, crei una instància a resoldre pel programa creat al punt anterior, i generació d'instàncies amb dades reals amb ell.
 - Execucions, i recopilació d'informació d'elles, del programa creat per a *Minizinc*, amb les instàncies creades al punt anterior.
 - Familiarització amb el funcionament dels *solvers* *MaxSat* i proves amb instàncies creades ad-hoc amb el *solver* *Open-WBO*.

- Disseny, implementació i test de l'algorisme que partir d'un conjunt de cicles, crei una instància a resoldre per el *solver Open-WBO*, i generació d'instàncies amb dades reals amb ell.
 - Execucions, i recopilació d'informació d'elles, del *solver Open-WBO*, amb les instàncies creades al punt anterior.
 - Comparació i decisió del que millors resultats doni.
 - Elaboració de la documentació pertinent.
- 4a. Fase: generació de planificacions i interfície d'usuari (2 *sprints*)
- Disseny, implementació i test dels algorismes per crear instàncies i interactuar amb el *solver* (fer la crida i recollir el resultat).
 - Implementació i test del càlcul dels valors dels cicles i de les seleccions segons els criteris.
 - Disseny, implementació i test de la sortida de dades amb informació de la planificació i els seus cicles.
 - Familiarització amb les llibreries *javax.swing* i *java.awt.event*.
 - Disseny, implementació i test de la interfície d'usuari, en endavant GUI, per carregar dades, calcular cicles i, crear (entrant els criteris l'usuari) i veure planificacions.
 - Lligar la interfície d'usuari amb el codi que pertoqui i testejar el funcionament.
 - Elaboració i ampliació de la documentació pertinent.
- 5a. Fase: fallades i interfície d'usuari (1 *sprints*)
- Disseny, implementació i test del processament de les fallades.
 - Disseny, implementació i test de la sortida de dades de les fallades.
 - Disseny, implementació i test de la part de la GUI per tractar fallades (les entra l'usuari) i veure els seus efectes.
 - Lligar la part nova de la interfície d'usuari amb el codi que pertoqui i testejar el funcionament.
 - Elaboració i ampliació de la documentació pertinent.

➤ 6a. Fase: reparacions (1 *sprint*)

- Implementació i test del càlcul dels valors dels cicles i de les seleccions segons els dos criteris addicionals, en reparacions.
- Disseny i implementació i test de les parts de la GUI per fer replanificacions (entrant els criteris l'usuari) i veure-les.
- Lligar la part nova de la interfície d'usuari amb el codi que pertoqui i testejar el funcionament.
- Elaboració i ampliació de la documentació pertinent.

➤ 7a. Fase: *save & load* (1 *sprint*)

- Implementació i test dels guardats de dades, informatives o per al programa.
- Implementació i test de la lectura de dades a partir de fitxers carregats pel programa.
- Elaboració i ampliació de la documentació pertinent.

En total 9 *sprints*, perquè hi havia una setmana de festa pel mig.

Aquesta planificació, reflectia els assoliments dels requeriments obligatoris, deixant marge a, la possibilitat de posposar la data de finalització a finals d'agost de 2018, per acabar requeriments obligatoris (si sortien imprevistos) i/ o per assolir més funcionalitats opcionals, tal com ha acabat passant. Veure capítol 11 per veure les desviacions que hi ha hagut d'aquesta planificació.

5. Requisits del sistema

5.1 Requisits

Ha de permetre a l'usuari:

- **Crear** pool's a partir del fitxer que triï i calcular-ne i veure tots els possibles cicles i la seva informació.
- **Fer optimitzacions** triant-ne els criteris i el seu ordre, i veure'n les solucions resultants en un temps raonable.
- **Triar** els criteris a maximitzar (i el seu ordre) en les optimitzacions. Els criteris seleccionables han de ser els següents:
 - Suma del nombre de **trasplantaments** dels cicles seleccionats.
 - Suma del nombre de **parelles self-compatible** dels cicles seleccionats.
 - Suma del nombre de **pacients difícils** dels cicles seleccionats.
 - Nombre de **cicles de 2** seleccionats.
 - Suma de les **puntuacions** (score) dels cicles seleccionats.
 - Suma del nombre de **cicles embedded** que tenen els cicles seleccionats entre la resta de seleccionables.
 - Suma del nombre de **cicles alternatius** que tenen els cicles seleccionats entre la resta de seleccionables.
- **Entrar les fallades** de darrera hora (donants o pacients no disponibles i/ o proves de compatibilitat positives) ocorregudes a una selecció i veure'n els efectes.
- **Calcular reparacions** a partir de les fallades indicades, triant-ne de nou els criteris de selecció el seu ordre, i veure les solucions resultants en un temps raonable.
- **Triar** els criteris a maximitzar en les reparacions d'entre els següents (a més dels dels criteris d'optimització indicats prèviament):
 - Nombre de pacients **preseleccionats**
 - Nombre de reparacions **internes**
- **Guardar qualsevol estat** del procés (pool, selecció, fallades i reparacions) en fitxers en format *json*.

- **Carregar** els fitxers format Json del punt anterior i seguir treballant-hi.
- **Guardar la informació** de qualsevol estat del procés (pool, selecció, fallades i reparacions) en format *txt* de forma llegible per a l'usuari.

5.2 Funcionalitats opcionals:

Permetre a l'usuari:

- **Treballar a la vegada amb diferents** pool's, optimitzacions, seleccions, fallades i reparacions de forma ordenada i clara.
- **Comparar** de manera fàcil, ordenada i clara els resultats de diferents optimitzacions, seleccions, fallades o reparacions fetes sobre el mateix punt de partida.
- **Obtenir tota la informació** de tots els pacients, donants, compatibilitats i cicles dels diferents pool's creats i carregats.
- **Obtenir informació de tots els passos fets** fins al moment de qualsevol selecció, fallades i reparacions.
- **Visualitzar** el graf de compatibilitats, el subgraf selecció i/o les fallades ocorregudes, de forma gràfica i a poder ser interactiva.
- **Triar** els criteris a maximitzar en les optimitzacions i reparacions d'entre els següents (a més dels obligatoris):
 - Nombre de cicles seleccionats que tenen **almenys un cicle embedded** entre la resta de seleccionables.
 - Nombre de cicles seleccionats que tenen **almenys un cicle alternatiu** entre la resta de seleccionables.
- **Fer** que l'aplicació sigui executable en diferents entorns.
- **Facilitar** la instal·lació i l'ús de l'aplicació.

6. Estudis i decisions

6.1 Llenguatge

El llenguatge de programació usat és *Scala*, segons ells mateixos «Scala combines object-oriented and functional programming in one concise, high-level language. Scala's static types help avoid bugs in complex applications, and its JVM and JavaScript runtimes let you build high-performance systems with easy access to huge ecosystems of libraries.» [14], que ja coneixia i he triat pels següents avantatges:

- Els seus compilats (byteCode.class), són executables amb la màquina virtual de *Java*, força extesa i de fàcil instal·lació.
- Permet l'ús de llibreries *Java*.
- És multi-paradigma, combina programació iterativa, orientada a objectes i funcional.

6.2 Llibreries

- ***java.io***: per totes les funcionalitats que llegeixen o guarden fitxers.
- ***javax.swing***: per als components que formen la GUI.
- ***java.awt.event***: per les funcionalitats de la GUI.
- ***scala.collection.mutable.HashMap***: per l'ús d'estructures de dades tipus HashMap.
- ***collection.JavaConverters***: per convertir llistes *Java* a llistes *Scala*.
- ***scala.util.parsing.json.JSON***: per al parsejat de fitxers *Json*.
- ***scala.util.Try***: per a la captura i tractaments d'excepcions.

6.3 Compilador i entorn d'execució

Venen determinats pel llenguatge utilitzat. Compilador d'*Scala* i l'*OpenJDK* de *Java*, que proporciona el compilador de *Java* i l'entorn d'execució (Màquina virtual de *Java*).

6.4 Explicació dels solvers

He provat dos tipus diferents, que són:

- *Minizinc*:

Tal com es defineixen ells mateixos « Minizinc is a **free and open source constraint modeling language**» used to «model constraint satisfaction and

optimization problems in a **high-level, solver-independent** way, taking advantage of a large library of pre-defined constraints.» and «Your model is then compiled into FlatZinc, a solver input language that is understood by a wide range of solvers.» [15]

Per tant, no és un *solver* en si, és una capa de software a alt nivell que permet descriure en un format concret d'alt nivell, la teva modelització d'un problema, i compilar-la on la sortida està en un format de modelització compatible amb molts *solvers* externs. La distribució conté, a més de les llibreries i el compilador, un IDE, que permet escriure, compilar i executar amb diferents *solvers* i opcions els teus models. Aquests es poden dotar d'entrada de dades des d'un fitxer extern, per tal de que un programa amb un model prou genèric pugui treballar amb diferents dades i volums de dades. Vaig provar diferents *solvers* que permet usar *Minizinc*. Es pot descarregar des del seu web, i no requereix cap instal·lació. [15]

- Open-WBO:

És un *MaxSAT solver* (veure capítol anterior), premiat en diverses ocasions, del qual distribueixen «An open source version of the MaxSAT solver WBO». [16]

El seu input són instàncies en format *.dimacs*, on s'especifica el nombre de variables i les clàusules lògiques de les restriccions entre les variables del problema amb el seu pes. Es pot descarregar des del seu web [16], i requereix una precompilació.

6.5 Comparació de *solvers*

Després de provar tots els solvers que permet usar MiniZinc, el que millors resultats va donar va ser el OSICBC 2.9/1.16 que és un solver MIP.

Abans de mostrar i analitzar la comparativa hi ha dos aspectes molt rellevants a remarcar, sobre els resultats dels temps d'execució:

- I. Aquestes estan fetes amb una modelització concreta i unes dades concretes, de manera que implementant una modelització diferent a aquesta i/o canviant de dades, els resultats dels temps d'execució podrien variar.
- II. En les proves d'execució fetes amb un sol criteri, s'han provat tots i s'ha utilitzat per comparar el pitjor valor obtingut, és a dir, el criteri amb el que ha tardat més.

A continuació la taula resum de la comparativa dels dos solvers:

| Propietats | «Solver» | |
|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| | OSICBC 2.9/1.16 | Open-WBO |
| Preinstal·lació necessària | No | És necessària una pre-compilació que pot ser estàtica. |
| Possibilitat d'execució des de programes externs | Si | Si |
| Possibilitat d'optimització amb pesos no enters | Si | No |
| Possibilitat d'optimització de tots els criteris junts | Si | Si (amb coneixements de codificació en CNF de restriccions pseudobooleanes). |
| Temps d'execució amb 1 criteri i dades creades (poc volum) | 4 segons 484 mseg | 628 mseg |
| Temps d'execució amb varis criteris i dades creades (poc volum) | 333 mseg | No provat per falta de temps, però teòricament viable. |
| Temps d'execució amb 1 criteri i dades reals | 38 segons 555 mseg | 1 segon 319 mseg |
| Temps d'execució amb varis criteris i dades reals | Inviabile (la conjunció de varis criteris porta, en la majoria de casos, a un overflow de la variable que conté el valor òptim) | No provat per falta de temps, però teòricament viable. |

Alguns dels resultats van ser realment sorprenents, en destacaria que amb el solver OSICBC 2.9/1.16, s'obtenen millors temps d'execució amb varis criteris que amb un sol criteri, això si amb dades fictícies i poc volum d'elles. Això és degut a que, en aquest cas, amb més criteris el problema queda més restringit i per tant, el solver tarda menys a trobar una solució.

Veient les propietats que m'ofereixen els diferents *solvers*, amb el model implementat, he triat l'Open-WBO, ja que és el *solver* que millor s'adequa a les necessitats del programa. Deixant de banda el handicap que suposa el fet que no permet treballar amb pesos no enters (fàcilment solucionable com es detalla al capítol 8), té l'avantatge de poder treballar amb varis criteris a la vegada i a més dóna uns temps de resposta bastant millors que el solver OSICBC 2.9/1.16, sobretot hi ha molta diferència treballant amb dades reals, que és el que més ens interessa, ja que serà amb el que treballarà realment el programa.

Per tant, el solver elegit, ja que s'adapta millor a les necessitats del programa a desenvolupar és l'Open-WBO, i aquest és el que serà utilitzat en el programa.

7. Anàlisi i disseny del sistema

7.1 Anàlisi

Aquest programa no ha de comptar amb una base de dades d'on obtenir la informació, per tant, tota la informació a tractar per ell és entrada en temps d'execució. També ve fortament marcat, en la fase inicial, per el format de les dades d'entrada. Aquest aspecte condiona la creació i organització dels seus components bàsics (donants, pacients i compatibilitats)

7.1.1 Model de dades

- Donants, pacients i compatibilitats: fortament determinats per el format de l'input, té tres factors clau.
 - I. No hi ha cap llistat específic de donants, venen donats a les compatibilitats.
 - II. Conté un únic llistat amb tots els pacients, on per cada pacient s'especifica un llistat amb totes les seves compatibilitats.
 - III. Els donants i el pacients tenen un identificador, que és un número enter.

Per tant aquest format de lectura és problemàtic de cara a establir lligams (referències) entre els pacients i els seus donants vinculats.

- Cicles: calculats en el propi programa, no té problemàtica a l'hora d'establir lligams amb les donacions (compatibilitats) que el formen, ja que el seu càlcul es fa a posteriori de la creació de les compatibilitats. Cal tenir en compte però, que són de mida variable i en un futur es podria incrementar la mida màxima de cicle contemplada.
- Pool's: no és necessari poder accedir a elements concrets ràpidament, com a molt es faran iteracions sobre la recopilació, per tant no precisen de cap estructura de dades ni solució específica.
- Optimitzadors: han d'estar ben estructurats per tenir classificat cada cicle, en un determinat moment, segons la funcionalitat que li proporciona (seleccionable, seleccionat, descartat, fallat o fet), així com obtenir la informació dels passos previs a l'actual.
- Gestió de diferents pools i optimitzacions: ha de permetre treballar a la vegada amb diferents pool's i optimitzacions, de manera que han d'estar identificats i ben recopilats en alguna estructura que permeti un ràpid accés a qualsevol element concret.

- Interfície d'usuari: ha de permetre triar els criteris d'optimització, les fallades. Així com guardar diferents tipus d'arxius, segons contingut i format del contingut.

7.1.2 Model de processos

- Càlcul dels possibles cicles d'un pool: és un procés simple i de poc cost en temps i memòria (no és crític), però s'ha intentat ser el màxim d'eficients.
- Estats dels cicles: l'estat de cada cicle no és inherent a ell mateix, si no als processos de selecció, fallades i reparacions que hagi sofert el seu entorn. Per tant, el cicle no canvia d'estat, el que canvia és com el classifica l'entorn on es troba en un determinat moment.

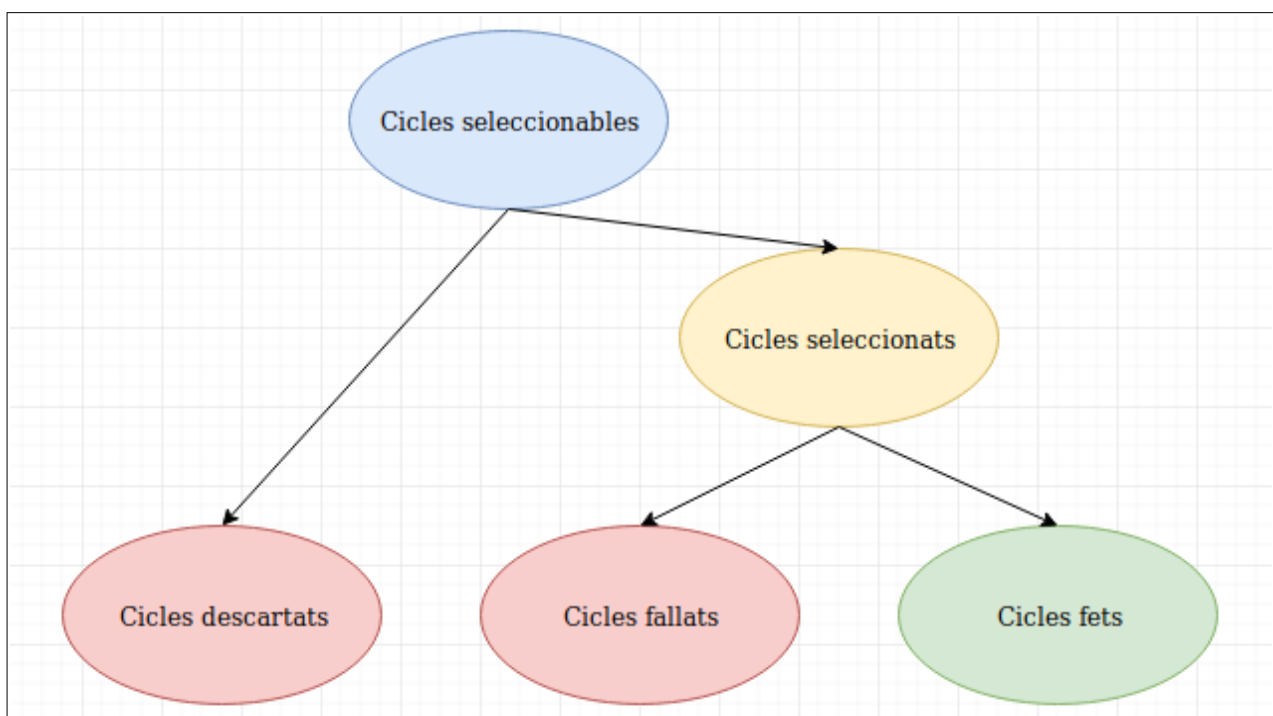


Fig. 9. Diagrama d'estats dels cicles.

