



**Universitat Oberta
de Catalunya**

www.uoc.edu

Màster en Programari Lliure

Treball Final de Màster

Projecte de recerca bàsica o aplicada

ESTUDI D'EINES I ALGORISMES DE TIMETABLING UNIVERSITARI A LA UPC

Sergio Pavón Salamanca -Daniel Riera Terrén - Joan Antoni Pastor

1	RESUM	5
2	INTRODUCCIÓ	6
3	TIMETABLING	7
3.1	Definició	7
3.2	Timetabling Educatiu	8
3.3	Timetabling universitari: Conceptes	9
3.3.1	Restriccions.....	12
4	ESTUDI DE ALGORISMES DE TIMETABLING	14
4.1	Anàlisi i resultats	17
4.2	Conclusions	21
5	ESTUDI D'EINES DE TIMETABLING	23
5.1	Paquets de programari lliure	23
5.1.1	UniTime	23
5.1.2	Drools Solver Planner	24
5.1.3	FET (Free Timetabling Software)	24
5.2	Paquets comercials	24
5.2.1	Lantiv Scheduling Studio.....	24
5.2.2	Eventmap Limited	24
5.3	Anàlisi i resultats	25
5.4	Conclusions	28
6	ESTUDI DEL TIMETABLING UNIVERSITARI A LA UPC	29
6.1	La UPC	29
6.2	Conceptes previs	29
6.3	Anàlisi i resultats	30
6.4	Conclusions	39
7	CONCLUSIONS GENERALS	40
7.1	Línies de Futur	40
8	APÈNDIX	41
8.1	Apèndix A: Relació d'algorismes i mètodes	41
8.2	Apèndix B: Referències Bibliogràfiques	44

1 Resum

El procés de construcció taules d'horaris (timetabling) on es distribueixen els diferents **recursos** (aularis, assignatures, professors) complint una sèrie de **restriccions** (solapaments, capacitat dels aularis, restriccions de calendari), intenta resoldre un problema recurrent que es produeix en cada inici de curs als centres universitaris.

El programari relatiu a la solució de problemes timetabling no representa una novetat: existeixen solucions des de fa molts anys, emprant tecnologies, algorismes i tècniques diverses. Amb el creixement l'Open Source, han sorgit moltes referències en quant a la resolució de taules d'horaris. El programari lliure obre una porta molt interessant des de el punt de vista de les Universitats Públiques, ja que ens permet analitzar la implementació dels algorismes de timetabling emprats, ampliar funcionalitats, compartir esforços i recursos.

L'objectiu principal d'aquest treball és reunir informació per tal de **categoritzar eines, algorismes i requisits** que ajudin a la presa de decisions als centres de la UPC en quant a la implantació de un sistema de gestió d'horaris existent o el desenvolupament d'un de nou.

2 Introducció

L'estudi que realitzo està dividit en dues parts:

- **Estudi de eines i algorismes de timetabling¹ universitari:** té com objectiu obtenir una visió general dels algorismes i la obtenció de les funcionalitats comunes de les eines del mercat.
 - **Estratègia a seguir:** Survey
 - **Mètode de generació de la informació:** Documents de la literatura, demos i presentacions dels paquets de software
 - **Anàlisi:** Qualitatiu
- **Estudi del timetabling universitari a la UPC:** té com objectiu l'estudi del procés de elaboració de taules horàries per obtenir els requeriments i restriccions comuns que tenen (o haurien de tenir) els sistemes de timetabling als centres de la UPC.
 - **Estratègia a seguir:** Estudi de Casos
 - **Mètode de generació de les dades:** Enquesta i Entrevistes
 - **Anàlisi:** Quantitatiu per l'enquesta i Qualitatiu per les entrevistes.

Totes dues parts es complementen en el fet de que tenir una visió general dels algorismes (en quins casos s'han de fer servir uns o altres) i de les eines del mercat (glosant les funcionalitats comunes), pot ajudar a entendre com es realitza (o com es pot enfocar) el timetabling als centres de la UPC.

Finalment dels productes obtinguts de cadascuna de les parts es realitzarà un anàlisi que permeti creuar requeriments i funcionalitats, que ajudin a un centre de la UPC a decidir quin sistema pot emprar en el procés de obtenció de taules horàries o bé quines són les funcionalitats,

requeriments i algorismes aplicables per al desenvolupament de un nou sistema.

¹ Programació horària de classes i/o exàmens

3 Timetabling

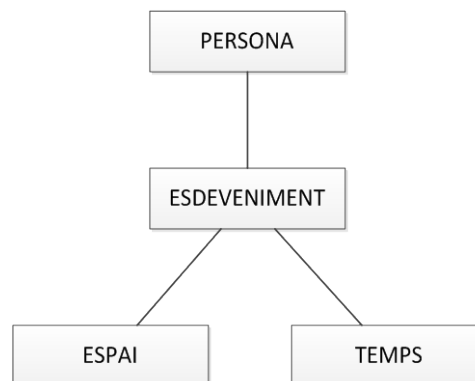
Aquest apartat ens ha de servir per a definir el problema que volem estudiar i descriure els conceptes relacionats que ens serviran de base per els apartats posteriors.

3.1 Definició

En el camp de la Investigació Operativa i de la Intel·ligència Artificial els problemes relatius a la planificació i gestió de taules d'horaris són coneguts com Timetabling Problems (TP) o Automated Timetabling Problems (ATP). Una definició de la mà de **Burke Kingston i Werra** [1] :

"A timetabling problem is a problem with four parameters: T , a finite set of times; R , a finite set of resources; M , a finite set of meetings; and C , a finite set of constraints. The problem is to assign times and resources to the meetings so as to satisfy the constraints as far as possible."

"Un problema d'horaris és un problema amb quatre paràmetres: T , un conjunt finit de espais de temps, R , un conjunt finit de recursos; M , un conjunt finit de reunions², i C , un conjunt finit de restriccions. El problema és l'assignació de espais de temps i recursos a les reunions per tal de satisfer les restriccions en la mesura del possible."



II·lustració 1 Conceptes i les seves relacions per problemes de timetabling en general

Segons **Schaerf** [2] els problemes de Timetabling, en general, es poden veure com variacions de:

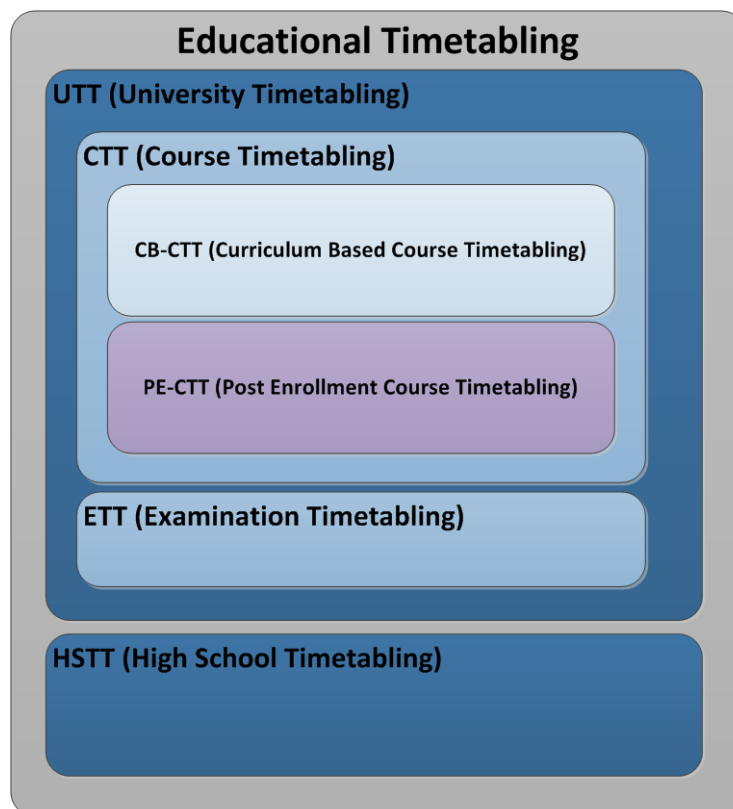
- **Problemes de cerca (Search Problems)**: busquen una taula de horaris que solucioni el problema complint les restriccions importants.
- **Problemes de optimització (Optimization Problems)**: intenten optimitzar la taula de horaris per a complir de forma òptima les restriccions.