Inicialment són **seleccionables**. De **seleccionables** poden passar **a seleccionats** per un procés d'**optimització** (el *solver* l'ha triat) **o a descartats** per un procés d'**entrada de fallades** (conté alguna fallada o algun pacient ja trasplantat). De **seleccionats** poden passar **a fallats** o **fets** per un procés de d'entrada de fallades (segons si contenen fallades o no)

- Estats dels optimitzadors: tenen diferents estats que conformen el seu cicle de vida dins de l'aplicació.

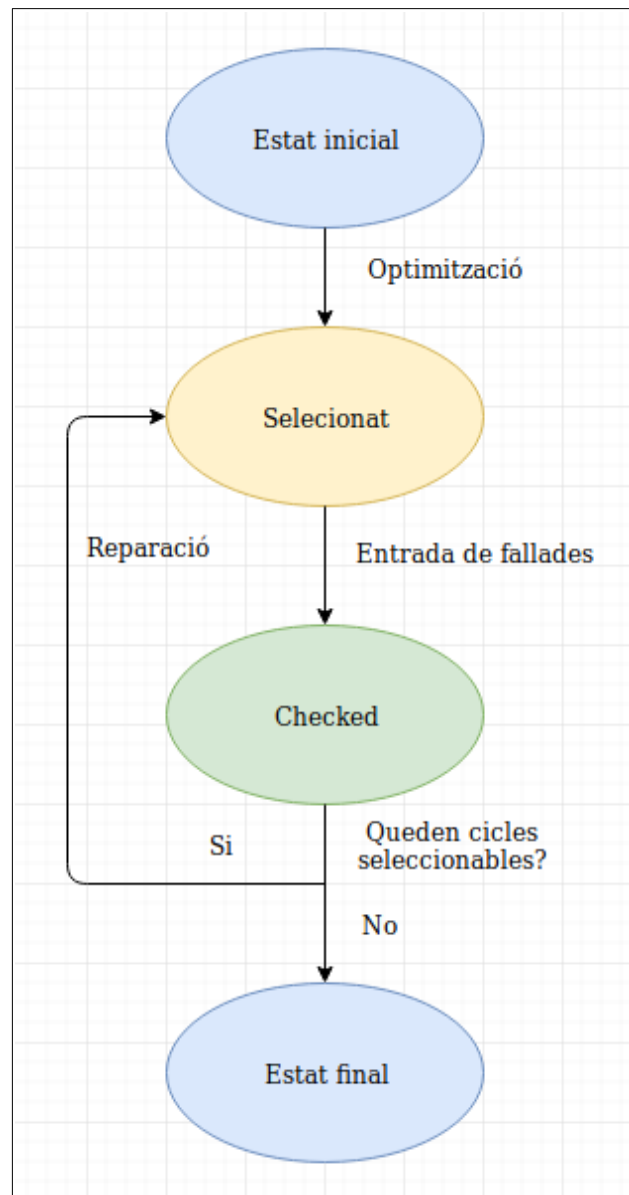


Fig. 10. Diagrama d'estats dels optimitzadors.

Els estats inicial i final, són estats especials. En el primer tots els cicles possibles del pool són seleccionables. En el segon no hi ha cicles seleccionables.

- Optimització i reparació: és un dels operadors de canvi d'estat. S'aplica sobre optimitzadors en estat inicial o checked. Donat que s'ha de poder seguir treballant amb aquest, no pot ser modificat.
- Entrada de fallades: és un dels operadors de canvi d'estat. S'aplica sobre optimitzadors en estat seleccionat i aquest mou els cicles del conjunt de seleccionats al conjunt de fallats o al de fets (segons si conté o no fallades) i del conjunt de cicles seleccionables al de descartats si pertoca. Donat que s'ha de poder seguir treballant amb aquest, no pot ser modificat.

7.2 Disseny

7.2.1 Model físic de dades

- Donants, pacients i compatibilitats: la problemàtica per establir els lligams entre ells, comentada en l'apartat 8.1, s'ha solucionat fent els lligams utilitzant l'identificador, en comptes de l'objecte.
- Cicles: consta d'una llista de mida variable de compatibilitats (donacions) la mida de la qual determina la mida del cicle.
- Optimitzadors: consten de cinc llistes de cicles, els seleccionables, els seleccionats, els descartats, els fallats i els fets. Les dues primeres són purament llistes de cicles, mentre que les tres últimes són llistes de llistes de cicles, per tenir-los organitzats per iteració, per tal de poder obtenir la informació dels passos previs de forma organitzada, o sigui pas a pas.
- Gestió de diferents pools i optimitzacions: per poder treballar amb diferents pool's i optimitzadors, s'ha pres la decisió de que l'usuari nombri cadascun dels passos que realitza. D'aquesta manera es pot identificar els diferents pool's i optimitzadors carregats en memòria. Aquests es guarden en estructures Hash, on la clau és el seu nom, per tal de tenir-hi un ràpid accés, sota demanda de l'usuari. Més específicament, hi ha una estructura Hash general que conté associat a cada clau, el pool i una altra estructura Hash amb tots els seus optimitzadors. D'aquesta manera, a més a més de tenir un ràpid accés als elements concrets, es poden repetir noms d'optimitzadors en diferents pools, per tal de que no es faci tant carregós per a l'usuari la tria de noms d'aquests.
- Interfície d'usuari: la tria de criteris consta de dues llistes per poder triar-los. Una inicialment conté tots els criteris, i una altra inicialment no en conté cap. L'usuari passa els criteris que vol utilitzar a la segona llista, en l'ordre que vol.

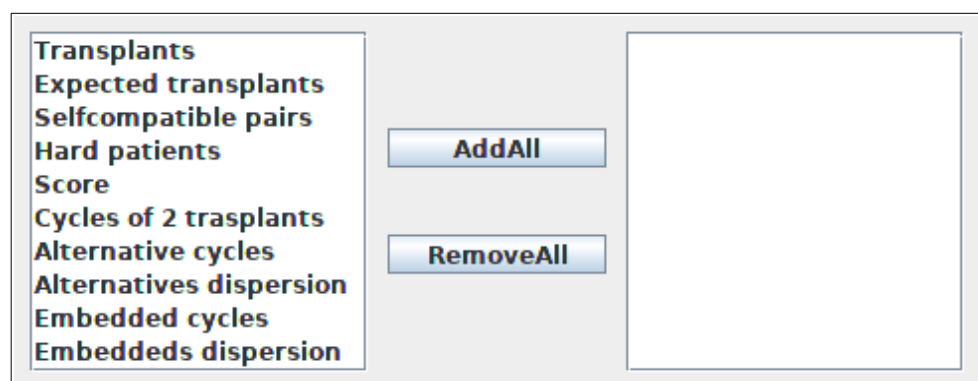


Fig. 11. Llistes que permeten triar a l'usuari els criteris i el seu ordre per fer optimitzacions.

Fig. 12. Llistes que permeten triar a l'usuari els criteris i el seu ordre per fer reparacions.

La tria de fallades consta de quatre llistes per poder triar les fallades ocorregudes. Les tres primeres contenen inicialment i per separat, tots els element dels cicles seleccionats subjectes a possibles fallades (pacients, donants i compatibilitats), la quarta està inicialment buida. L'usuari passa els elements fallats a l'última llista.

| Recipients | Donors | Compatibilitys | Failures |
|------------|--------|----------------|-------------|
| 1 | 2 | 1 -> 2 | Donor 1 |
| 2 | 3 | 3 -> 1 | 2 -> 3 |
| | | | Recipient 3 |
| | | | |

Fig.13. Llistes que permeten triar a l'usuari les fallades.

La tria dels diferents tipus d'arxius a guardar es fa mitjançant desplegable on es selecciona el tipus a guardar.

Fig.14. Desplegable per triar el tipus d'arxiu a guardar (contingut i format)

7.2.2 Model físic de processos

Els detallo per fases de desenvolupament, com a la planificació, així reflecteixen la metodologia usada són més fàcils d'explicar i entendre i es veu clarament que a cada fase es desenvolupava un subprograma, que s'encarrega d'un processos molt específics.

- Fase 1 (sistema de classes): pertany tot al model físic de dades de donants, pacients, compatibilitats i cicles i no desenvolupa processos físics.
- Fase 2 (pools): aquesta utilitza els models de dades físic desenvolupats en la primera fase i desenvolupa els següents processos:
 - I. Lectura de dades del pool, fàcilment realitzat utilitzant llibreries externes de parsejat de format json.
 - II. Càlcul de tots els cicles possibles del pool.
- Fase 3 (comparació de solvers): no desenvolupa models físics de processos.
- Fase 4 (planificacions): fortament lligada amb la funcionalitat de la interfície d'usuari, desenvolupa els següents processos:
 - I. Demanda de criteris a l'usuari, com es pot veure a l'apartat anterior 8.2.1. Aquests venen donats en un llistat de paraules clau, que representen els diferents criteris, ordenats amb l'ordre que els ha donat l'usuari.
 - II. Obtenció de totes les solucions que maximitzen els criteris en ordre lexicogràfic. Inclou dos subprocessos:
 - 1) Cerca de totes les solucions que maximitzen el primer criteri. Això inclou quatre subprocessos:
 - i. Creació de la instància a resoldre pel solver, a partir de les restriccions de les incompatibilitats entre cicles, i el valor de cada cicle amb el primer dels criteris.
 - ii. Crida al solver a resoldre la instància creada al punt anterior i recull i tractament de la sortida de la crida. Si hi ha solució, guardat d'aquesta en un llistat de solucions trobades (és la única de moment) i guardat del seu valor òptim d'aquest primer criteri.
 - iii. Adició a la instància creada d'una restricció més a complir, l'última solució trobada negada, d'aquesta manera diem al solver que no volem aquella solució concreta.
 - iv. Crida al solver a resoldre la instància i recull i tractament de la sortida. Si hi ha solució, comparació del valor òptim en el primer criteri d'aquesta amb el valor òptim guardat al punt ii. Si aquest és igual, s'afegeix la nova solució al llistat de solucions trobades i es torna al punt iii.

- 2) Per cadascun del criteris restants, filtratge de les solucions que tenen valor màxim en el criteri.
- III. Filtratge de la llista de cicles seleccionables dels cicles no seleccionats.
 - IV. Mostra a l'usuari, mitjançant la interfície gràfica, de les solucions obtingudes del procés II.
- Fase 5 (fallades): fortament lligada amb la funcionalitat de la interfície d'usuari, desenvolupa els següents processos:
 - I. Demanda de fallades a l'usuari, com es pot veure a l'apartat anterior 8.2.1. Aquestes venen donades en un llistat d'elements fallats (pacients, donants i compatibilitats).
 - II. Processament de les fallades. Això inclou dos subprocessos:
 - 1) Processament dels cicles seleccionables. Inclou quatre subprocessos:
 - i. Filtratge dels cicles seleccionats que no contenen cap de les fallades entrades i obtenció del llistat de pacients que hi participen.
 - ii. Filtratge dels cicles seleccionables dels cicles que no contenen cap dels pacients del llistat obtingut al punt anterior.
 - iii. Filtratge del llistat de cicles obtingut al punt anterior dels cicles que no contenen cap element fallat.
 - iv. La llista de seleccionables passa a ser la obtinguda al punt anterior.
 - 2) Processament dels cicles seleccionats. Inclou tres subprocessos:
 - i. Filtratge dels cicles seleccionats que contenen algun dels element que han fallat i adició d'aquesta llista resultant a la llista de llistes de cicles fallats.
 - ii. Filtratge dels cicles seleccionats que no contenen cap element fallat i adició d'aquesta llista resultant a la llista de llistes de cicles fets.
 - iii. La llista de seleccionats passa a ser la llista buida.
 - Fase 6 (replanificacions): ídem a la fase 4, en quant al model físic de processos.

- Fase 7 (save & load): fortament lligada amb la funcionalitat de la interfície d'usuari, desenvolupa els següents processos en desats:
 - I. Demanda de la tipologia de fitxer a guardar a l'usuari com es pot veure al punt 8.2.1. així com el nom i la carpeta de destí.
 - II. Creació del fitxer amb el nom i el destí especificats.
 - III. Escriptura del contingut del fitxer segons la tipologia triada i tancat d'aquest.
- I els següents processos en lectures:
 - I. Demanda a l'usuari de la ruta del fitxer.
 - II. Obertura i lectura del fitxer.
 - III. Creació de les dades físiques en memòria.

8. Implementació i proves

Detallo la implementació i les proves organitzades per paquets de software.

8.1 Model de classes

8.1.1 Package poolMembers

Conté totes les classes representatives dels elements que conformen un pool.

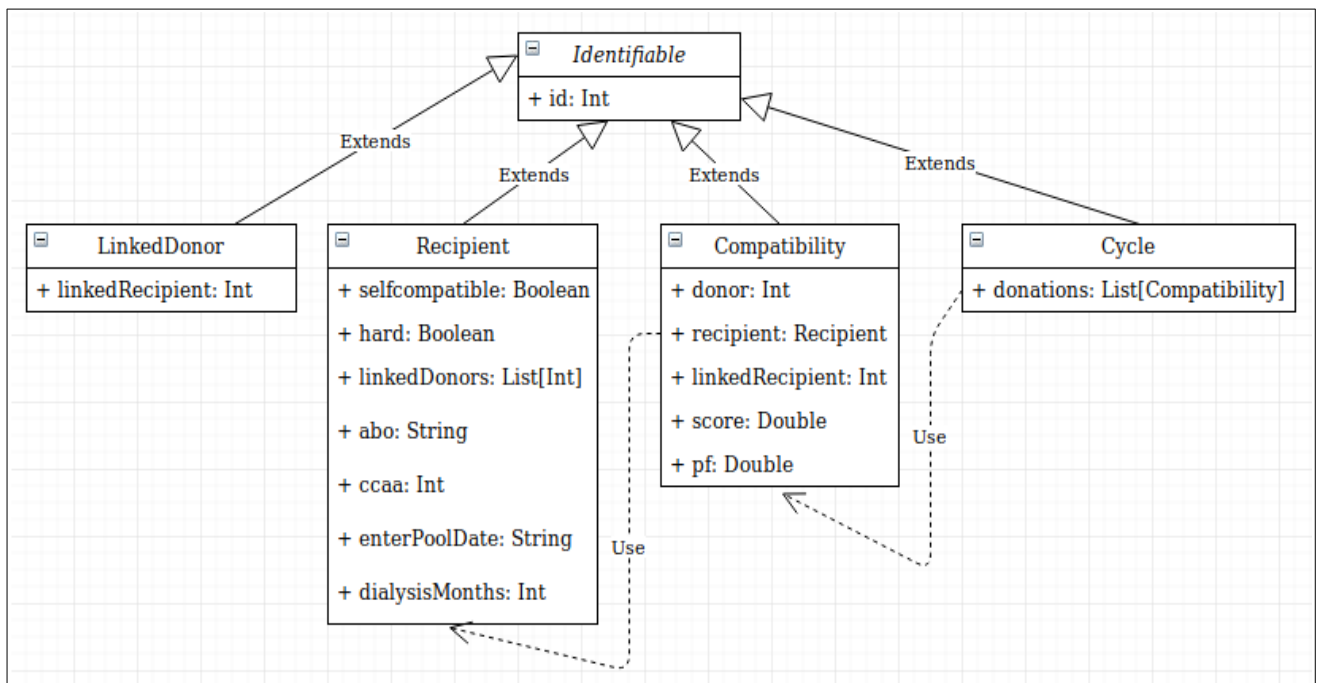


Fig. 15. Model de classes package poolMembers.

Implementa els models físics de dades relatius a la fase 1 de desenvolupament planificada, vistos al capítol anterior. S'ha creat una classe abstracte base, que simplement conté un identificador i la redefinició del mètode que en dona el seu codi hash, que és el seu mateix identificador. Totes les altres classes d'aquest paquet hereten d'aquesta classe, tot i que les compatibilitats i els cicles no tenen perquè tenir-lo. És coherent fer-ho així perquè no té sentit tenir cicles ni compatibilitats repetides en un pool i així es facilita molt la comparació d'igualtat entre ells, només s'ha de comparar l'identificador, i determina la comparació (`<` i `>`) que també es fa per identificador, útil per exemple en el càlcul de cicles.

8.1.2 Package pool

Conté la classe representativa de les recopilacions de donant, pacients, compatibilitats i per tant utilitza les classes del paquet del punt anterior.

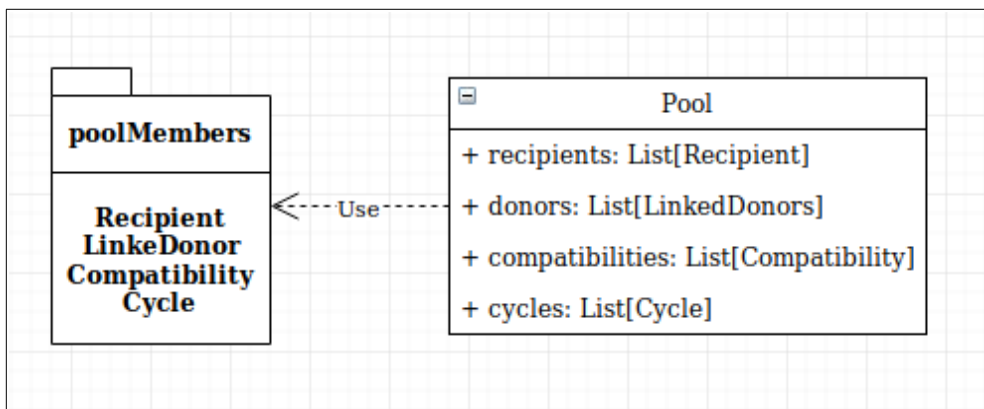


Fig. 16. Model de classes package Pool.

Implementa el càlcul de cicles, l'algorisme del qual es detalla al punt 9.1.2.

8.1.3 Package poolOptimizer

Proveeix totes les funcionalitats d'optimització i de processament de fallades.

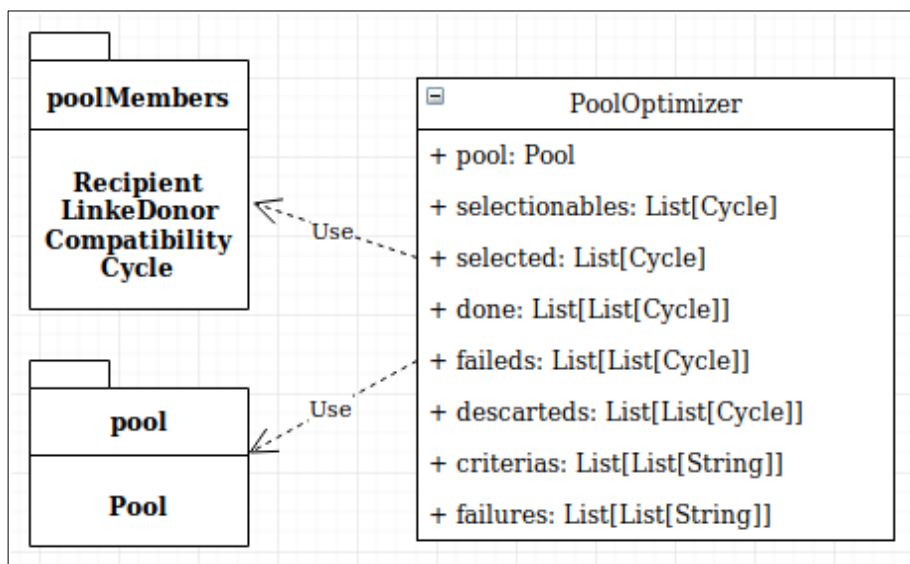


Fig. 17. Model de classes package poolOptimizer.

Implementa els models físics de processos corresponents a les fases de desenvolupament planificades 4,5 i 6 vistos al capítol anterior.

8.1.4 Interfície d'usuari (GUI) i classe principal executable (dss)

La classe gui implementa els models físics de dades i de processos relatius a la interfície d'usuari. Aquesta implementa diferents interfaces de *Java*:

- ActionListener: usada per detectar quan es clica un botó i actuar en conseqüència.
- ComponentListener: usada per detectar canvis en la mida de la finestra i obrar en conseqüència (reposicionar i canviar de mida components de la finestra).
- ItemListener: usada per detectar canvis d'element seleccionat en desplegable i actuar en conseqüència (generalment canviar la informació a mostrar).
- MouseListener: usada per detectar dobles clics fets amb el ratolí i obrar en conseqüència (passar elements d'una llista a una altra)
- MouseMotionListener: usada per detectar arrosseaments amb el ratolí i actuar en conseqüència (canviar l'ordre dels elements de la llista corresponent).

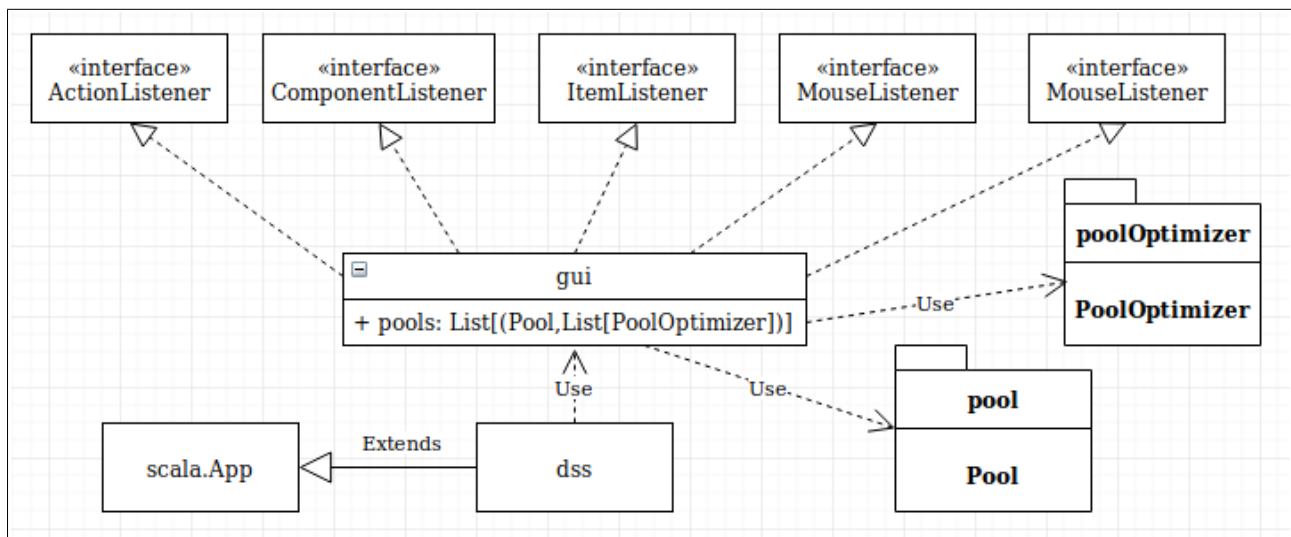


Fig. 18. Model de classes general.

L'objecte dss hereta d'un trait d'scala que permet que sigui executable, és a dir que sigui com el mètode main.

8.2 Algorisme de càlcul de cicles

Parteix d'un llista de parelles (pacient, llista de compatibilitats del pacient) ordenada per pacient creixentment. És el següent:

```
Per cada parella (p1,cp1) de la llista {  
  Per cada c1 de cp1 {  
    p2 = pacient vinculat al donant de c1  
    cp2 = compatibilitats del p2  
    Per cada c2 de cp2 {  
      p3 = pacient vinculat al donant de c2  
      Si (p1 < p2 i p1 == p3) hi ha cicle de dos format per c1 i c2  
      altrament si (p1 < p2) {  
        cp3 = compatibilitats del p3  
        per cada c3 de cp3 {  
          p4 = pacient vinculat al donant de c3  
          si (p1 < p3 i p2 != p3 i p1 == p4) {  
            Hi ha cicle de tres format per c1, c2 i c3  
          }  
        }  
      }  
    }  
  }  
}
```

En els cicles de dos es comprova que:

- el segon pacient sigui diferent del primer pacient per evitar contemplar les compatibilitats de les parelles compatibles. No es fa explícitament, bé implícit en la comparació $p1 < p2$.
- El segon pacient sigui més gran que el primer, per tal de no repetir cicles, per exemple (1→2→1) i (2→1→2).

En els de tres es comprova el mateix que en els de dos ($p1 < p2$), i també que:

- $p1 < p3$ per evitar cicles repetits (1→2→3→1) i (2→3→1→2) però no ($p2 < p3$) ja que llavors perdríem (1→3→2→1).
- $p2 \neq p3$ per evitar contemplar les compatibilitats de les parelles compatibles.

Té l'avantatge, d'avant d'altres algorismes de càlcul de cicles (per exemple fent combinacions de pacients), de generar menys combinacions inútils, ja que aquest es basa en les compatibilitats existents. Tot i així, aquest avantatge depèn de les dades concretes, per exemple un pool en que existeixin compatibilitats entre tots els pacients, aquesta avantatge no existeix (a més compatibilitats menys avantatge).

8.3 Problemes i solucions apareguts

La gran majoria de problemes han vingut donats per l'ús del solver. Detallo un llistat dels més rellevants:

- El solver usat no contempla pesos no enters: la solució usada és multiplicar els valors no enters per un cert valor de precisió (10 dècima, 100 centèsima, etc) i els transformo en enters per crear la instància a resoldre pel solver.
- El solver usat no proporciona totes les solucions: la solució usada és afegir a la instància, com a restriccions que s'han de complir, les solucions trobades fins al moment negades, i anar repetint les crides al solver, obtenint a cada pas una nova solució, fins que el valor òptim de la solució trobada decreixi o ja no se'n trobin més.
- Sense coneixements de codificació de restriccions pseudobooleanes no puc trobar solucions òptimes de més d'un criteri usant el solver: la solució usada és usar el solver per trobar totes les solucions que maximitzen el primer dels criteris, i anar-les guardant en una llista. Un cop les tinc totes, n'hi ha prou d'anar repetint aquest dos passos, tants cops com criteris restants hi hagi:
 - Recórrer totes les solucions buscant el valor màxim del criteri
 - Filtrar les que tinguin aquest valor màxim en aquest criteri

8.4 Proves

Totes les proves s'han realitzat utilitzant dades creades ad-hoc, ja que la quantita d'informació de les dades reals, fa molt difícil les proves de correctesa, ja que s'haurien de realitzar càlculs molt extensos manualment. Tot i ser volums de dades més petits dels reals, s'ha intentat que aquestes proves contemplassin la totalitat del possibles casos reals.

8.4.1 Package pool

S'han dissenyat i realitzat dos tipus de proves per aquest ordre:

- Comprovació de la correctesa de la lectura de dades.
- Comprovació de la correctesa del càlcul dels cicles.

8.4.2 Package poolOptimizer

S'han dissenyat i realitzat sis tipus de proves per aquest ordre:

- Comprovació de la correctesa de les solucions obtingudes d'optimitzacions, comprovant cada criteri per separat
- Comprovació de la correctesa de les solucions obtingudes d'optimitzacions, amb diferents configuracions de diferents criteris
- Comprovació del correcte funcionament del procés de fallades contemplant cada tipologia de fallada per separat
- Comprovació del correcte funcionament del procés de fallades contemplant les tipologies de fallades juntes
- Comprovació de la correctesa de les solucions obtingudes de reparacions, comprovant cada criteri per separat
- Comprovació de la correctesa de les solucions obtingudes de reparacions, amb diferents configuracions de diferents criteris

8.4.3 Interfície d'usuari (GUI)

- Comprovació del correcte funcionament de totes les funcionalitats, en l'ús correcte de la interfície
- Comprovació del correcte funcionament quan hi ha sobreescritura de noms
- Comprovació del correcte funcionament de totes les funcionalitats, en l'ús erroni de la interfície (optimitzacions sense criteris, etc)

9. Implantació i resultats

9.1 Implantació

Cal tenir present que l'aplicació desenvolupada és un prototipus amb un doble propòsit: transferència a curt o mig termini a l'ONT i, d'altra banda, ús per a finalitats de recerca. Per tant, la implantació no està contemplada en aquest projecte, però sí que s'ha fet una planificació d'aquesta que constaria de dues branques diferenciades i, que és la següent.

➤ Implantació a la ONT:

- I. Distribució de la versió distribuïble del programa a la ONT.
- II. Període de proves del programa per part del personal de l'ONT.
- III. Reunions amb el personal informàtic i mèdic de l'ONT, per determinar possibles errors i millores.
- IV. Modificació del programa per tal de solucionar els errors i complir amb les millores i requisits de sortides demanats.

➤ Distribució del programa:

- I. Compra d'un domini web o disposar d'un allotjament.
- II. Creació de la web on s'allotjaria la versió distribuïble del programa.

9.2 Resultats

A continuació detallo tots els requisits del programa i el seu assoliment:

- ✓ Creació del graf de compatibilitats a partir del fitxer d'entrada amb pacients i donants, càlcul de tots els possibles cicles i obtenció, com a fitxer de sortida, el llistat d'aquests cicles, junt amb informacions rellevants del pool de pacients/donants i dels cicles obtinguts.

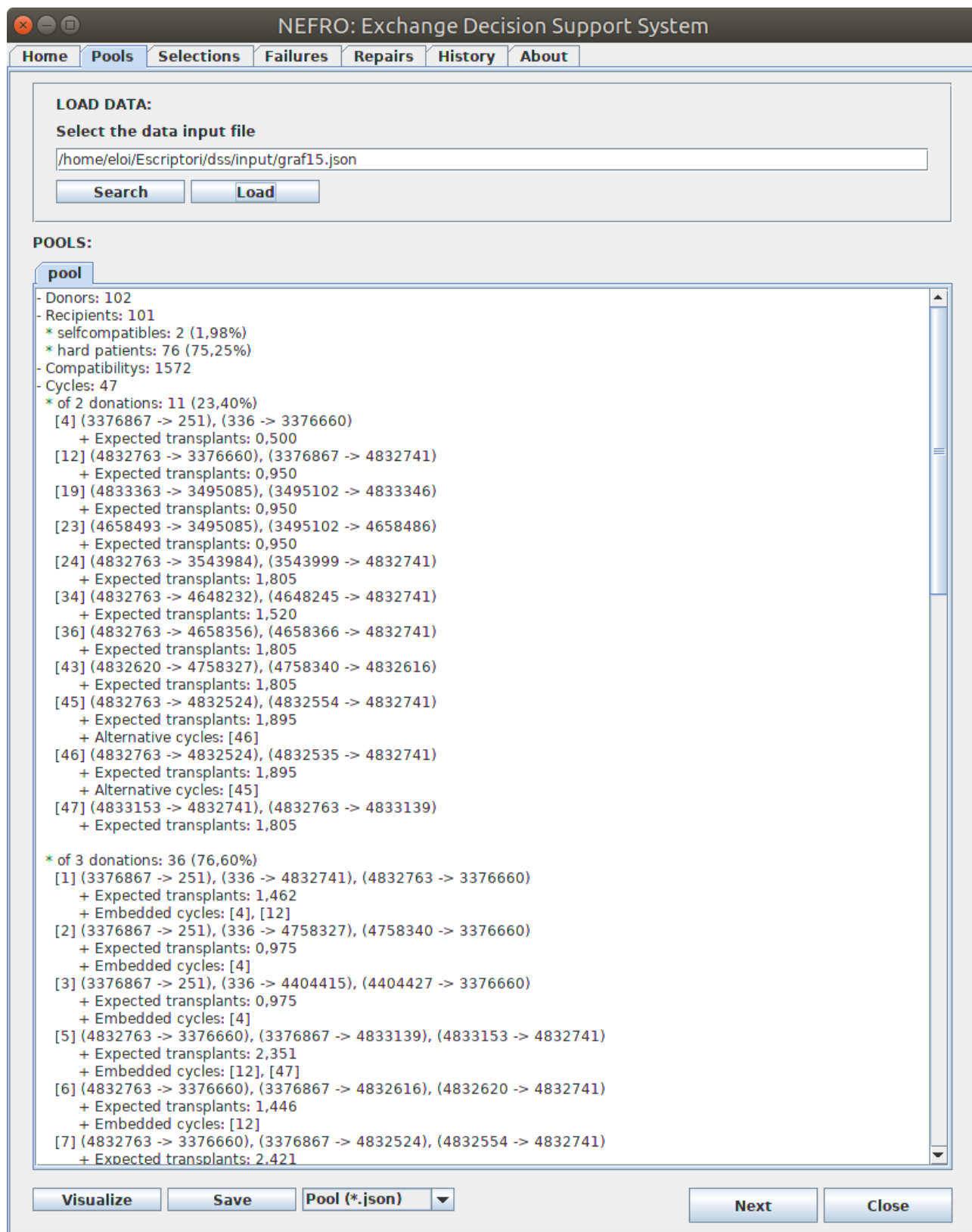


Fig. 19. Obtenció del llistat de cicles a partir de les dades de pacients i donants

- ✓ **Fer optimitzacions** triant-ne els criteris i el seu ordre, obtenint les solucions resultants en un temps raonable.

The screenshot displays the NEFRO: Exchange Decision Support System interface. The main window has a menu bar with 'Home', 'Pools', 'Selections', 'Failures', 'Repairs', 'History', and 'About'. The 'Selections' tab is active.

CHOOSE POOL TO OPTIMIZE:

There is a 'Load' button and a dropdown menu showing 'pool'.

CONFIGURE THE OPTIMIZATION:

Two lists of criteria are shown with 'AddAll' and 'RemoveAll' buttons between them:

- Left List:** Expected transplants, Hard patients, Cycles of 2 trasplants, Embeddeds dispersion.
- Right List:** Transplants, Selfcompatible pairs, Alternative cycles, Embedded cycles, Alternatives dispersion, Score.