² En el cas del Timetabling universitari un *meeting* es pot veure com un esdeveniment de formació: la impartició de una classe (també anomenada lectura) o examen.

La complexitat del tractament d'aquest problema resideix en trobar **solucions viables (feasible solutions)** que satisfacin les restriccions de forma òptima. Aquests són problemes considerats computacionalment **NP-Complets** [3], com ho constaten la gran majoria de articles de la literatura (la tesi de **Willemen** 2002 [4] és un dels exemples més complets) ja que no hi ha una forma de arribar a la solució en un temps polinòmic. És per això que s'han anat creant algorismes que fan servir mètodes, tècniques i heurístics per tal de trobar solucions viables i que veurem en un apartat posterior.

3.2 Timetabling Educatiu

Els centres educatius s'enfronten a cada inici de curs a la feixuga tasca construir taules d'horaris. Els problemes de Timetabling per educatius han estat tradicionalment classificats segons aquestes categories principals[2], [5] que es mostren en el diagrama:



II-lustració 2 Classificació dels problemes de timetabling educatiu

Nosaltres ens centrarem en el timetabling per a universitats, anomenat **University Timetabling o UTT**. La diferència principal amb el timetabling per centres de secundària (**High School Timetabling o HSTT**) radica en que els estudiants han de tenir ocupades totes les hores del dia, i han de ser supervisats.

Als centres universitaris el problema es defineix com la distribució en el **temps** dels diferents **recursos** (espai, professors, alumnes) assignant-los a **lectures/exàmens** complint una sèrie de **restriccions** (evitar solapaments, capacitat dels aularis, limitacions de calendari, plans d'estudi). Aquesta feina implica tant a gestors d'horaris, com a caps de departaments docents, professors i la pròpia direcció acadèmica.

Els problemes de Timetabling per universitats han estat tradicionalment classificats segons aquestes categories principals[2], [5]:

- **Problemes d'horaris de lectures (Course Timetabling problems CTT):** Les lectures (o classes tradicionals) es programen abans que els estudiants es matriculin, de forma que es permet (tot i que no es desitja) que hi hagi solapament de estudiants. Una lectura no es pot donar en dues ubicacions alhora, ni dues lectures poden compartir ubicació. Entre aquests problemes trobem dos subtipus diferents depenent del moment de la matrícula [6], [7]:
 - **Basats en el pla d'estudis (Curriculum Based Course Timetabling CB-CTT):** En aquest cas les taules horàries es construeixen abans que els estudiants es matriculin en base als plans d'estudi (normalment dividits en nivells temporals o conjunts d'assignatures anomenats mòduls) publicats per els centres. Les assignatures d'un mateix nivell o mòdul no poden solapar-se.
 - **Basats en la matrícula (Post Enrollment Course Timetabling PE-CTT):** En aquests casos les taules horàries es confeccionen segons les previsions de matrícula dels estudiants i per tant es confeccionen post matrícula. Els alumnes no han de tenir solapaments en les assignatures matriculades.
- **Problemes d'horaris d'exàmens (Examination Timetabling problems ETT):** Els exàmens han de ser programats sense que un estudiant hagi d'assistir a dos exàmens alhora. Un examen es pot programar en dues ubicacions alhora, i dos exàmens poden compartir ubicació.

Sobre aquests problemes existeixen variacions que normalment acosten la definició a la realitat, però augmenten la complexitat inicial. Així tenim el **particionament** dels estudiants en grups (**Sectioning Timetabling** [8]), la **distribució** de la resolució del problema per unitats organitzatives com poden ser els departaments (**Distributed Timetabling** [9]–[11]), o bé la possibilitat creació interactiva de taules de horaris (**Interactive Timetabling** [12]–[14]).

3.3 Timetabling universitari: Conceptes

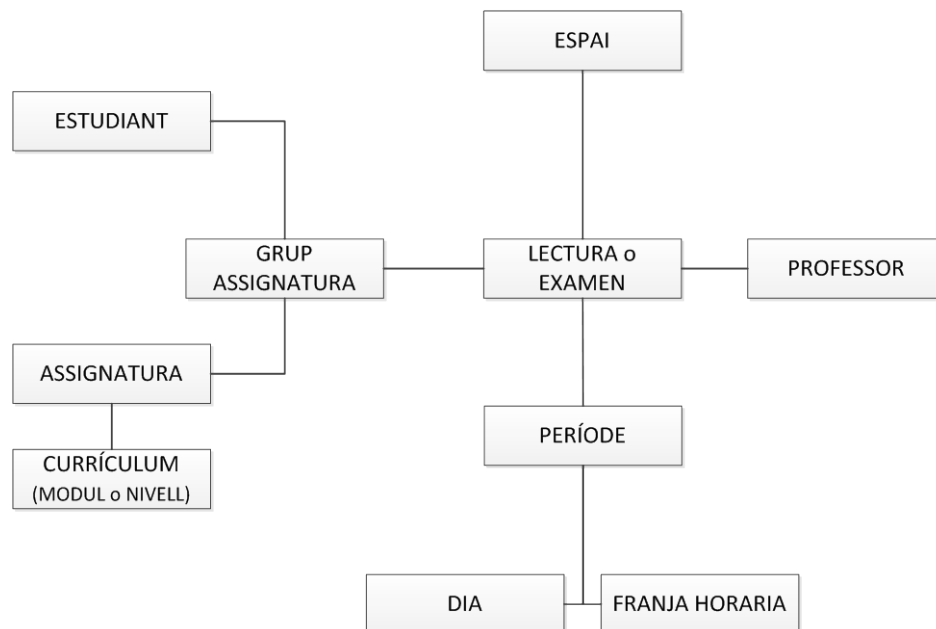
Una vegada definits i classificats els problemes de Timetabling ens centrarem en els problemes de gestió d'horaris per universitats (CTT), ja que aquest és el tema d'objecte d'aquest treball³. Primer descriurem els diferents conceptes que hi apareixen a la literatura per tal d'entrar en un anàlisi de les restriccions que es solen definir.