Give name to this optimization: optimization0 **OPTIMIZE**

OPTIMUM SELECTIONS:

optimization0

selection#1 selection#2

CRITERIAS:
[Transplants, Selfcompatible pairs, Alternative cycles, Embedded cycles, Alternatives dispersion, Score,]

SELECTED CYCLES:
[44] (4832139 -> 4758327), (4758340 -> 4832720), (4832747 -> 4832134)
[28] (4832763 -> 4404415), (4404427 -> 4832524), (4832554 -> 4832741)
[20] (4658493 -> 3495085), (3495102 -> 4833229), (4833296 -> 4658486)
[4] (3376867 -> 251), (336 -> 3376660)

STATISTICS:
- Trasplants: 11
- Score: 1064.7
- Selfcompatibles pairs: 1
- Hard patients: 9
- Alternative cycles: 1
 * Alternatives dispersion: 1
- Embedded cycles: 3
 * Embeddeds dispersion: 2
- Cycles of 2 trasplants: 1
- Expected transplants: 5.6013

Buttons at the bottom: Visualize, Save, Selection (*.json), Next, Close.

Fig. 20. Exemple d'optimització en el programa.

- ✓ **Tria** dels criteris de maximització a utilitzar i de la seva prioritat. Els criteris seleccionables són els següents:
 - ✓ Suma del nombre de **trasplantaments** dels cicles seleccionats.
 - ✓ Suma del nombre de **parelles selfcompatible** dels cicles seleccionats.
 - ✓ Suma del nombre de **pacients difícils** dels cicles seleccionats.
 - ✓ Nombre de **cicles de 2** seleccionats.
 - ✓ Suma de les **puntuacions** dels cicles seleccionats.
 - ✓ Suma del nombre de **cicles embedded** que tenen els cicles seleccionats entre la resta de seleccionables.
 - ✓ Suma del nombre de **cicles alternatius** que tenen els cicles seleccionats entre la resta de seleccionables.

Com es pot veure la figura 20, es poden triar tots els criteris esmentats i tots els possibles ordres entre ells.

- ✓ **Entrar les fallades** de darrera hora (donants o pacients no disponibles i/ o proves de compatibilitat positives) ocorregudes a una selecció i veure'n els efectes.

NEPHRO: Exchange Decision Support System

Home Pools Selections Failures Repairs History About

CHOOSE SELECTION TO CHECK:

Load pool/optimization0/selection#1

CHOOSE FAILURES:

| Recipients | Donors | Compatibilitys | Failures |
|------------|--------|----------------|----------|
| 1 | 3 | 2 -> 3 | 11 -> 2 |
| 2 | 11 | 3 -> 1 | Donor 2 |
| 3 | | | |

Give name to this failures: failures0 CHECK

DONE TRASPLANTS INFO:

failures0

ITERATION NUMBER 1

- Failures:
[Donor 2, Compatibility 11->2,]

- Failed cycles:
[3] (3 -> 1), (11 -> 2), (2 -> 3)

- Done cycles:

- Descarted cycles:
[4] (3 -> 1), (1 -> 2), (2 -> 3)
[2] (2 -> 1), (1 -> 2)
[1] (2 -> 1), (11 -> 2)

- Trasplants: 0
- Score: 0.0
- Selfcompatibles pairs: 0
- Hard patients: 0
- Done cyles: 0
* Of 2 trasplants: 0
* Of 3 trasplants: 0

TOTAL DONE:

- Done cycles:
- Trasplants: 0
- Score: 0.0
- Selfcompatibles pairs: 0
- Hard patients: 0
- Done cyles: 0
* Of 2 trasplants: 0
* Of 3 trasplants: 0

Visualize Save Selection (*.json) Next Close

Fig. 21. Exemple d'entrada i procés de fallades en el programa.

- ✓ **Calcular reparacions** a partir de les fallades indicades, triant-ne de nous criteris de selecció el seu ordre, i veure les solucions resultants en un temps raonable.

NEPHRO: Exchange Decision Support System

Home Pools Selections Failures **Repairs** History About

CHOOSE SELECTION TO REPAIR:

pool/optimization0/selection#1/failures0

CONFIGURE THE REPAIRS CRITERIA:

Intracycle reparations

Expected transplants

Selfcompatible pairs

Hard patients

Score

Cycles of 2 trasplants

Alternative cycles

Alternatives dispersion

Embedded cycles

Embeddeds dispersion

Preselected recipients

Transplants

Give name to this reparation:

OPTIMUM REPARATIONS:

reparation0

selection#1

CRITERIAS:
[Preselected recipients, Transplants,]

SELECTED CYCLES:
[4] (3 -> 1), (1 -> 2), (2 -> 3)

STATISTICS:
- Trasplants: 2
- Score: 186.9
- Selfcompatibles pairs: 1
- Hard patients: 1
- Alternative cycles: 0
 * Alternatives dispersion: 0
- Embedded cycles: 0
 * Embeddeds dispersion: 0
- Cycles of 2 trasplants: 1
- Expected transplants: 0.95
- Preselected recipients: 2
- Intracycle raparations: 1

Selection (*.json)

Fig. 22. Exemple de reparacions en el programa.

- ✓ **Triar** els criteris a maximitzar en les reparacions d'entre els següents (a més dels criteris d'optimització indicats prèviament):
 - ✓ Nombre de pacients **preseleccionats**
 - ✓ Nombre de reparacions **internes**

Com es pot veure a la figura 22, es poden triar tots els criteris esmentats i tots els possibles ordres entre ells.

- ✓ **Guardar qualsevol estat** del procés (pool, selecció, fallades i reparacions) en fitxers en format *Json*.

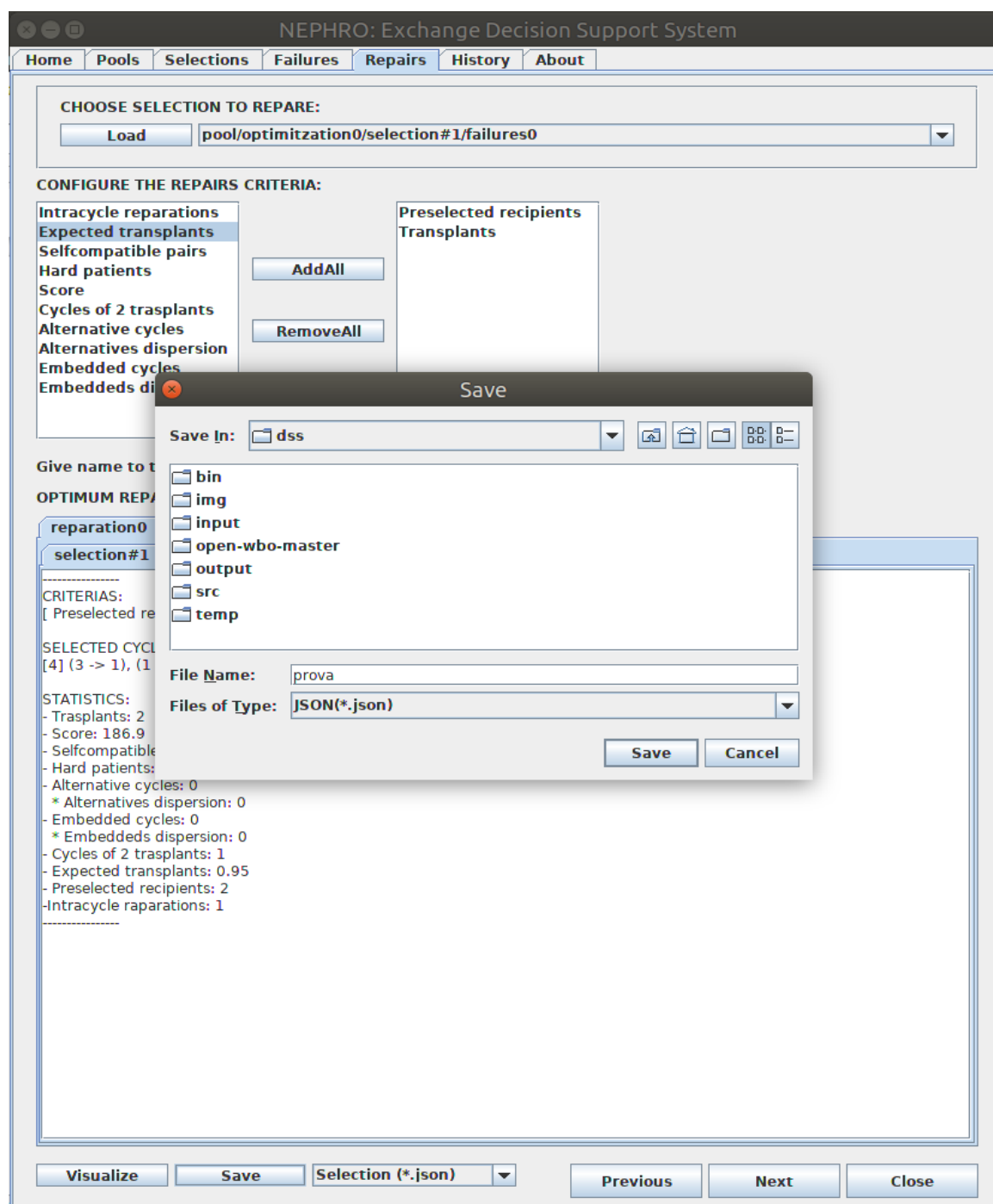


Fig. 23. Exemple de guardat d'un estat.

- ✓ **Carregar** els fitxers format Json del punt anterior i seguir treballant-hi.

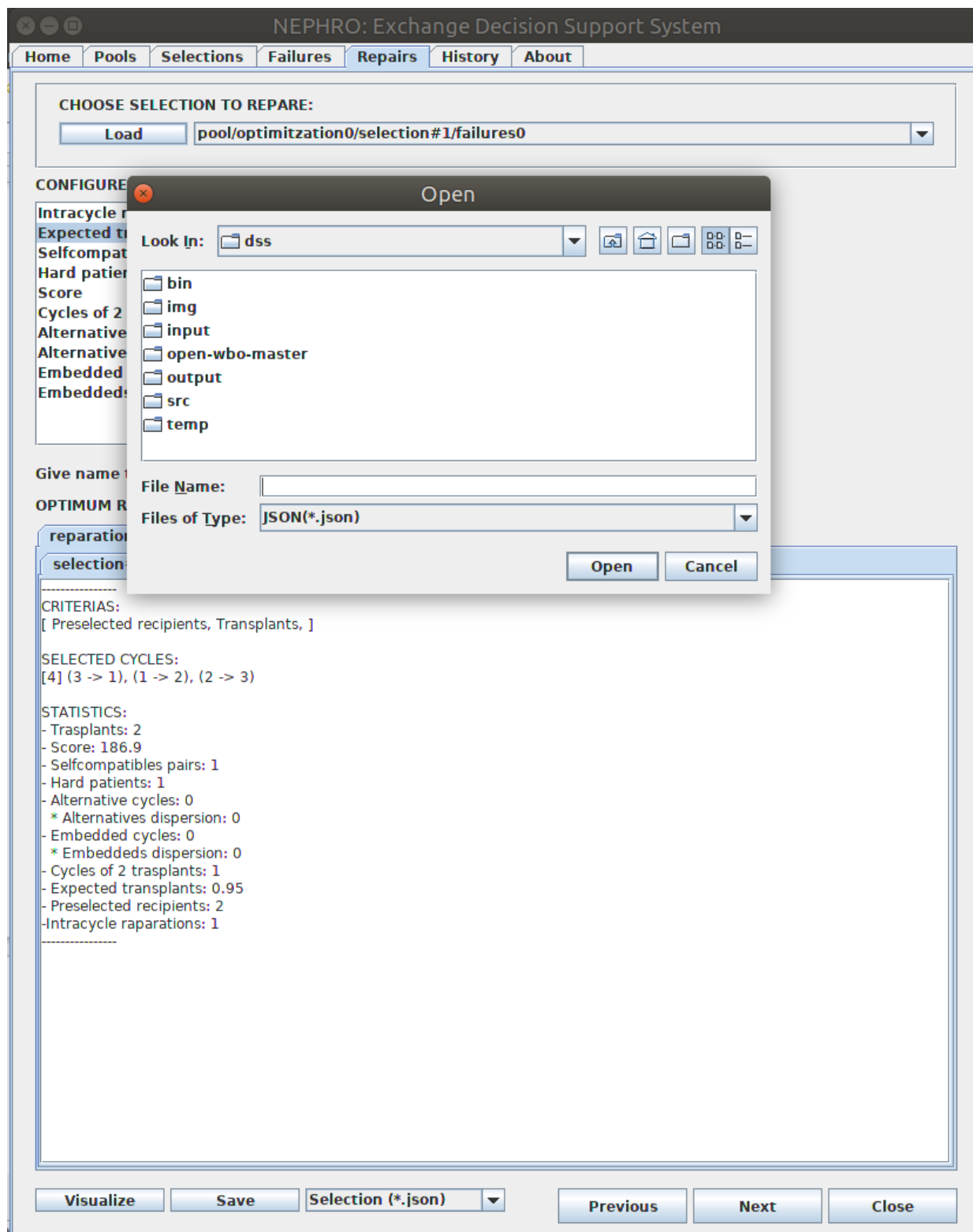


Fig. 24. Exemple de carregat de dades.

- ✓ **Guardar la informació** de qualsevol estat del procés (pool, selecció, fallades i reparacions) en format *txt* de forma llegible per a l'usuari.

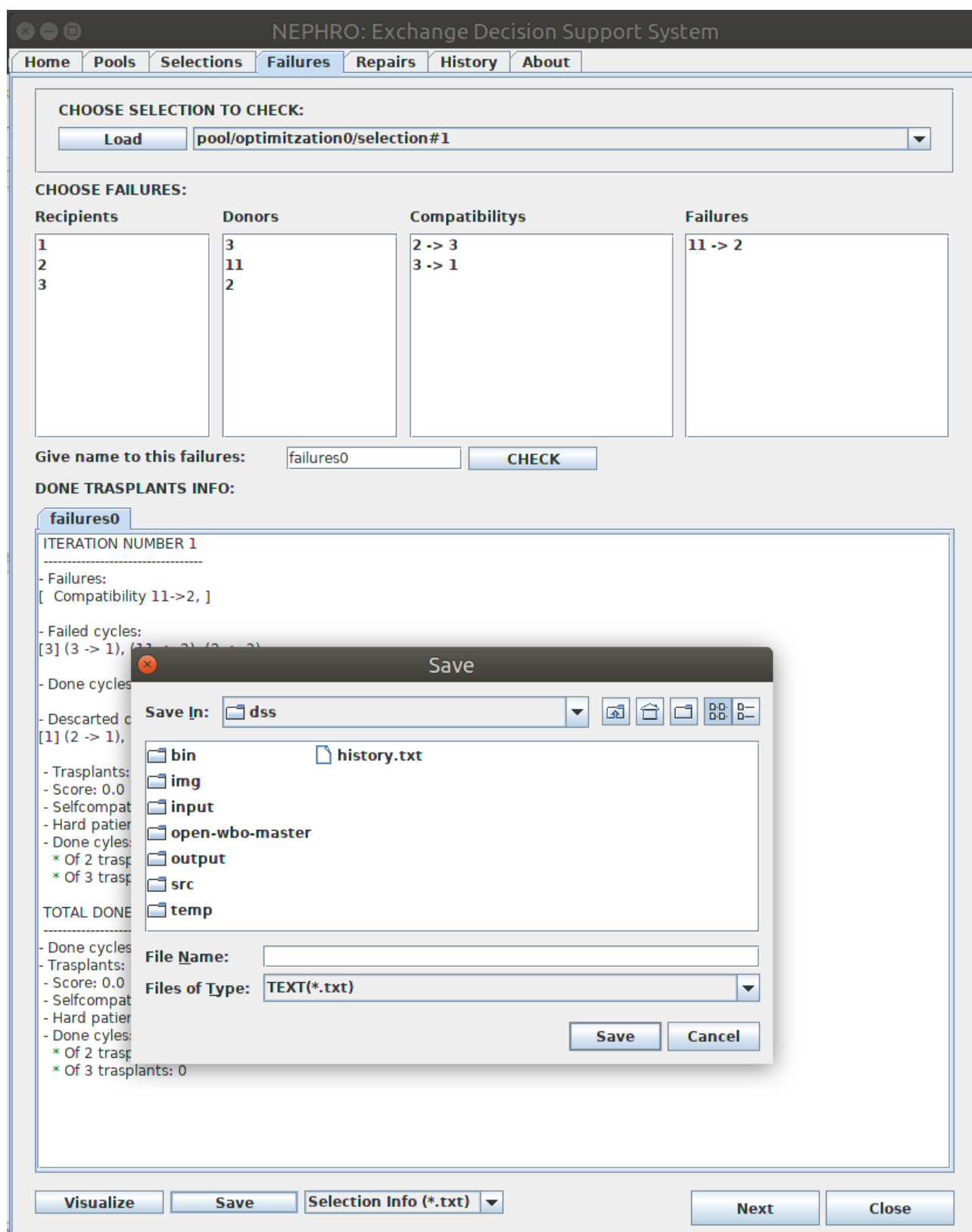


Fig. 25. Exemple de guardat de dades en format llegible.

- ✓ **Treballar a la vegada amb diferents** pool's, optimitzacions, seleccions, fallades i reparacions de forma ordenada i clara.

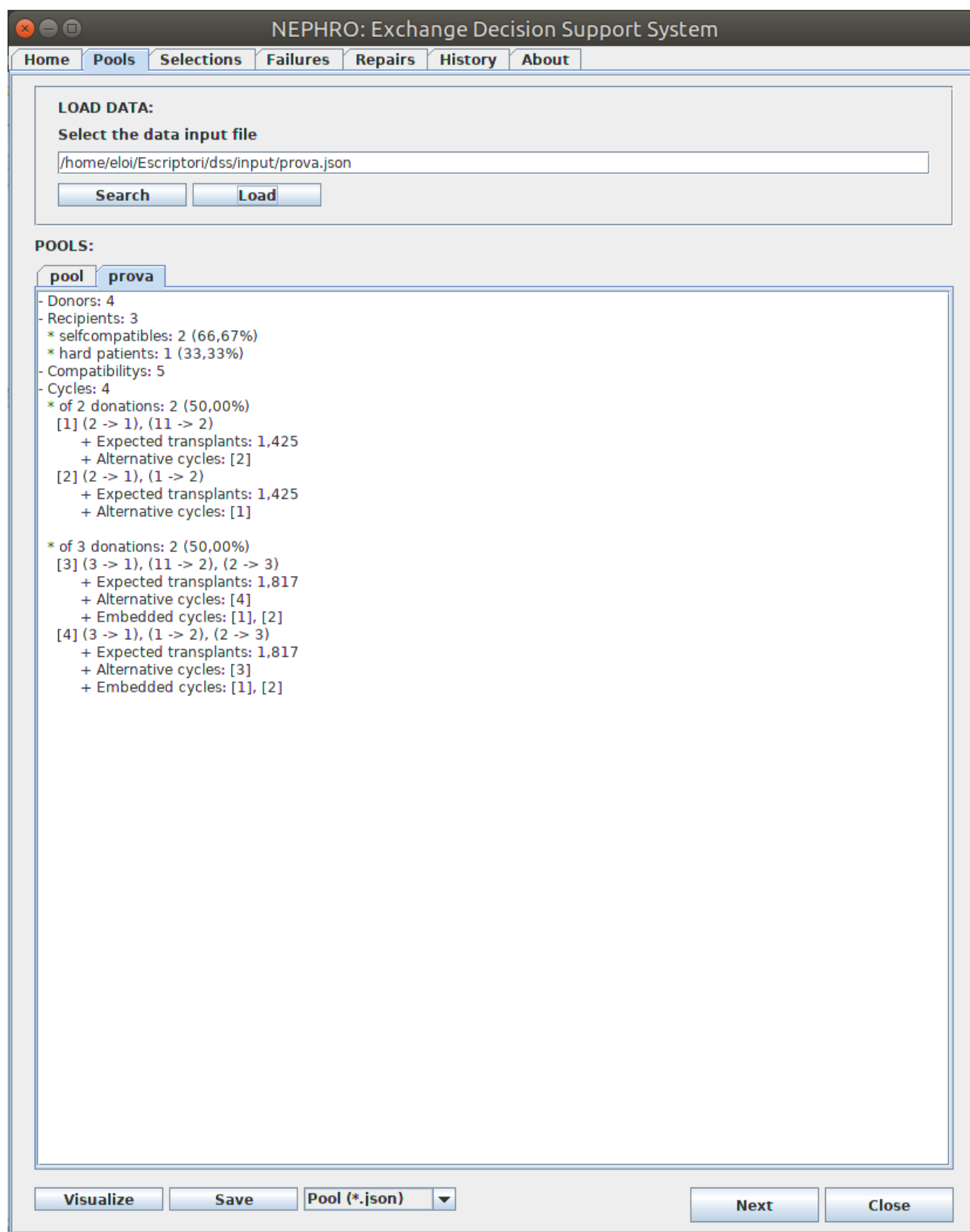


Fig. 26. Exemple de treball amb dos pools.

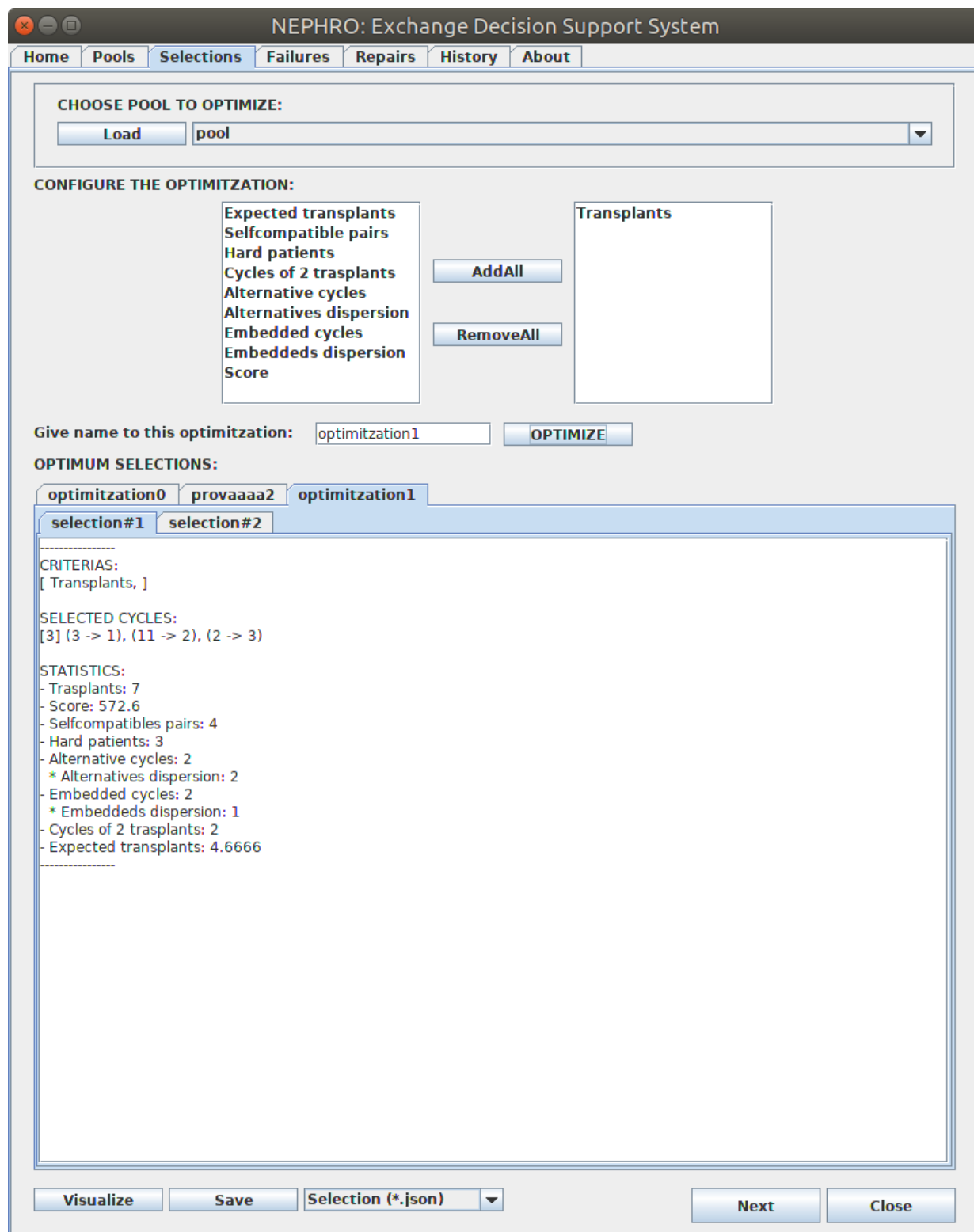


Fig. 27. Exemple de treball amb tres optimitzacions i les seves solucions.

- ✓ Comparar de manera fàcil, ordenada i clara els resultats de diferents optimitzacions, seleccions, fallades o reparacions fetes sobre el mateix punt de partida.

Com es pot veure a la figura 27, es poden comparar d'una manera fàcil i ordenada entre diferents optimitzacions i seleccions fetes sobre el mateix pool.

- ✓ **Obtenir informació de tots els passos fets** fins al moment de qualsevol selecció, fallades i reparacions.

NEPHRO: Exchange Decision Support System

Home Pools Selections Failures Repairs **History** About

LOAD/CHOOSE POOL OR SELECTION TO SEE HIS INFORMATION:

pool

Recipients

1

- Id: 1
- ABO: 0
- CCAA: 1
- Dialysis months: 0
- Enter pool date: 2011-01-01
- Selfcompatible: false
- Hard patient: true
- Linked Donors: [1, 11]

Donors

1

- Id: 1
- Linked recipient: 1

Compatibilities

1

- Id: 1
- Donor: 3
- Recipient: 1
- Score: 75,0
- Failure probability: 5.0%

Cycles

1

- Id: 1
- Donations: 2
- * [2] 2 -> 1
- * [4] 11 -> 2
- Selfcompatible pairs: 1
- Hard patients: 1
- Score: 86.9

History:

- Donors: 4
- Recipients: 3
- * selfcompatibles: 2 (66,67%)
- * hard patients: 1 (33,33%)
- Compatibilities: 5
- Cycles: 4
- * of 2 donations: 2 (50,00%)
- [1] (2 -> 1), (11 -> 2)
- + Expected transplants: 1,425
- + Alternative cycles: [2]
- [2] (2 -> 1), (1 -> 2)
- + Expected transplants: 1,425
- + Alternative cycles: [1]
- * of 3 donations: 2 (50,00%)
- [3] (3 -> 1), (11 -> 2), (2 -> 3)
- + Expected transplants: 1,817
- + Alternative cycles: [4]
- + Embedded cycles: [1], [2]
- [4] (3 -> 1), (1 -> 2), (2 -> 3)
- + Expected transplants: 1,817
- + Alternative cycles: [3]
- + Embedded cycles: [1], [2]

Fig. 28. Exemple de visualització de la informació.

- x **Visualitzar** el graf de compatibilitats, el subgraf selecció i/o les fallades ocorregudes, de forma gràfica i a poder ser interactiva.

Per falta de temps no s'ha complert aquest objectiu, tot i que s'ha contemplat l'opció. Com es pot veure en la figura 27 hi ha el botó que permetria fer-ho (botó «Visualize»), tot i que no funciona.

- **Triar** els criteris a maximitzar en les optimitzacions i reparacions d'entre els següents (a més dels obligatoris):
 - Nombre de cicles seleccionats que tenen **almenys un cicle embedded** entre la resta de seleccionables.
 - Nombre de cicles seleccionats que tenen **almenys un cicle alternatiu** entre la resta de seleccionables.

Com es pot veure a la figura 108, es poden triar aquests dos criteris, anomenats «embedded dispersion» i «alternatives dispersion» respectivament.

- ✓ **Fer** que l'aplicació sigui executable en diferents entorns.

Com es pot veure al manual d'instal·lació s'ha complert.

- ✓ **Facilitar** la instal·lació i l'ús de l'aplicació.

Com es pot veure al manual d'instal·lació i al manual d'ús també s'han complert.

Aquest programa compleix amb la legislació i normatives vigents, en el que fa referència a:

- Llei orgànica de protecció de dades de caràcter personal (LOPD) atès que, és impossible identificar cap persona concreta, donat que vénen identificats per nombres enters ja a les dades d'entrada.
- Llei de serveis de la societat de la informació i comerç electrònic (LSSICE), el solver usat consta d'una llicència per copyright que dona llibertat a utilitzar-la sempre i quant es mostri els autors i aquesta mateixa llicència, cosa que s'ha fet, i els llenguatges, les llibreries i les eines de compilació i execució usades són d'accés i ús obert.

Aquest programa es distribuirà sota una llicència Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

10. Conclusions

10.1 Assoliment d'objectius

Els principals objectius d'aquest projecte eren:

- ✓ Creació d'una aplicació informàtica d'ajuda a la presa de decisions al personal mèdic que gestiona els intercanvis de ronyó; aquest programa els ha de permetre comparar possibles planificacions, triar-ne una, gestionar les fallades de darrera hora i replanificar els intercanvis a partir de les fallades sofertes, i els ha d'oferir informació detallada dels trasplantaments realitzats efectivament, de les fallades experimentades, etc.
- ✓ Explorar i comparar el comportament de diferents metodologies per a la resolució de problemes combinatoris, per a resoldre el problema de selecció dels cicles de trasplantaments a realitzar. Les tecnologies considerades són Programació amb restriccions (per exemple mitjançant MiniZinc i algun dels seus resoladors) i MaxSAT (per exemple el solver Open-WBO).
- ✓ Revisar i incorporar al programa la possibilitat d'ús de nous criteris (no utilitzats actualment al KEP espanyol), entre ells l'estimació probabilística del nombre de trasplantaments que es realitzaran, així com, la distribució entre cicles de les possibles reparacions (fins ara s'utilitza el nombre de reparacions totals sense tenir en compte la distribució d'aquestes).

S'han aconseguit assolir tots els objectius marcats a l'inici del projecte. En el tercer punt, més concretament, s'han introduït tres criteris no usats a l'ONT en el procés de selecció de cicles. Aquests nous criteris són:

- Nombre esperat de trasplantaments
- Dispersió d'embeddeds
- Dispersió d'alternatius

El més innovador és el primer, ja que els altres dos són variants de criteris usats fins ara. També és el més complet ja que té en compte tant les probabilitats de fallada dels cicles seleccionats en primera instància com les probabilitats de fallada dels cicles incorporats al procés de reparació. Tots tres criteris aporten un valor afegit a la feina i al programa, ja que permeten en certa manera quantificar la robustesa dels cicles, per tal de comparar-los i triar els millors en aquest sentit.

10.2 Assoliment dels requisits del programa desenvolupat

S'ha assolit tots els requisits del programa així com, totes les funcionalitats opcionals menys una:

- x **Visualitzar** el graf de compatibilitats, el subgraf selecció i/o les fallades ocorregudes, de forma gràfica i a poder ser interactiva.

No és problemàtic, ja que no és un aspecte fonamental del programa, simplement és una eina alternativa de visualització de resultats. En contrapartida s'ha implementat una sortida de resultats en format text molt completa.

D'altra banda, vull remarcar que s'ha aconseguit un temps de resposta molt baix pel que fa al càlcul del conjunt de cicles seleccionats (tant en primera instància com a la fase de reparacions). Cal tenir en compte que aquest càlcul suposa un problema combinatori dur. Aquest cost computacional tan reduït s'ha aconseguit mitjançant un bon modelatge i codificació del problema i també gràcies a la tria d'un *solver* molt adequat a aquesta tipologia de problemes.

Per tant el grau d'assoliment dels requisits del programa desenvolupat és molt alt, i els resultats obtinguts són satisfactoris.

10.3 Desviacions en la planificació original

La planificació va ser molt optimista, ja que els temps reals d'estudi i implementació han estat força superiors als planificats en 3 de les 7 fases en què es va dividir el desenvolupament del projecte, arribant a doblar en algunes el temps planificat. Això ha provocat l'endarreriment de la data d'entrega, aspecte no molt crític i que ja estava contemplat.

En definitiva era una bona planificació en quant a plantejament de fases, s'han seguit totes, però no en els temps planificats, tres de les 7 totals no s'han assolit en els terminis planificats.

10.4 Crítica dels resultats

El resultats han estat molt bons, però sempre hi ha aspectes a millorar, entre ells en destaco dos:

- La comparació de llenguatges i solvers només ha contemplat dues tipologies diferents. Es podia haver introduït a la comparació una tercera tipologia, Mixed Integer Linear Programming (MILP), però no s'ha fet per falta de temps. Tot i així no és un aspecte crític ja que amb un dels dos procediments usats s'han obtingut molt bons resultats.
- El procediment per calcular les solucions és «millorable». Degut a la falta de coneixement en la codificació de restriccions pseudobooleanes, no s'han per fer optimitzacions de diferents criteris a la vegada, prescindint així dels filtratges posteriors. Estava plantejat i fins i tot vaig arribar a estudiar diferents possibilitats de codificació d'aquestes, però finalment no se n'ha implementat cap per falta de temps. Tot i que, si augmenta molt el volum de dades a tractar, podria arribar a ser un aspecte crític del programa, ara per ara no ho és ja que el marge de creixement

d'aquests és gran i no està previst que de moment que hi hagi n'hi hagi d'haver d'importants.

En definitiva, hi ha un parell d'aspectes millorables rellevants, però que no són crítics per al funcionament d'aquest programa.