Els conceptes que apareixen als problemes UTT són [4], [15]:

- Conceptes Temporals
 - **Dia:** Dies en els que es poden assignar lectures o exàmens. En el problema CTT normalment són els dies en base a la setmana i solen ser 5 (els laborables) o 6.
 - **Franja Horària:** Els dies es divideixen en franges horàries com 8:00 a 9:00
 - **Període:** La combinació de un Dia i una Franja Horària
 - Les lectures de una assignatura durant la setmana poden seguir **patrons temporals**, per exemple 3 hores dilluns i 1 hora dimecres. Aquests permeten la distribució de l'assignatura en el temps i deixen temps als estudiants per assimilar la matèria.
- Conceptes relacionats amb persones:
 - **Estudiants:** alumnes del centre universitari enrolats en un programa d'estudis
 - **Professors:** Personal docent i investigador

³ Tot i que en l'anàlisi de algorismes i restriccions em estudiat la literatura referent al conjunt de problemes de timetabling universitari (tant lectures com exàmens), nosaltres em intentat enfocar aquest estudi més sobre els problemes CTT que no pas el ETT, i especialment els problemes CB-CTT ja que són el principal problema dels gestors d'horaris a la UPC.

- Conceptes Espacials
 - **Espais**: Ubicacions de docència disponibles que poden ser Aules, Laboratoris amb equipaments i capacitats específics. És important conèixer en quins edificis o campus hi són per tal de reduir la mobilitat de persones.
- Conceptes Acadèmics
 - **Curriculum (Programa d'estudis)**: Conjunt de assignatures establertes al pla d'estudis del programa que s'està cursant. Normalment s'agrupen mitjançant **mòduls/blocs curriculars** o **nivells** (per anys/quadrimestres)
 - **Assignatures(Cursos)**: Les unitats docents o cursos que s'imparteixen sobre un camp de coneixement concret dins del Programa.
 - **Grup**: Quan es particionen(sectioning) les assignatures en grups més petits deguts a problemes de capacitat.
- **Lectures/Exàmens**: L'assignatura es divideix en diferents esdeveniments educatius que són impartits per un professor, en un espai i durant un període determinat dins de la setmana.



II-lustració 3 Diagrama de conceptes i relacions entre ells per problemes CTT i ETT. Estan afegits conceptes de particionament (Grups) i Currículum

Un darrer concepte important és el d'**instància**. Una instància és una representació d'un problema concret de timetabling on es configuren els diferents conceptes indicats anteriorment i els seus valors (anomenats paràmetres inicials), així com les **restriccions** que han de complir per trobar una solució. Com exemples de paràmetres inicials d'una instància tenim la capacitat de les aules, el número d' alumnes, número de assignatures, número de períodes, número de professors.

Cada instància varia en el seus paràmetres i restriccions de forma que els problemes poden ser, a priori, mes o menys senzills de resoldre de forma que es poden emprar diverses estratègies en funció de les característiques d'una instància concreta.

Darrerament les instàncies es representen mitjançant **fitxers XML** [15], [16] que permeten validar-les i estandaritzar la definició de problemes. Existeixen diversos **repositoris d'instàncies** [17] que permeten testejar algorismes utilitzant aquestes configuracions. Normalment aquestes

instàncies tenen més de una solució possible i ha estat emprades en diferents concursos com el **International Timetabling Competition (ITC)** [15]

```

Name: Instance_name
Courses: Number of courses
Rooms: Number of rooms
Days: Number of days
Periods per day: Number of periods
Curricula: Number of curricula
Constraints: Number of constraints
COURSES:
<CourseID> <Teacher> <#Lectures> <MinWorkingDays>
<#Students>
<CourseID> <Teacher> <#Lectures> <MinWorkingDays>
<#Students>
<CourseID> <Teacher> <#Lectures> <MinWorkingDays>
<#Students>
. . .
ROOMS:
<RoomID> <Capacity>
<RoomID> <Capacity>
<RoomID> <Capacity>
. . .
CURRICULA:
<CurriculumID> <#Courses> <CourseID>. . . <CourseID>
<CurriculumID> <#Courses> <CourseID>. . . <CourseID>
<CurriculumID> <#Courses> <CourseID>. . . <CourseID>
. . .
UNAVAILABILITY_CONSTRAINTS:
<CourseID> <Day> <Time>
<CourseID> <Day> <Time>
<CourseID> <Day> <Time>
. . .

```

II·lustració 4 Format de Fitxer XML per a instància per ITC 2007 (Gasparo et al 2007)

3.3.1 Restriccions

Les **restriccions** són les **condicions** que han de complir les solucions als nostres problemes d'horaris. La definició d'aquestes restriccions depèn en gran mesura dels requeriments que cada centre universitari imposa segons les seves directrius i recursos. Aquesta variabilitat fa difícil una definició genèrica del problema així com la tria de un algorisme per trobar-ne solucions.

Burke et al [18] ja va veure al seu estudi sobre les universitats de Gran Bretanya que cada centre tenia uns requeriments diferents. Tot i així aquest estudi ha analitzat la literatura per trobar punts en comú en la definició de restriccions i ajudar a classificar-les.

La literatura de Timetabling [2], [5], [19] normalment categoritza les restriccions en dos grans grups:

- **Restriccions Fortes (Hard Constraints)**: Aquestes restriccions són crítiques i no poden ser violades sota cap concepte. Per exemple: un professor no pot fer dos lectures alhora, un estudiant pot tenir dos exàmens alhora (son exemples de col·lisions).
- **Restriccions Febles (Soft Constraints)**: Aquestes restriccions són **desitjables**, però no són crítiques i per tan es poden violar. Per exemple: un alumne no pot tenir més de dos exàmens en un mateix dia. Per exemple: és desitjable que els alumnes no tinguin hores al final del dia.

Les solucions que no violen cap restricció forta són anomenades **solucions viables (feasible solutions)**. Les solucions viables que compleixen més restriccions febles són anomenades **solucions òptimes (optimal solutions)**.

De fet seria impossible trobar solucions viables als problemes complint totes les restriccions febles. Segons **Petrovic i Burke** [19] (2004):

"la qualitat de una solució viable pot ser avaluada en quant a com de bé es resolen les restriccions febles".

3.3.1.1 Classificació de restriccions

Una relació dels principals tipus de restriccions la trobem a **Burke et al** [5] (1997). Analitzant aquesta i altres tipologies de [4], [6], [20]–[23] hem creat la següent classificació:

- **CA Calendar Constraint:** no es poden programar lectures/exàmens fora dels períodes establerts al calendari. Sempre és una restricció forta.
- **RC Resource Conflict:** cap recurs (espai, professor, assignatura, alumne) no pot ser programat més de una vegada en un període.
 - **CC Curriculum Conflict:** les lectures de dues assignatures diferents en un mateix mòdul curricular no poden programar-se en el mateix període. És una restricció forta per als problemes CB-CTT.
- **COM Completeness:** Totes les lectures/exàmens han de ser programades/ts. Sol ser una restricció forta.
- **RA Resource Assignment:** Un recurs imposa una restricció sobre un altre recurs, o sobre un esdeveniment (classe/examen). Per exemple, un professor prefereix ensenyar en una sala en particular (per la raó que sigui), o un examen hauria de tenir lloc en un edifici en particular. Poden ser restriccions fortes o dèbils.
 - **RSU Room Suitability:** és un subtipus que ens indica alguns esdeveniments només es poden donar en determinats espais per les seves característiques. Solen ser restriccions fortes.
- **TA Time Assignment:** Un recurs ha de ser assignat a un període de temps. Aquesta restricció pot ser utilitzada per especificar els dies en que un professor no està disponible, o per a pre-assignar períodes de temps a esdeveniments (classes/exàmens). Solen ser restriccions fortes.
 - **TRA Teacher/Room Availability:** indiquen si un professor o espai estan o no disponibles. Solen ser restriccions fortes.
- **TCBM Time Constraints between Meetings:** Aquestes restriccions ens permeten limitar que una classe comenci abans que una altra o que dos exàmens han de realitzar-se simultàniament. Poden ser restriccions fortes o dèbils.
- **MS Meeting Spread:** Les classes i exàmens han de distribuir-se en el temps raonablement. Un estudiant no hauria de tenir més de un examen en un dia, o tenir lectures distanciades amb temps morts. Solen ser restriccions dèbils.
 - **NEB No End/Beginning Day:** No programar al principi o final de un dia. És per evitar un alumne amb una lectura/examen aïllada al final o principi del dia. És una restricció dèbil
 - **CCM Curriculum Compactness:** Les lectures de assignatures de un mòdul curricular han d'anar seguides. És una restricció dèbil.
 - **BDW Bounds on Daily Workload:** Estableix els límits superior i inferior de treball diari per estudiants i professors. És una restricció dèbil.
 - **BNWD Bounds on Number of Working Days:** Estableix els límits superior i inferior de dies de treball per a estudiants i professors. És una restricció dèbil.
 - **BTIT Bounds on Total Idle Time:** Estableix els límits superior i inferior de temps morts que pot tenir un estudiant o professor. És una restricció dèbil.
- **MC Meeting Coherence:** Pensades per a produir horaris més organitzats i convenients, sovint entren en conflicte amb el tipus de restriccions anterior (Meeting Spread). Per exemple un professor pot preferir tenir totes les seves classes en dos dies, tenint tres dies lliures de classes. Solen ser restriccions dèbils.
- **RCP RoomCapacity:** El nombre d'estudiants en un espai no ha d'excedir la seva capacitat. És una restricció forta.
- **CON Continuity:** Restriccions que tenen com a objectiu que algunes característiques d'un horari siguin constants o predictibles. Per exemple: les classes de un mateix curs haurien de ser programades en el mateix espai (**RST RoomStability**) o en el mateix període del dia. Solen ser restriccions dèbils.

4 Estudi de algorismes de timetabling

La recerca en els camps de la Investigació Operativa ens ha aportat diferents enfocaments per a resoldre problemes de Timetabling Universitari mitjançant diversos mètodes, tècniques i algorismes.

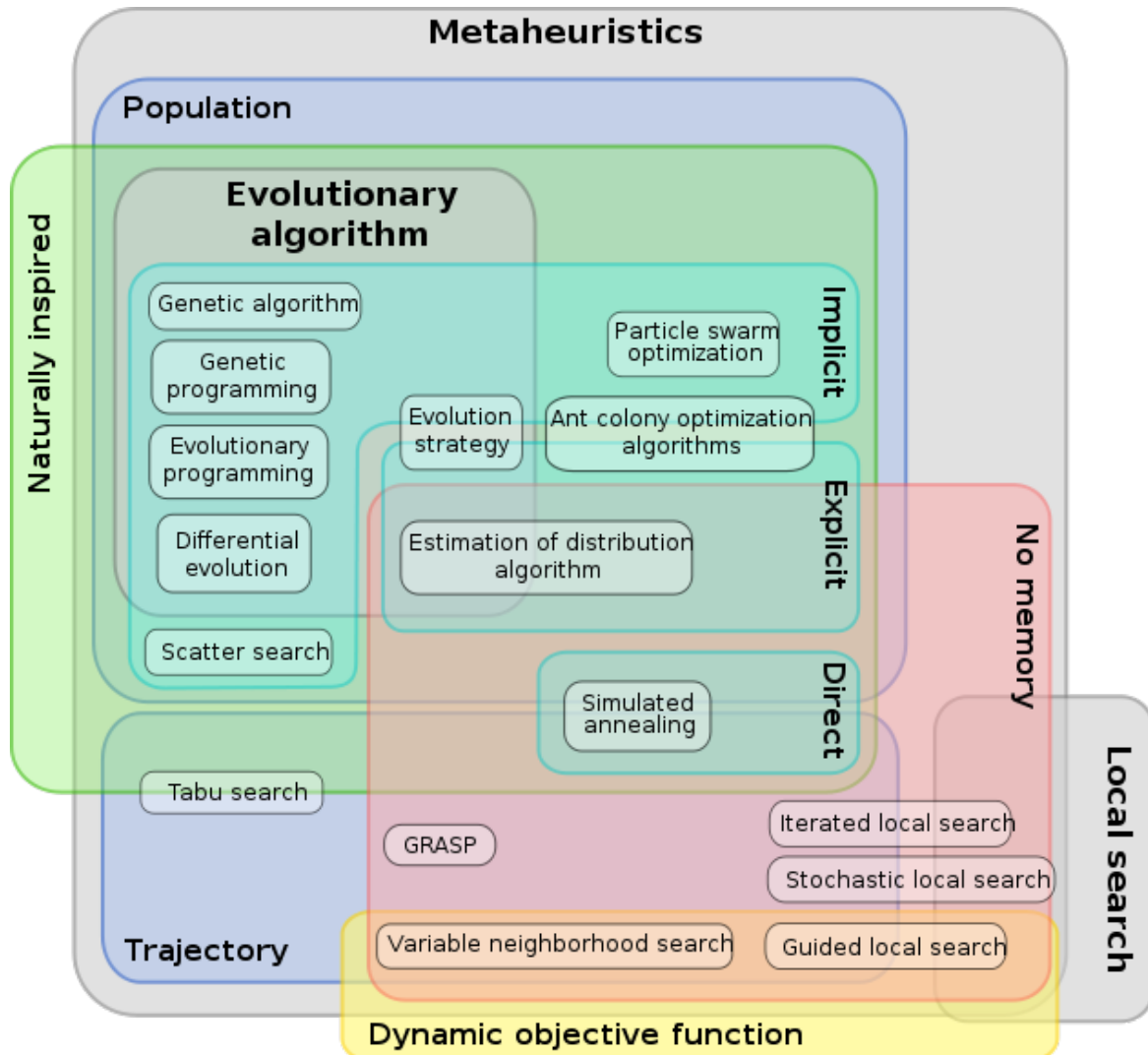
En els darrers anys la feina de investigació en aquest àrea s'ha incrementat molt. Com a exemple tenim l'aparició de **WATT**[24] un grup específic en la recerca sobre problemes de Automated Timetabling. Aquest grup organitza bianualment la conferència **PATAT**[25], font de molts dels articles de recerca .