11. Treball futur

Tot i ser un projecte força extens, hi ha moltes possibles ampliacions i treballs a fer per completar-lo i millorar-lo. Tot seguit enumero les que, des del meu punt de vista, són més rellevants:

- Incorporar donants altruistes, pacients de la llista d'espera donant cadàver i el càlcul de les cadenes de trasplantaments al programa.
- Implementar un mòdul de càlcul de les compatibilitats i les seves puntuacions a partir de les dades dels donants i pacients. Completaria tot el procés seguit en els KEP's.
- Investigar i definir com calcular les probabilitats de fallada de les donacions i incorporar aquests càlculs al mòdul del punt anterior.
- Implementar una base de dades de recopilació de donants i pacients i lligar-la amb el programa de manera que les dades d'entrada les prengui de la base de dades.
- Investigar com determinar les probabilitats de fallada dels pacients i dels donants segons criteris mèdics i incorporar-les a la formulació probabilística del criteri «trasplantaments esperats».
- Implementar un mòdul de visualització de pools (graf de compatibilitats), seleccions (subgrafs del graf de compatibilitats), fallades (ressaltant les fallades dels subgrafs que formen la selecció) i les reparacions de forma similar a les seleccions. Existeixen força llibreries *Java* de visualització de grafs.
- Implementar un mòdul de selecció manual, que permeti a l'usuari fer seleccions de cicles manualment. Podria anar lligat amb el mòdul de visualització, fent-la interactiva.

En definitiva, hi ha moltes possibilitats d'ampliació i millora del programa, que crec, és una molt bona base que inclou tot els aspectes bàsics i un xic més.

12. Bibliografia

- [1] Guirado i Perich, Lluís. (2012). El trasplantament renal a Catalunya. *Annals de Medicina*, 95, 154-156. Recuperat el 20 d'agost de 2018 de: www.raco.cat/index.php/AnnalsMedicina/article/download/283294/371191
- [2] Domínguez-Gil, B., de la Oliva Valentín, M., Martín Escobar, E., Cruzado, J., Pascual, J., i Fernández Fresnedo, G. (2010). Situación actual del trasplante renal de donante vivo en España y otros países: pasado, presente y futuro de una excelente opción terapéutica. *Revista Nefrología*, (Suppl 2):3-13. doi: 10.3265/Nefrologia.pre2010.Nov.10686
- [3] Organización Nacional de Trasplantes. (2017). *Memoria de Actividad. ONT 2017*. Recuperat el 20 d'agost de 2018 de: http://www.ont.es/infesp/Memorias/Memoria%20Renal%202017_para%20colgar.pdf
- [4] Diversos autors. (Agost de 2018). *Modelling and Optimisation in European Kidney Exchange Programmes*. Proporcionat pels tutors del projecte.
- [5] Gabinete de prensa del MINISTERIO DE SANIDAD, CONSUMO Y BIENESTAR SOCIAL. (Agost de 2018). Nota de premsa sense títol. Recuperat el 20 d'agost de 2018 de: <http://www.ont.es/Documents/NP%208%20de%20agosto%20de%202018%20-%20Espa%C3%B1a%20e%20Italia%20protagonizan%20el%20primer%20trasplante%20renal%20cruzado%20internacional.pdf>
- [6] Mouzo Quintáns, Jessica. (9 d'agost de 2018). El primer trasplante cruzado a ambos lados del Mediterráneo. *El País*. Recuperat el 20 d'agost de 2018 de: https://elpais.com/elpais/2018/08/08/ciencia/1533749337_297560.html
- [7] Wikipedia. (2018). *Boolean satisfiability problem*. Extret el 27 d'agost de : https://en.wikipedia.org/wiki/Boolean_satisfiability_problem
- [8] Wikipedia. (2018). *Maximum satisfiability problem*. Extret el 27 d'agost de : https://en.wikipedia.org/wiki/Maximum_satisfiability_problem
- [9] Valentín, María O., García, Marta., Delgado, Pablo. (2017). *Description of the Spanish National Kidney Exchange Program*. Organización Nacional de Trasplantes. Facilitat pels tutors del projecte.
- [10] Manlove, David. (2014). *Models and algorithms for the UK Living Kidney Sharing Schemes*. Facilitat pels tutors del projecte.

- [11] Santos, Nicolau., Viana, Ana., Klimentova, Xenia., Rais, Abdur. (2018). *Description of the DSS used by the Portuguese Kidney Exchange Program*. Facilitat pels tutors del projecte.
- [12] Andersson, Tommy. (2018). *Models and Algorithms for the Scandiatransplant Kidney Exchange Program*. Facilitat pels tutors del projecte.
- [13] Viquipèdia. (2018). Scrum. Recuperat el 21 d'agost de 2018 de <https://ca.wikipedia.org/wiki/Scrum>
- [14] Extret el 27 d'agost de: <https://www.scala-lang.org/>
- [15] Extret el 27 d'agost de: <http://www.minizinc.org/>
- [16] Extret el 27 d'agost de: <http://sat.inesc-id.pt/open-wbo/>

13. Annexos

Es troben a l'usb adjunt a l'entrega i són:

- Exemple d'input de dades (graf15.json)
- Formulació del criteri probabílistic «trasplantaments esperats» (formulacio.pdf)
- Algorismes de simulació de Montecarlo usats per comprovar la correctesa de la formulació del punt anterior (simulador.scala)
- Model i variants de codificació del problema de selecció d'elements incompatibles entre ells amb SAT (codificacio.pdf)

*A partir d'aquí la numeració de les figures es reinicia, ja que el manual d'usuari té la seva pròpia numeració que només té efecte dins del capítol 14. Manual d'usuari/o instal·lació.

14. Manual d'usuari i/o instal·lació

Aquest programa «NEPHRO» ha estat provat en sistemes operatius Linux i Windows. En Mac OS no s'ha provat i per tant no es pot garantir el seu bon funcionament.

14.1 Manual d'instal·lació

L'únic prerequisite per a la utilització de *NEPHRO* és tenir instal·lada la *Màquina Virtual de Java* (<https://www.java.com/es/>).

- Si no saps què és, pots esbrinar-ho visitant aquest enllaç:

https://www.java.com/es/about/whatis_java.jsp

- Si no saps si la tens instal·lada, per esbrinar-ho visita aquest enllaç:

<https://java.com/es/download/installed.jsp>

- Si no la tens instal·lada, pots descarregar-la al següent enllaç:

<https://www.java.com/es/download/>

- Si necessites ajuda, o vols més informació, visita aquest enllaç:

<https://www.java.com/es/download/manual.jsp>

- Si necessites més ajuda, visita aquest enllaç:

<https://www.java.com/es/download/support.jsp>

14.2 Manual d'usuari

NEPHRO és un sistema de suport a la decisió d'intercanvis, pensat i dissenyat per intercanvis de ronyons, d'aquí en ve el nom, però que pot ser utilitzat per intercanvis d'altres òrgans, en que es segueixi un procés similar.

Està compost de 6 pestanyes, cadascuna pensada per a una funcionalitat diferent dins del procés d'intercanvi. Són les següents:

- «Home» benvinguda i tria, si es vol, del directori de treball.
- «Pools» creació i visualització de les dades de les recopilacions de donants i pacients i les seves compatibilitats.
- «Selections» permet fer, comparar i veure la informació de diferents planificacions inicials d'intercanvis sobre diferents *pool's*.
- «Failures» permet entrar fallades ocorregudes sobre planificacions i replanificacions concretes i veure'n informació relacionada.
- «Repairs» permet fer, comparar i veure la informació de replanificacions (reparacions) sobre les fallades ocorregudes.
- «History» permet veure informació de donants, pacients, compatibilitats i cicles de *pool's* concrets, així com veure l'historial i l'estat actual de planificacions, fallades i replanificacions concretes.
- «About» permet veure informació de *NEPHRO*: la llicència, l'autor del programa, software extern que utilitza i les seva llicència, etc.

14.2.1 Moure's per les pestanyes

Es pot fer de dues maneres:

- Clicant sobre la pestanya on es vol anar a la part superior de la finestra. La ressaltada en color blau és la pestanya actual.

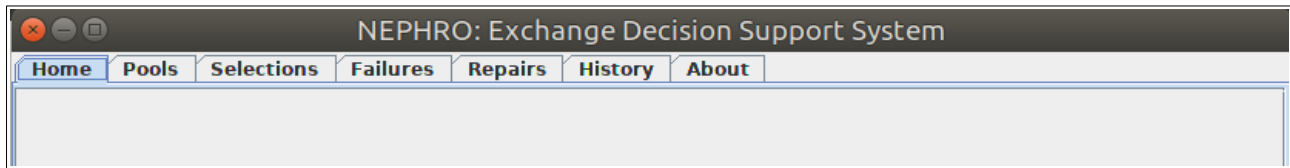


Fig. 1. Pestanyes del programa

- Clicant els botons de navegació de cada pestanya, que es troben a la part inferior dreta de la finestra.

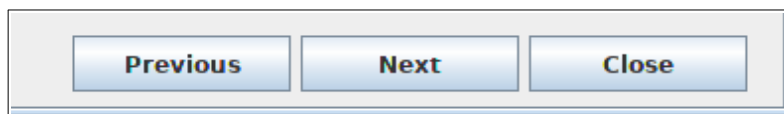


Fig. 2. Botons de navegació.

- i. «Previous» passar a la pestanya anterior.
- ii. «Start/Next» passar a la pestanya següent.
- iii. «Close» tancar l'aplicació.

14.2.2 Pestanya «Home»

Dona la benvinguda a al programa i serveix per triar, si es vol, el directori de treball. Aquest defineix el directori on començarà la cerca del directori des d'on es vol guardar o carregar fitxers.

No és obligatori triar-ne un, però fa el procés de guardat i carregat de fitxer (veure apartat 15.2.9) més còmode. Pots afegir i triar el directori sempre que vulguis.

- i. Afegir directoris: es pot tenir més d'un directori guardat. Però només un, el que triïs, serà l'utilitzat. Es poden afegir directoris clicant al botó «Add» de la part inferior esquerra de la pestanya. (Veure fig. 3).
- ii. Triar el directori de treball actual: només es pot tenir un directori triat. Aquest serà l'utilitzat pel programa mentre no el canviïs. Es pot triar clicant sobre el desplegable, per veure tots els disponibles (els afegits fins al moment), i triar clicant el que es vol. Després de fer-ho veuràs que és el que es veu. (Veure fig. 3.).

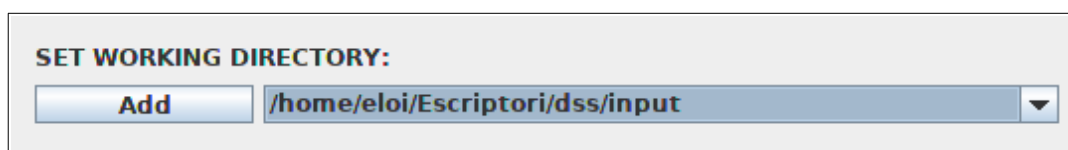


Fig. 3. Tria del directori de treball.

No cal triar-lo cada cop que obris el programa, ja que aquest, guarda tots els que hagi afegit i deixa com a directori actual de treball l'últim que hagi triat. Quan tornis a obrir el programa estaran tal i com els vas deixar quan el vas tancar.

14.2.3 Pestanya «Pools»

Aquesta pestanya permet la creació i visualització de les dades de les recopilacions de donants i pacients i les seves compatibilitats, obtinguts del fitxer d'entrada especificat.

- i. Tria del fitxer: hi ha dues maneres de fer-ho, totes dues a la part superior de la pestanya.

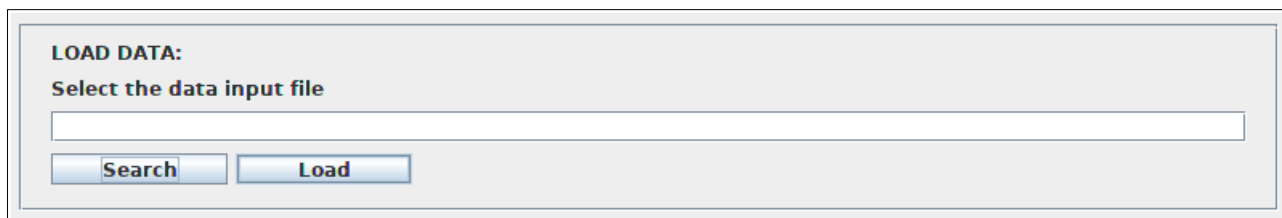


Fig. 4. Creació de pools.

1ª. Clicant el botó «Search» (veure fig. 4) a la part superior de la pestanya. S'obrirà una finestra (veure fig. 5) que et permetrà moure't per les carpetes de l'ordinador. Quan estiguis a la carpeta que conté el fitxer que vols, selecciona'l i clica el botó «Open». Un cop fet veuràs que surt la ruta del fitxer al requadre inicialment blanc. Si te n'has desdit, pots clicar el botó «Cancel».

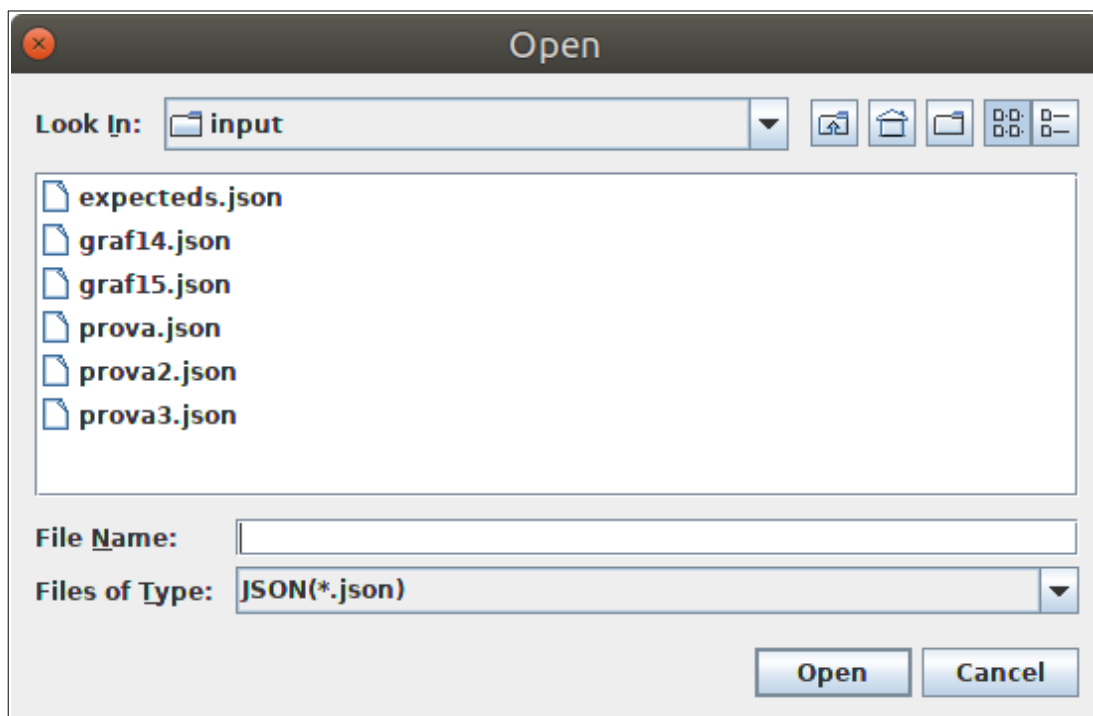


Fig. 5. Finestra per triar el fitxer.

2ª. Escrivint la ruta completa del fitxer al requadre blanc. (veure fig. 4)

- ii. Carregar el fitxer: un cop triat el fitxer, es pot carregar clicant el botó «Load» (veure fig. 4), s'obrirà una petita finestra on li pots donar el nom que vulguis al pool (no pot contenir el caràcter /).

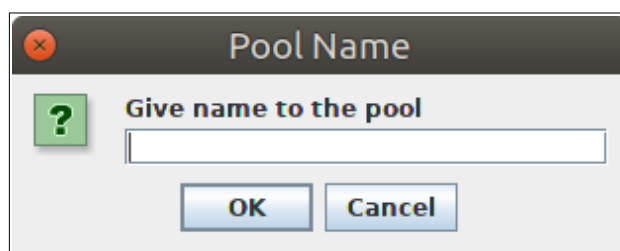


Fig. 6. Finestra per escriure el nom del pool.

En uns segons apareixerà al tauler de pools, una pestanya amb el nom donat al pool i la seva informació.

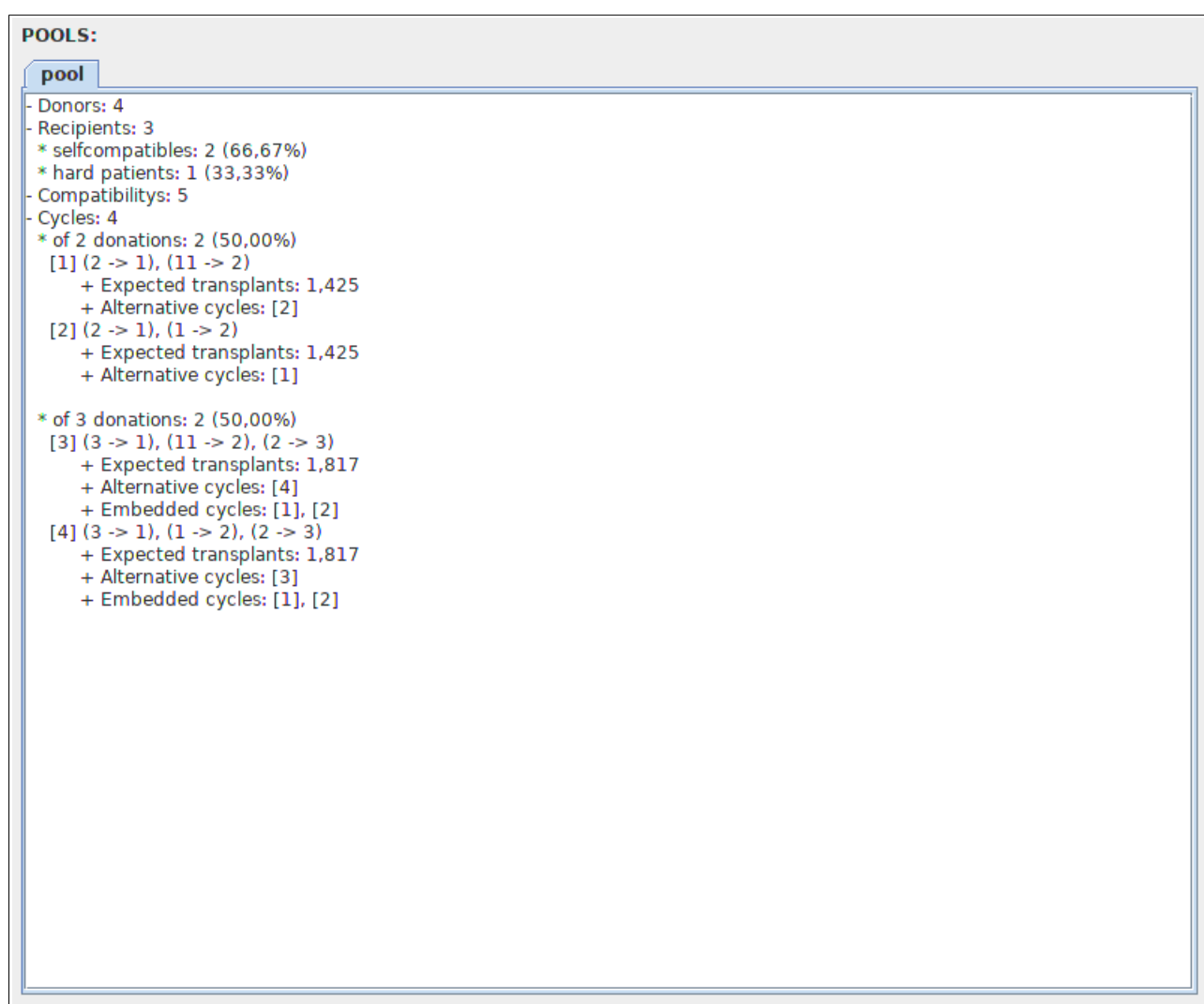


Fig. 7. Tauler de pools

14.2.4 Pestanya «Selections»

Aquesta pestanya permet fer, comparar i veure la informació de diferents planificacions inicials d'intercanvis sobre diferents *pool's*.

- i. Tria del pool: només es pot tenir un pool triat. Aquest serà l'utilitzat pel programa mentre no el canviïs. Es pot triar clicant sobre el desplegable, per veure tots els disponibles (els carregats fins al moment), i triar clicant el que es vol. Després de fer-ho veuràs que és el que es veu.

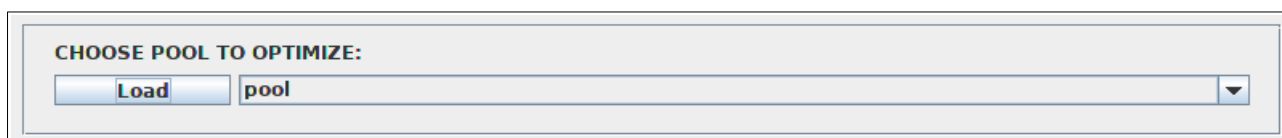


Fig. 8. Tria del pool

- ii. Elecció de criteris i el seu ordre: hi ha dues llistes, inicialment la de l'esquerra conté tots els criteris i la de la dreta cap. Es tracta de posar a la segona llista tots els criteris que es vulguin utilitzar, i deixar-los ordenats, el primer és el de dalt de tot i així anar baixant. Per moure'ls d'una llista a l'altra hi ha dues maneres:

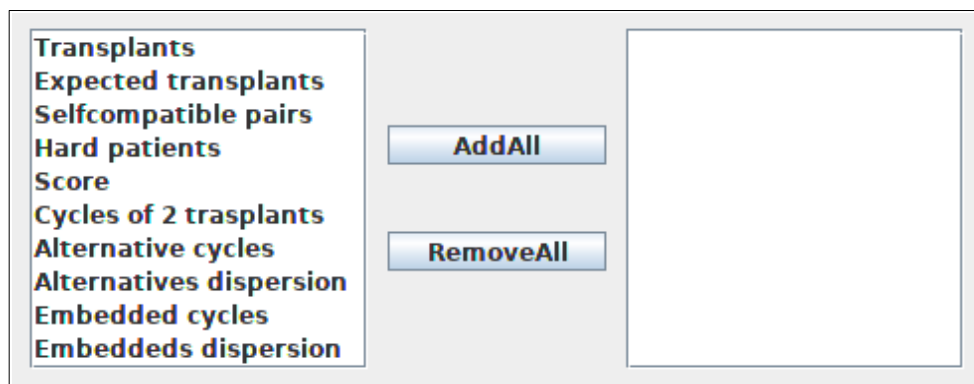


Fig. 9. Llistes d'elecció de criteris de la pestanya «Selections».

1ª. Fent doble clic sobre el criteri que vulguis utilitzar.

2ª. Seleccionant tots els que vulguis i clicant el botó «AddAll» per passar-los de la primera llista a la segona o clicant el botó «RemoveAll», per passar-los de la segona llista a la primera.

Per triar-ne l'ordre, manté clicat el criteri que vulguis i arrossega'l fins a la posició desitjada.

- iii. Donar nom a l'optimització: és obligat donar-li nom (no pot contenir el caràcter /). Es fa escrivint el nom al requadre blanc de sota les llistes de criteris.

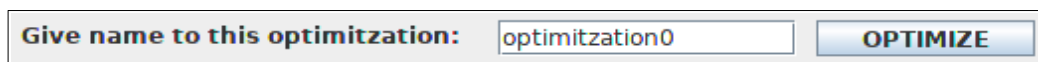


Fig. 10. Tria de nom de l'optimització.

- iv. Executar: un cop fets tots els passos anteriors, clica el botó «OPTIMIZE» i en poc segons apareixerà al tauler de seleccions, una pestanya amb el nom triat al pas iii. amb les diferent seleccions (planificacions) proposades (cadascuna una pestanya) sobre el pool triat al pas i. que maximitzen els criteris triats, en l'ordre triat al pas ii.

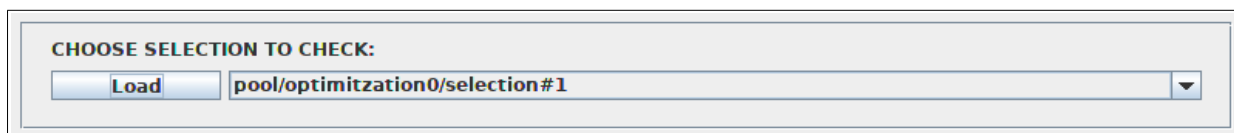


Fig. 11. Tauler de seleccions.

14.2.5 Pestanya «Failures»

Aquesta pestanya permet entrar fallades ocorregudes sobre planificacions i replanificacions concretes i veure'n informació relacionada.

- i. Tria de la selecció: només es pot tenir una selecció triada. Aquesta serà l'utilitzada pel programa mentre no el canviïs. Es pot triar clicant sobre el desplegable, per veure totes les disponibles (les creades fins al moment), i triar clicant la que es vol. Després de fer-ho veuràs que és la que es veu.

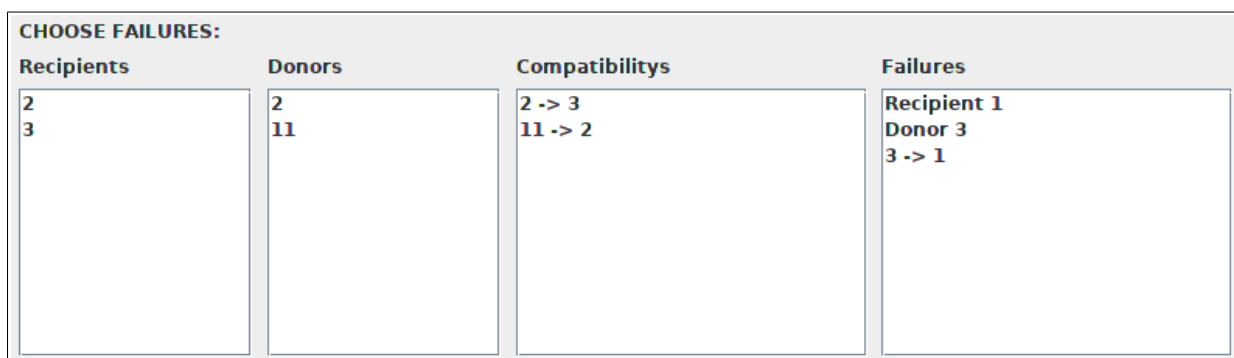


CHOOSE SELECTION TO CHECK:

Load pool/optimization0/selection#1

Fig. 12. Tria de la selecció.

- ii. Tria de les fallades ocorregudes: hi ha 4 llistes, inicialment les tres de l'esquerra contenen tots les possibles fallades de la selecció i la de més a la dreta cap. Es tracta de posar a la última llista totes les fallades que hagin ocorregut. Per moure-les a l'última llista es fa doble clic sobre la fallada en qüestió i veuràs que aquesta canvia de llista. Per treure-les de l'última llista també es fa fent doble clic sobre la falla en qüestió i veuràs que torna a la seva llista corresponent. Poden no haver ocorregut fallades, en aquest cas, simplement deixa la llista de la dreta buida i passa al pas següent.

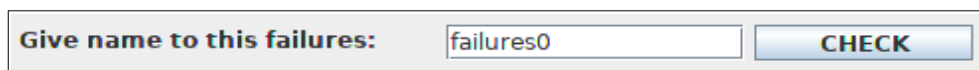


CHOOSE FAILURES:

| Recipients | Donors | Compatibilitys | Failures |
|------------|---------|-------------------|----------------------------------|
| 2 3 | 2 11 | 2 -> 3 11 -> 2 | Recipient 1 Donor 3 3 -> 1 |

Fig. 13. Llistes d'elecció de fallades.

- iii. Donar nom a les fallades: és obligat donar-li nom (no pot contenir el caràcter /). Es fa escrivint el nom al requadre blanc de sota les llistes de d'elecció de fallades.



Give name to this failures: failures0 CHECK

Fig. 14. Tria de nom de l'optimització.

- iv. Executar: un cop fets tots els passos anteriors, clica el botó «CHECK» i en poc segons apareixerà al tauler de fallades (veure fig. 16) una pestanya amb el nom triat al pas iii. amb informació. Si ja no és possible realitzar més intercanvis sortirà una finestra avisant d'això (veure fig. 15). Pots seguir treballant amb el pool, però no amb la selecció triada al pas i, només veure'n la seva informació.

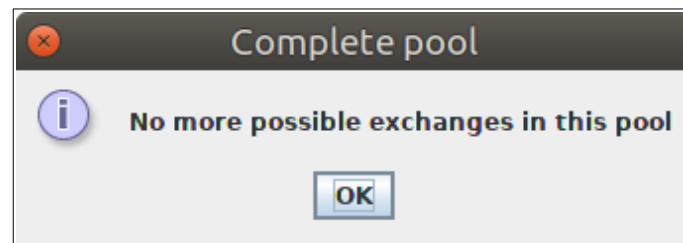


Fig. 15. Avís de que no es poden fer més intercanvis.

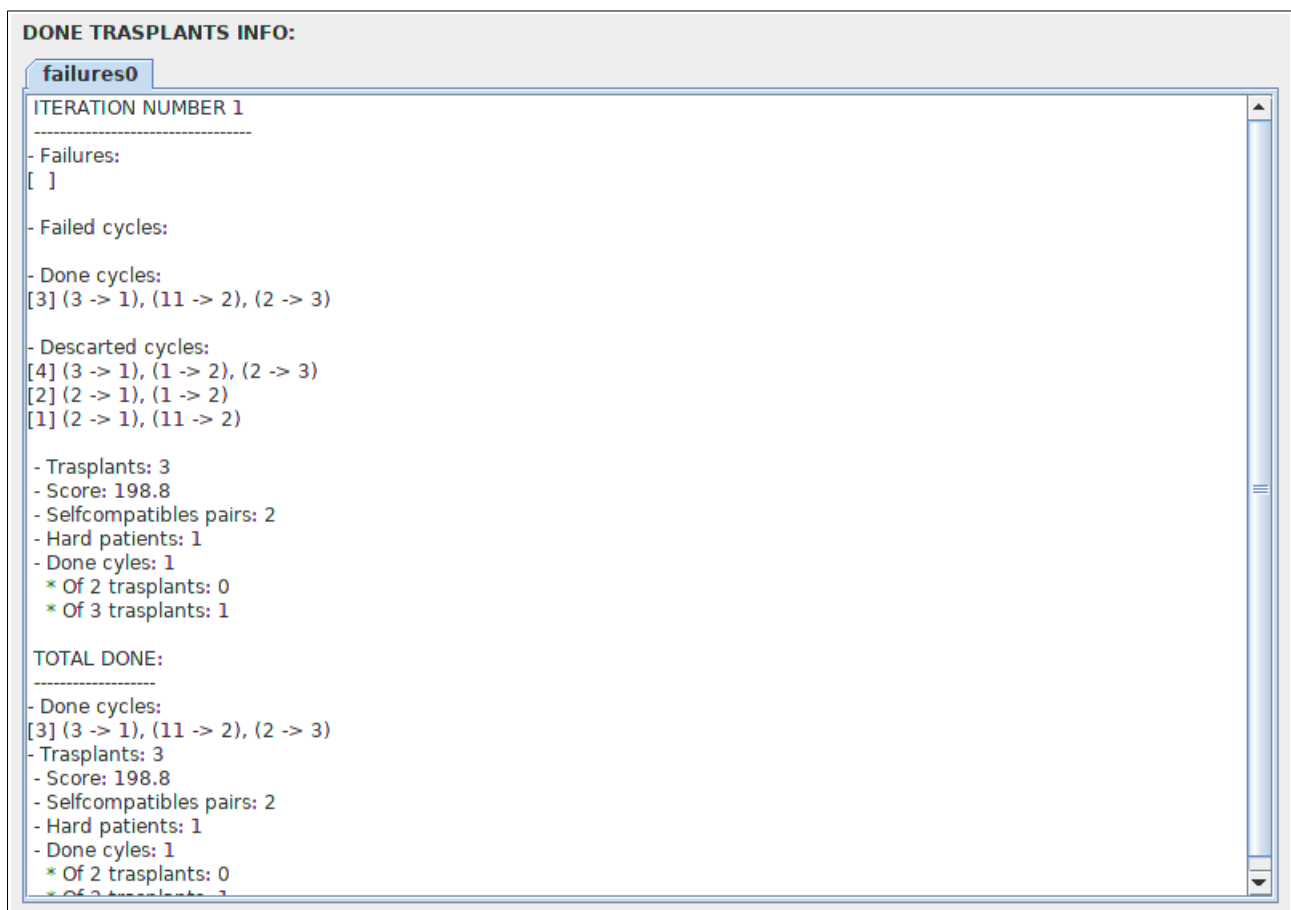


Fig. 16. Tauler de fallades.

14.2.6 Pestanya «Repairs»

Aquesta pestanay permet fer, comparar i veure la informació de reparacions (replanificacions) sobre les fallades ocorregudes.

- i. Tria de les fallades a reparar: només es pot tenir unes fallades triades. Aquestes seran les utilitzades pel programa mentre no les canviïs. Es pot triar clicant sobre el desplegable, per veure totes les disponibles (les creades fins al moment), i triar clicant la que es vol. Després de fer-ho veuràs que és la que es veu.

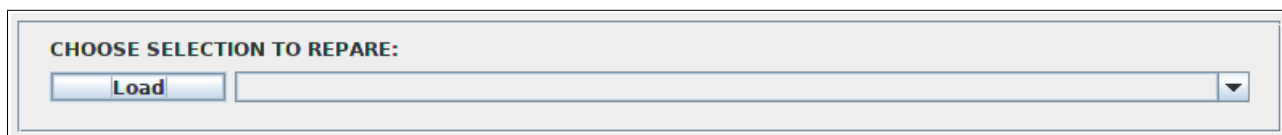


Fig. 17. Tria de les fallades a reparar.