Degut a la gran quantitat de enfocaments, diferents autors els han classificat en els seus estudis de la literatura. Així **Carter i Laporte** [26] ofereixen una classificació dels enfocaments apareguts del 1986 fins al 1996. Mes endavant **Petrovic i Bruke** [19], [27] amplien aquesta classificació. Al 2009 **R.Qu et al** [28], ofereix una nova classificació en el seu treball de estat de l'art del Timetabling. Després d'analitzar tot aquest treball classifiquem els enfocaments basats en:

- **Tècniques matemàtiques(Math)**: normalment es fa servir una formulació del problema basada en **Integer(IP)** o **Linear Programming (LP)**, on el problema es formula com a notacions basades en enters. Per la seva senzillesa va ser de les primeres tècniques, tot i que recentment s'han tornat a fer estudis com el de **Helber et al.** [29] al 2007.
- **Tècniques seqüencials basades en grafs(Graph)**: Aquestes tècniques es basen en modelar el problema com a un graf[30], on els esdeveniments són els nodes i les arestes representen restriccions. S'empren heurístics per determinar quins són els millors camins. Al 1994 **Burke** [31] desenvolupa un prototipus fent servir tècniques de coloració de grafs, demostrant que es pot fer servir aquesta tècnica per problemes de Timetabling. Al 2004 **Burke i Newall** [32] estableixen una estratègia de ordenació dels heurístics. Altres tècniques com la lògica difusa [33] han estat emprades per la ordenació dels heurístics.
- **Tècniques basades en restriccions (Constraint)**: Les tècniques **CLP** (Constraint Logic Programing) sobre **CSP** (Constraint Satisfaction Problems) permeten expressar de forma molt senzilla les restriccions dels problemes de Timetabling. Molts dels treballs [34] han emprat llenguatges (com PROLOG) per expressar les seves restriccions. Entre les tècniques CLP més habituals trobem **backtracking** , **consistency technique**, **branch & bound**.
- **Meta-heurístics**: Aquestes tècniques es basen en cercar solucions òptimes dintre de un espai de cerca. Un conjunt de heurístics serveixen per a nodrir una funció que indica la solució òptima. Al 2003 **Rossi-Doria et al.** [35] fa una comparació dels diferents meta-heurístics.
 - **Tècniques basades en cerca local(Local Search, LS)**: Les tècniques basades en cerca local resolen els problemes mitjançant la cerca entre un conjunt de resultats. Una funció s'encarrega de determinar quina és la millor solució. Els treballs de **Di Gaspero** [36] i **Abdulah i Burke** [37] estudien aquestes tècniques en el món del Timetabling acadèmic.
 - **Tabú Search(TS)**: La cerca es basa en marcar quines són les millors solucions , en base a que una següent iteració amb pitjors resultats es marcada com a Tabú (i no es torna a visitar).
 - **Simulated Annealing(SA)**: Es basa en la simulació del procés de l'enduriment per refredament[38]. La cerca comença de forma general i una variable de Temperatura va baixant descartant els pitjors resultats segons una funció probabilística. El procés acaba quan la Temperatura arriba a 0 (frozen State). **Burke et al** [39] van estudiar una variació de l'algoritme al 2004 anomenada Deluge Algorithm[40].

- **Algorismes de cerca de veïns (VNS):** Anomenats per la literatura Variable Neighbourhood Search són algorismes que permeten ampliar l'espai de cerca a base de petits canvis en les solucions parcials. **Abdulah et al.** [41] al 2005 fa un estudi interessant sobre aquests algorismes en problemes CTT, així com **Burke et al.** [42] al 2010 creant algorismes híbrids que exploten l'espai de cerca de solucions que dona VNS.
- **Algorismes Great Deluge (GD):** Algorismes que simulen la tria de millors solucions en base a un factor de temps límit. Darrerament s'han emprat amb algorismes VNS (**Shaker i Abdullah** [43] 2009).
- **Algorismes GRASP:** Combinació d'un algorisme voraç sobre l'espai de cerca, en conjunt amb un algorisme aleatori i una llista de candidats a solució. Va ser estudiat per **Casey y Thompson** [44] al 2003 i per **Burke et al.**[45] al 2009 sobre problemes ETT.
- **Algorismes de cerca local iterativa(ILS):** és un algorisme que cerca trajectòries òptimes en l'espai de cerca, sense modificar les solucions parcials. Al 2010 **Burke et al.** [46] el compara amb diferents hyper-heurístics.
- **Algorismes de Hill Climbing(HC):** Intenta cercar una solució fent petits canvis en la solució actual, i intenta seguir fins que no hi hagin millores. **Ross i Corne** [47] al 1995 fan un estudi de algorismes de Hill Climbing combinats amb altres mètodes.
 - **Algorismes de basats en poblacions**
 - **Algorismes Genètics(AG):** La idea és simular els mecanismes genètics, codificant les solucions com a cromosomes. La idea és anar iterant per aconseguir una evolució de la solució inicial que sigui òptima.
 - **Algorismes Mimètics(MM):** Estenen els algorismes genètics creant híbrids amb algorismes altres algorismes basats en cerca local. Com a extensions dels anteriors les solucions poden anar millorant durant el seu cicle de vida i no en generacions posteriors. **Hart et al** (2004) [48] mostren els avenços en algorismes mimètics.
 - **Algorismes Formiga(ACO):** Simulen el procés per que les formigues cerquen el camí més curt per arribar al menjar. **Socha et al** [49] revisen al seu treball l'estat de l'art d'aquests algorismes emprats al Timetabling acadèmic.
- **Tècniques multi-criteri(MultiC):** L'enfocament és veure els heurístics que valoren les restriccions com a un vector i no com la suma de valors. Al 2001 **Burke et al.** [50] van desenvolupar un sistema que feia servir 9 criteris per arribar a una solució. Al 2009 **Geiger** [7] fa servir un sistema basat en algorismes evolutius (genètics, mimètics).
- **Hyper-heurístics(Hyper):** en realitat és la combinació de diferents tècniques i meta-heurístics per tal de arribar a solucionar problemes generalistes. **Kendall i Hussin** al 2004 fan una adaptació basada en Tabú Search[51]. Al 2007 **Burke et al** [52] fan una aproximació fent servir coma a base grafs colorejats. Una altra aproximació la dona **Burke et al.**[53], [54] al fer servir una tècnica basada en el coneixement anomenada **Case-based reasoning** i que fa servir com a espai de cerca solucions prèvies.
- **Tècniques de descomposició/clustering (Cluster/Decomp):** La idea és trencar el problema en sub-problemes més senzills per després fusionar-los. Al 1983 **Carter** [55] va fer una primera aproximació a les tècniques de descomposició, que en els darrers anys ha tingut continuïtat en els recents estudis de **Qu. et al.** [56], [57] i **Burke et al.**[58]. En aquesta categoria entren els algorismes que decideixen dividir el problema en agents que s'encarreguen de tractar parts del problema. Entre d'altres, al 2006 **Oprea** [59] realitza un estudi de múltiples agents en timetabling. El 2011 **Wangmaeteekul** [11] realitza un estudi de múltiples agents distribuïts per resoldre CTT.

La següent imatge mostra les diferents famílies de meta-heurístics. Destacar que no tots els meta-heurístics que es mostren s'han fet servir encara sobre problemes de timetabling.



Il·lustració 5 Diagrama on apareixen les diferents famílies de meta-heurístics. Font: Wikipedia

Els metaheurístics i les seves combinacions són les tècniques més emprades en el Timetabling Universitari, ja que segons la web **Metaheuristic Network**[60] "poden ser vistos com un framework algorítmic general que pot ser aplicat a diferents problemes de optimització amb relativament pocs canvis per adaptar-los a problemes específics". A més aquests es poden combinar per crear algorismes que en diverses fases poden arribar a solucions òptimes.