- ii. Elecció de criteris i el seu ordre: hi ha dues llistes, inicialment la de l'esquerra conté tots els criteris i la de la dreta cap. Es tracta de posar a la segona llista tots els criteris que es vulguin utilitzar, i deixar-los ordenats, el primer és el de dalt de tot i així anar baixant. Per moure'ls d'una llista a l'altra hi ha dues maneres:

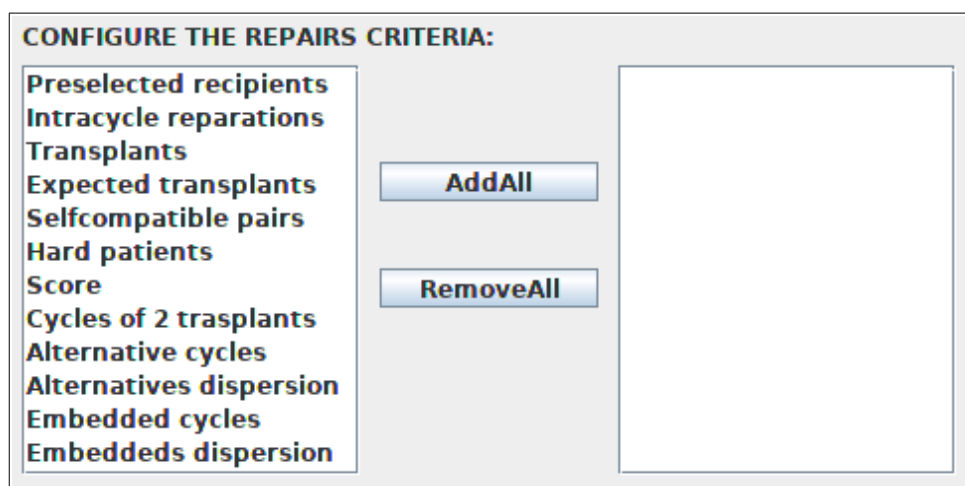


Fig. 18. Llistes d'elecció de criteris de la pestanya «Repairs».

- 1ª. Fent doble clic sobre el criteri que vulguis utilitzar.
- 2ª. Seleccionant tots els que vulguis i clicant el botó «AddAll» per passar-los de la primera llista a la segona o clicant el botó «RemoveAll», per passar-los de la segona llista a la primera.

Per triar-ne l'ordre, manté clicat el criteri que vulguis i arrossega'l fins a la posició desitjada.

- iii. Donar nom a la reparació: és obligat donar-li nom (no pot contenir el caràcter /). Es fa escrivint el nom al requadre blanc de sota les llistes de criteris.

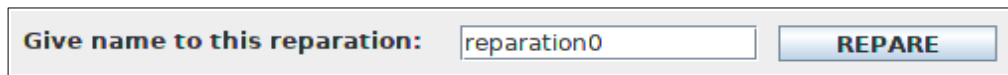
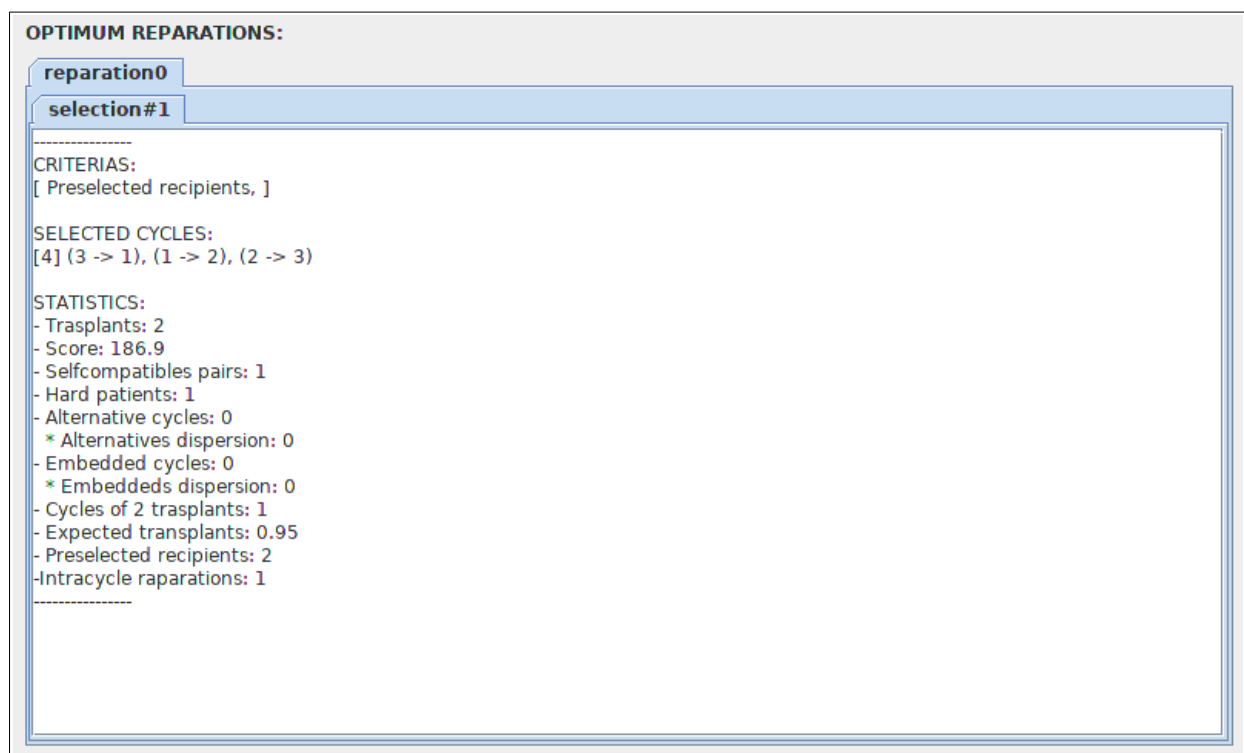


Fig. 19. Tria del nom de la reparació.

- iv. Executar: un cop fets tots els passos anteriors, clica el botó «REPARE» i en poc segons apareixerà al tauler de reparacions, una pestanya amb el nom triat al pas iii. amb les diferent seleccions (replanificacions) proposades (cadascuna una pestanya) sobre les fallades triades al pas i. que maximitzen els criteris triats en l'ordre triat al pas ii.



OPTIMUM REPARATIONS:

reparation0

selection#1

CRITERIAS:
[Preselected recipients,]

SELECTED CYCLES:
[4] (3 -> 1), (1 -> 2), (2 -> 3)

STATISTICS:

- Trasplants: 2
- Score: 186.9
- Selfcompatible pairs: 1
- Hard patients: 1
- Alternative cycles: 0
- * Alternatives dispersion: 0
- Embedded cycles: 0
- * Embeddeds dispersion: 0
- Cycles of 2 trasplants: 1
- Expected transplants: 0.95
- Preselected recipients: 2
- Intracycle raparations: 1

Fig. 20. Tauler de reparacions.

14.2.7 Pestanya «History»

Aquesta pestanya permet veure informació de donants, pacients, compatibilitats i cicles de pools concrets, així com veure l'historial i l'estat actual de planificacions, fallades i replanificacions concretes.

- i. Tria del pool, planificacions, fallades o replanificacions a visualitzar: només es pot tenir-ne un de triat. Aquest sera l'utilitzat pel programa mentre no el canviïs. Es pot triar clicant sobre el desplegable, per veure tots els disponibles (els creats fins al moment), i triar clicant el que es vol. Després de fer-ho veuràs que és la que es veu.

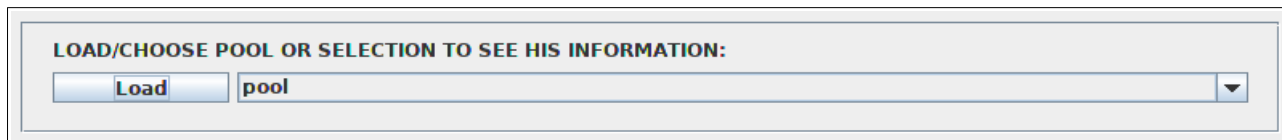


Fig. 21. Tria de pool o selecció a visualitzar.

- Pacients, donants, compatibilitats i cicles: la visualització de la informació d'aquests es realitza al costat esquerra de la pestanya. Estan separats per tipus i cadascú té un desplegable i un requadre amb informació. Es pot triar clicant sobre el desplegable, per veure tots els que té el pool o selecció triats al punt i. i triar clicant el que es vol visualitzar. Després de fer-ho veuràs que és la que es veu.

Recipients

1

- Id: 1
- ABO: 0
- CCAA: 1
- Dialysis months: 0
- Enter pool date: 2011-01-01
- Selfcompatible: false
- Hard patient: true
- Linked Donors: [1, 11]

Donors

1

- Id: 1
- Linked recipient: 1

Compatibilities

1

- Id: 1
- Donor: 3
- Recipient: 1
- Score: 75.0
- Failure probability: 5.0%

Cycles

1

- Id: 1
- Donations: 2
- * [2] 2 -> 1
- * [4] 11 -> 2
- Selfcompatible pairs: 1
- Hard patients: 1
- Score: 86.9

Fig. 22. Visualització de la informació dels elements bàsics del pool.

- Pools i seleccions: la visualització de la informació d'aquests es realitza al costat dret de la pestanya. Es mostra la informació del triat al punt i.

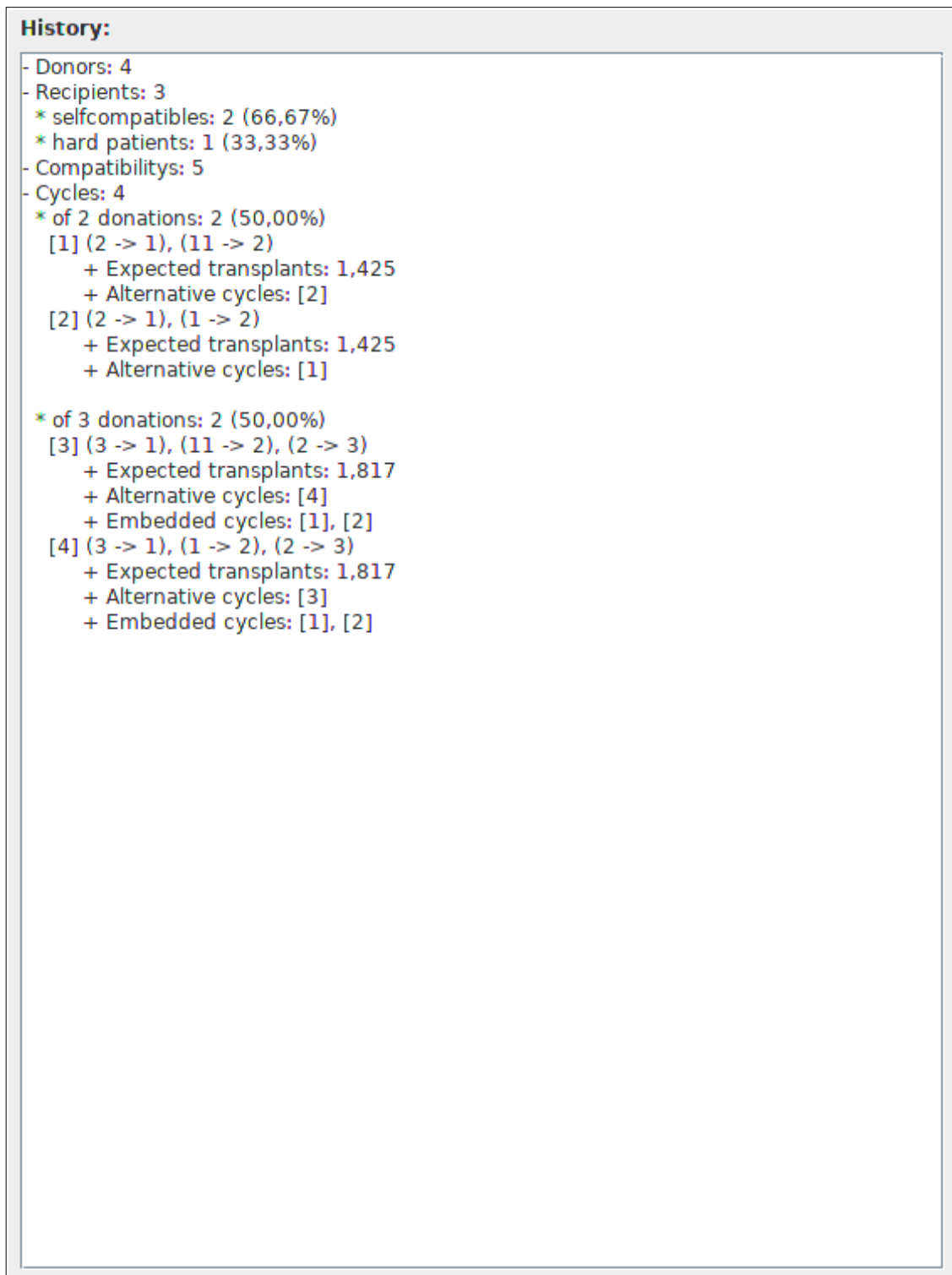


Fig. 23. Visualització d'historials.

14.2.8 Guardat de fitxers

NEPHRO permet guarda informació en format .txt, l'estat actual en que es troben pools, seleccions, fallades i reparacions en format .json, així com els seus historials en format .txt. Es pot fer en qualsevol de les pestanyes on es treballa amb aquests «Pools», «Selections», «Failures» i «Repairs». Concretament es poden guardar a la part inferior esquerra d'aquestes pestanyes.

- i. Tria del pool, planificacions, fallades o replanificacions a guardar: es fa clicant la seva pestanya, del tauler corresponent, és a dir la pestanya on surt la seva informació.
- ii. Tria del tipus de fitxer a guardar: només es pot tenir un tipus triat. Aquest serà l'utilitzat pel programa mentre no el canviïs. Es pot triar clicant sobre el desplegable (veure fig. 24), per veure tots els disponibles, i triar clicant el que es vol. Després de fer-ho veuràs que és el que es veu.

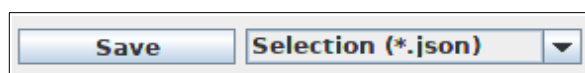


Fig. 24. Guardat de fitxers.

- iii. Tria del nom del fitxer i la carpeta on es guardarà: clicant el botó «Save» (veure fig. 24). S'obrirà una finestra (veure fig. 25) que et permetrà moure't per les carpetes de l'ordinador. Quan estiguis a la carpeta on vols guardar-lo, posa el nom al requadre blanc on diu «File Name:». Un cop fet clica el botó «Save».

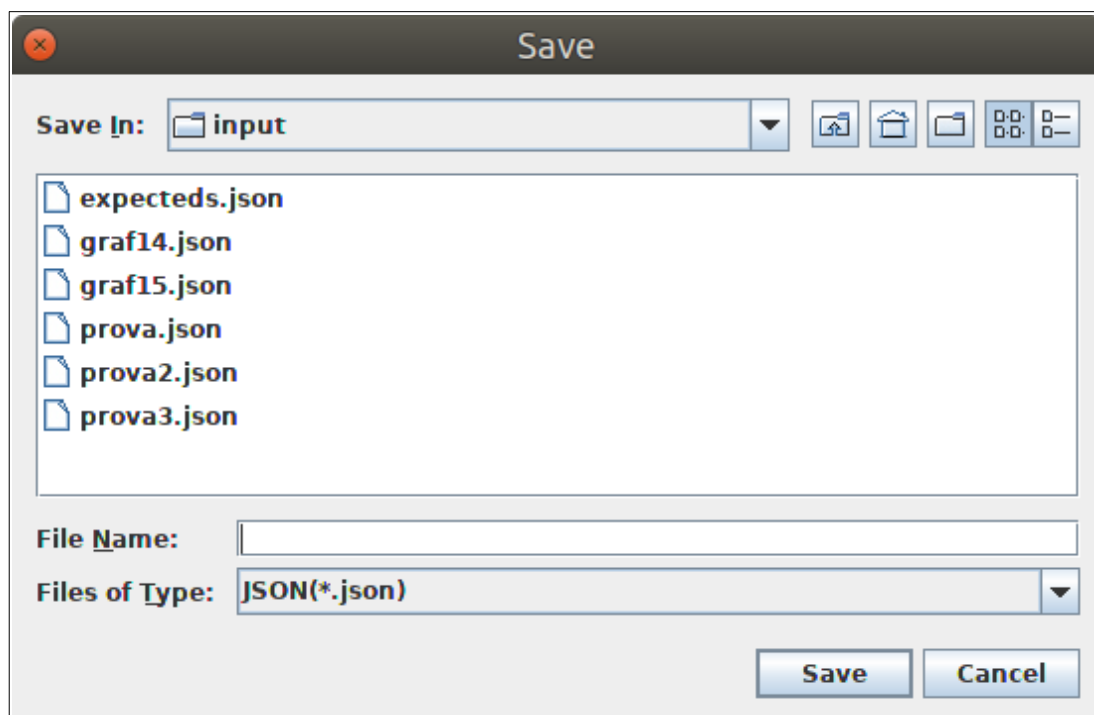


Fig. 25. Finestra per triar la carpeta i el nom.

14.2.9 Carrega de fitxers

NEPHRO permet carregar pools, seleccions, fallades i reparacions desats en format .json per tal de seguir treballant-hi. Es pot fer en qualsevol de les pestanyes «Selections», «Failures», «Repairs» i «History».

Es fa clicant el botó «Load» (veure fig. 8, 12, 17 i 21) a la part superior esquerra de la pestanya. S'obrirà una finestra (veure fig. 5) que et permetrà moure't per les carpetes de l'ordinador. Quan estiguis a la carpeta que conté el fitxer que vols, selecciona'l i clica el botó «Open». Un cop fet veuràs que surt al desplegable corresponent.