Així un estudi de **Lewis** [61] (2008) classifica els algorismes de timetabling universitari segons la forma en la que aquests ataquen les restriccions. Les categories són:

1. **Algorismes d'optimització en 1 etapa(AO1E)**: intenten arribar a una solució que compleixi les condicions fortes i dèbils simultàniament. Normalment fan servir una funció de pes per les restriccions, ponderant amb més pes les restriccions fortes que les restriccions dèbils i afegint un factor de prioritats entre els subconjunts. Això els fa molt fàcils de implementar i a més tenint una única funció objectiu ens permet aplicar gairebé qualsevol tècnica d'optimització.

Avantatges: Solen ser senzills i arribar amb facilitat a resultats viables. Funcionen molt bé amb espais de solucions grans

Inconvenients: Amb espais de solucions grans poden arribar a ser molt costosos.

2. **Algorismes d'optimització en 2 etapes(AO2E):** on les restriccions dèbils s'intenten satisfer només si s'ha arribat a una solució viable (que satisfà totes les restriccions fortes). Així tenim una fase de **construcció** per obtenir resultats viables que no violen cap restricció forta i després tenim una fase de **optimització** per tal de minimitzar el número de restriccions dèbils violades. La fase de construcció sol tenir algorismes de cerca basats en grafs i la fase de optimització fa servir meta-heurístics. Per tal de no passar per solucions no viables a la fase de optimització s'utilitzen tècniques basades en cerca de solucions veïnes viables (neighbourhood). Les més utilitzada s'anomena **Kempe Chain Neighbourhood** que sol donar els millors resultats.

Avantatges: Permeten la construcció de solucions viables i es centren en cercar solucions òptimes que violin el mínim número de restriccions dèbils.

Inconvenients: Si l'espai de solucions viables és petit, aquest tipus d'algorismes en dues fases es presenta com poc eficient.

3. **Algorismes que permeten relaxacions(ARELAX):** les restriccions fortes es compleixen gràcies a la relaxació d'alguna característica del problema, per tal de satisfer les restriccions dèbils, tenint en compte que després s'hauran d'eliminar relaxacions. Com a exemple tenim el fet d'afegir més franges horàries o bé deixar esdeveniments que no poden col·locar-se de forma viable per després de l'anàlisi de restriccions dèbils.

Avantatges: Obren un gran ventall de combinacions i poden trobar solucions òptimes on no es podien trobar sense relaxar restriccions fortes.

Inconvenients: Poden ser costosos o ineficients si l'espai de solucions inicial és molt gran.

4.1 Anàlisi i resultats

Diferents treballs com els **Rossi-Doria et al.** [13] i **Chiarandini et al.** [34] han estudiat com mesurar els resultats obtinguts per diferents meta-heurístics i altres tècniques per tal d'ajudar en la tria dels mateixos per resoldre problemes. Aquest treballs són força interessants tot i que la majoria revelen la comparació es fa difícil, ja que és molt complex arribar a una formulació que abasti tots els casos (**Bonutti et al.** [23]). Això és degut a que cada institució i departament defineix el problema segons les seves regles i característiques particulars. Estudis posteriors han intentat definir rànquings on situar els heurístics i algorismes per saber quins poden donar millors resultats com **Bonutti et al.**[23] (2010).

Com vam comentar a la introducció, la **International Timetabling Competition(ITC)** es va crear amb l'objectiu de crear un marc comú per a definir formulacions i instàncies genèriques a problemes de timetabling com CTT (**McCollum et al.** [63]). La competició en les seves 3 edicions ha ajudat a la definició i formulació dels problemes, i els algorismes presentats han ajudat molt en la recerca en el camp del Timetabling Universitari i les seves aplicacions.

A banda de la categorització de mètodes en base a la literatura mostrat en l'apartat anterior, a l'**apèndix A** he creat una taula que relaciona els principals articles que plantegen algorismes de timetabling universitari, els anys de publicació i els mètodes que fan servir. Queda clar que els mètodes més emprats són els meta-heurístics, i entre ells els algorismes genètics, simulated annealing i tabú search.

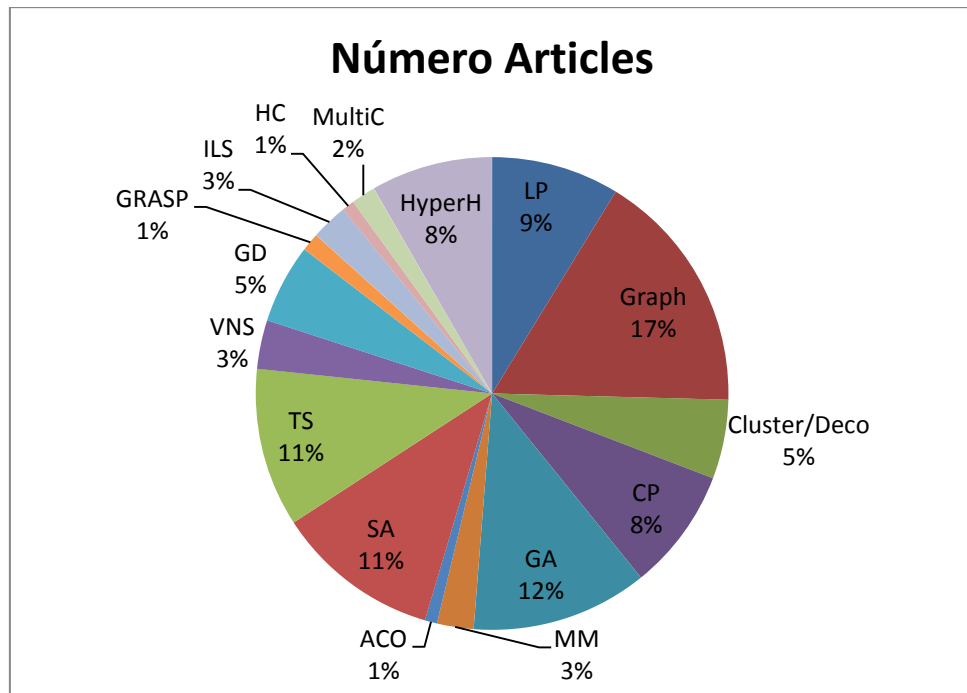
La següent taula mostra els principals heurístics en relació a la categorització de **Lewis** [61] citada abans:

Taula 1 Algorismes que fan servir Meta-heurístics classificats per el tipus de construcció: AO1E (1 Etapa) AO2E (2 etapes), ARELAX(relaxació de restriccions)

Ref	Any	Tipus	Meta-heurístic				
			GA	ACO	SA	TS	ILS
[64]	1990	AO1E	•				
[65]	1991	AO1E	•				
[66]	1998	AO1E	•				
[20]	1995	AO1E	•				
[67]	2001	AO1E	•				
[68]	1991	AO1E			•		
[69]	1993	AO1E			•		
[70]	1998	AO1E			•		
[71]	1994	AO1E				•	
[72]	1991	AO1E				•	
[73]	1996	AO1E				•	
[74]	2000	AO1E				•	
[75]	1998	AO2E(1-graph)			2		
[44]	2003	AO2E(1-grasp)			2		
[76]	2005	AO2E(1-math)				2	
[49]	2003	AO2E		1			2
[77]	2002	AO2E		1			2
[62]	2006	AO2E			2	1	1
[78]	2003	ARELAX			•		
[79]	1998	ARELAX	•				

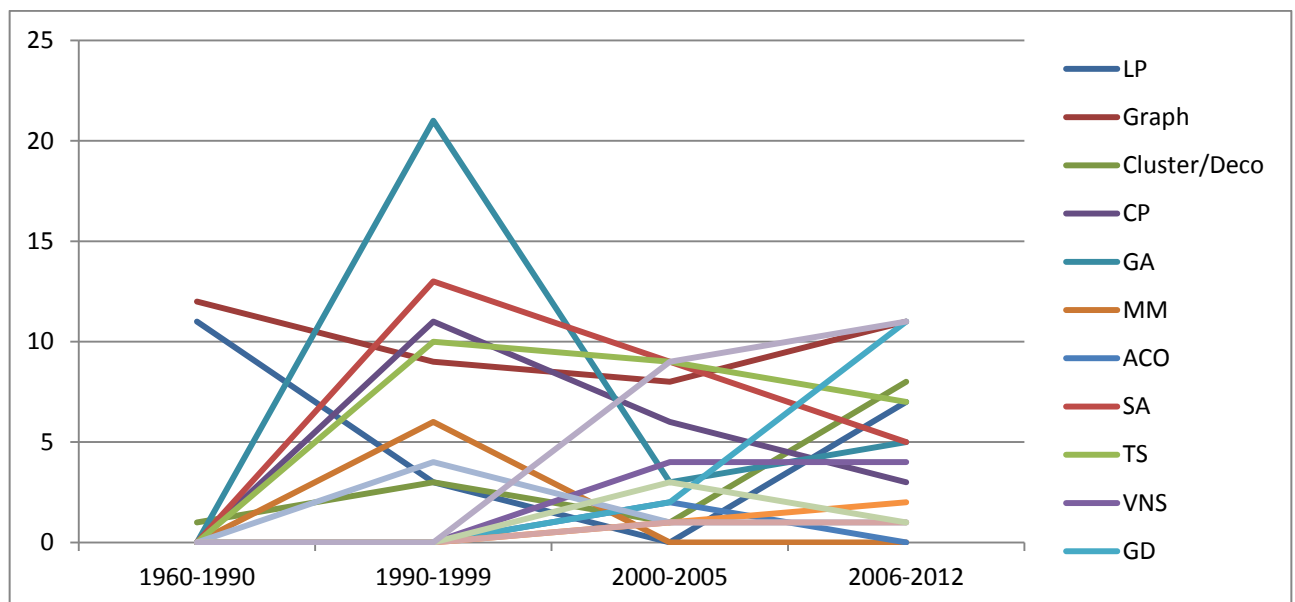
Aquí podem veure que els algorismes fan servir els mètodes meta-heurístics tant per aproximació en una etapa, com en dos etapes (tot i que Simulated Annealing és molt utilitzat en la segona etapa).

Per fer l'anàlisi també m'he basat en **Zotero**. Zotero és una eina per guardar referències d'articles acadèmics que he fet servir en aquest estudi. Zotero permet analitzar les referències que hi conté (entre elles gran part de les referències de les conferències **PATAT** [25]). Com es pot veure, més de la meitat dels articles que proposen algorismes fan servir meta-heurístics, entre aquests els més populars són **GA**, **SA** i **TS**.



II-lustració 6 Distribució del número d'articles que proposen algorismes basats en diferents mètodes

Si ens fixem en la tendència es pot veure que algorismes com **GD**, **HyperC** i els **Graphs** són els que han tingut una tendència positiva remarcable. **GD** és un algoritme relativament nou, així com l'enfoc dels hyper-heurístics. Els **Graphs** sempre són una forma de reduir el problema CTT.



II-lustració 7 Tendència dels mètodes segons els articles que proposen algorismes que els fan servir

Per acabar he creat un DAFO sobre cadascun dels mètodes categoritzats:

Taula 2 Anàlisi DAFO de mètodes

Mètodes			DAFO			
			Anàlisi Intern		Anàlisi Extern	
			Fortaleses	Debilitats	Oportunitats	Amenaces
Math	LP	Senzill d'entendre i implementar.	No idoni per instàncies grans.	Creació de paquets de software de forma senzilla.	Es un tema que no està a l'agenda de molts investigadors per que ja s'ha treballat molt.	
		Funcionen bé en instàncies petites.				
Graph		Funcionen bé en instàncies petites.	No dona els millors resultats en instàncies grans.	Combinació amb altres mètodes i algorismes de dos fases (per la fase de construcció)	S'han utilitzat tant que alguns estudis no els tenen en compte.	
		La majoria de problemes UTT es poden representar per grafs.				
Cluster /Decom		Molta literatura				
		Funcionen bé en grans instàncies, sobre tot els de descomposició)	No cerquen solucions òptimes (s'han de combinar amb algorismes de optimització)	Han estat sorgint noves aproximacions (tema de interès sobre tot la descomposició).	Algorismes poc genèrics	
Constraint	CP	Funcionen molt bé com algorismes de creació de solucions viables.				
		Aplicables a molts problemes com a algorismes de construcció de solucions viables.	Poden arribar a ser molt complexes.	Combinació amb altres mètodes en algorismes de 2 fases.	Tema molt treballat, alguns autors no el tenen com a línia de futur.	
		Existeixen marcs de treball i solvers de CLP.		Fàcils d'aplicar en molts tipus de problemes.		
		Molta literatura				
Meta- heuristics	Poblacions	GA	Son dels algorismes més utilitzats per optimització	Difícils d'implementar	Combinació amb altres mètodes en algorismes de 2 fases.	Molt treballat
		MM	Bons candidats per a algorismes de 1 fase.			
			(Son extensions de GA)	Difícils d'implementar	Combinació amb altres mètodes en algorismes de 2 fases.	Poc treballat
	ACO	Son dels algorismes més utilitzats per optimització				
		Bons candidats per a algorismes de 1 fase.	Pot arribar a solucions viables però s'ha d'optimitzar per no bloquejar-se.	Fàcils de combinar amb altres algorismes de optimització i sobretot amb algorismes basats en grafs	Poc treballat	
		Bons candidats per a algorismes de dues fases, oferint bons resultats.				
Cerca Local	SA	Son dels algorismes més utilitzats per optimització	Molt costosos	Combinació amb altres mètodes (normalment TS i ILS)	Massa treballat	
		TS	Son bons algorismes per trobar solucions viables (ideals per fase de construcció) o per algorismes de 1 fase.	No sempre troben els resultats de forma òptima	Combinació amb altres mètodes.	Massa treballat

		DAFO			
		Anàlisi Intern		Anàlisi Extern	
Mètodes		Fortaleses	Debilitats	Oportunitats	Amenaces
	VNS	Permeten ampliar l'espai de cerca en la fase de construcció i optimització	Pot ampliar massa l'espai de cerca	Es poden combinar amb gairebé tots els meta-heurístics.	Poc treballat. Pot ser relegat
	GD	Potent mètode amb molt bons resultats en instàncies mitjanes o grans	No ideal per instàncies petites.	Combinat amb SA dona molt bons resultats	Relativament poc treballat
	GRASP	Bons candidats a algorismes de 1 fase	Demana molts recursos.	Els bons resultats fan que sigui un camp d'estudi.	Poc treballat
	ILS	Són bons algorismes per trobar solucions viables (ideals per fase de construcció)	Simple, però costós.	Nous estudis i combinacions li donen nova vida.	Molt treballat
	HC	Algorisme general de cerca local. Fàcil d'implementar	Pot obviar alguns resultats viables	La seva combinació amb altres mètodes	Necessita millores per evitar bloquejos.
Multic		Utilització de tècniques no habituals.	Es difícil saber sobre quins problemes i instàncies pot anar millor, molt específic.	Obre la porta a adaptacions d'altres tipus de problemes al problema del timetabling.	Poc treballat
HyperH		El fet de treballar amb solucions ja trobades l'acosta a problemes reals (case-based reasoning)	Complexitat en la implementació i definició	Camp per estudiar	S'ha de saber com combinar heurístics i quins fer servir

4.2 Conclusions

En el moment de triar un algorisme primer s'ha d'analitzar quin és el **tipus de problema**, la **dimensió** i **restriccions**. Així per problemes amb paràmetres inicials senzills podem pensar que les tècniques basades en grafs (**Graph**) o en formulacions matemàtiques (**Math**) com la programació lineal (**LP**) són bons candidats per trobar solucions sense un gran cost en la seva implementació.

A mesura que augmenta la complexitat altres tècniques poden funcionar millor. Les tècniques de **Cluster** i sobre tot les de **Descomposició** ens permeten tractar millor el problema, tot i que la implementació és també complexa. La qualitat de les solucions no sol ser òptima, però és en aquest moment que es poden fer servir **Hyper-heurístics** o **Meta-heurístics** per tal de optimitzar les solucions.

Els meta-heurístics són els millors candidats a mètodes per resoldre i, sobre tot, optimitzar les solucions a problemes de timetabling. Entre aquests hi destaquen els mètodes basats en cerca lineal com **Tabú search** i **Simulated Annealing**, i mètodes evolucionaris com els **Algorismes Genètics**. Aquests poden ser fàcilment combinats per crear algorismes que resolen el problema en múltiples etapes. És per això que tenim un gran arsenal de meta-heurístics disponibles i són els mètodes més emprats.

La combinació de diferents tècniques ens obre la possibilitat de que apareguin noves tècniques, i per tant la hibridació de mètodes és una de les línies de futur com es veu en els treballs de **Qu et al.**[57].

Altres mètodes com les tècniques **multi-criteri** i els **hyper-heurístics** ens obren nous camps en la recerca de algorismes per a timetabling.

5 Estudi d'eines de timetabling

La necessitat d'eines que permetin facilitar la feina de resoldre els problemes de generació i gestió de taules d'horaris universitaris, ha fet que al llarg dels anys hagin anat apareixent paquets de software en aquest àmbit.

Gràcies la recerca en el Timetabling Universitari han aparegut sistemes i llibreries, tant comercials com de programari lliure, basats les tècniques descrites en aquesta recerca. De fet, molts projectes de recerca i tesis han seguit l'estratègia de recerca de **Disseny i Creació** [80] obtenint artefactes que han esdevingut en paquets de software.

En aquest estudi ens centrarem en analitzar una sèrie de paquets de software, tant lliure com comercial, que són força utilitzats en institucions arreu del món. En centrarem primordialment en els paquets de programari lliure, ja que aquest és el camp d'estudi del Màster, i perquè aquestes eines són més fàcils d'analitzar (lliure accés al codi, documentació,...).

Tot i així s'ha de destacar que alguns autors han desenvolupat artefactes que s'han convertit en paquets de software per a les universitats o centres universitaris on han realitzat la recerca. El problema d'aquests paquets és que són molt específics per el centre que els fa servir. No obstant, les metodologies descrites en els seus treballs poden ser la base per a la creació de paquets generalistes. Trobem exemples als treballs de **Lim et al** [81] amb el seu sistema **UTTSEExam** per a la universitat de Singapur, i al de **Dimopoulou et al** [82] per al universitat d'Atenes emprant tècniques de Programació Linear.

Com a referència ens trobem un estudi de **Peter Karich** [83] que recull algunes de les principals eines, tant comercials com de programari lliure, de Timetabling. Aquest estudi va en la mateixa línia d'aquest treball de recerca, tot i que nosaltres ens centrarem en una sèrie de paquets concrets

5.1 Paquets de programari lliure

5.1.1 UniTime

Alguns dels autors com **Müller i Rudová** han aprofitat la seva recerca en àmbits com la Programació de Restriccions (Constraint Programming CP) i les Tècniques de Descomposició [13], [21], [84], [85] per desenvolupar i millorar sistemes. En aquest cas Müller és el principal desenvolupador del paquet i API **UniTime** [86] que permet la creació automàtica de taules d'horaris i exàmens per a institucions acadèmiques.

UniTime és un sistema multi-plataforma escrit en Java que permet trobar solucions a tots els tipus de problemes de timetabling educatiu (ETT, CB-CTT, PE-CTT, HSCTT), a més de permetre particionar els estudiants (**Sectioning Timetabling**), i interactuar amb els planificadors (**Interactive Timetabling**)⁴

Està format per un motor basat en resolució de restriccions (Constraint Programming Solver), un mòdul d'optimització (que accepta algorismes com Simulated Annealing, Tabú Search) i un mòdul de presentació i interacció amb l'usuari basat en a web. Aquesta estructura permet adoptar nous enfocaments en quant a l'optimització de forma que el fa molt flexible.

UniTime és un dels paquets que permet trobar solucions a problemes reals, que tenen un gran número de restriccions a complir. Destacar que el seu motor de resolució de restriccions ha resultat **guanyador en el ITC2007**.

⁴ En l'apartat de definició d'aquest treball trobem les definicions d'aquests problemes.

Una bona documentació acompanya al programari, tant per l'usuari com per els desenvolupadors.

5.1.2 Drools Solver Planner

Drools Solver Planner[87] no és un paquet de programari complet, ja que no té una interfície que permeti interactuar sinó només un motor per buscar solucions.

És un sistema multi-plataforma escrit en Java que permet trobar solucions a tots els problemes de timetabling universitari(ETT, CB-CTT, PE-CTT), si aquests es modelen correctament.

Està format per un motor de regles basat en l'algorisme de **Rete**[88] i un mòdul que permet incorporar algorismes d'optimització fàcilment. Ara per ara incorpora Tabu Search, Simulated Annealing, Great Deluge i Hill Climbing.

Drools destaca per la seva flexibilitat i facilitat per crear formulacions de problemes. Crear les restriccions basades en regles és molt senzill i intuïtiu. Està orientat a objectes i accepta entrades en format XML. A més, permet afegir mòduls d'optimització molt ràpidament.

És per això que es emprat a moltes institucions com a solver de problemes de timetabling universitari. Pot ser el nucli d'un sistema desenvolupat ad-hoc.

Drools forma part del projecte Apache JBoss i gaudeix d'una gran comunitat i d'una documentació excel·lent.

5.1.3 FET (Free Timetabling Software)

En la mateixa línia que UniTime, **Liviu Lalescu** ha emprat la recerca en Algorismes Genètics [89], [90] i de cerca local per desenvolupar el paquet de software lliure **FET** [91].

FET és un sistema multi-plataforma escrit en C++ que permet trobar solucions a problemes de timetabling educatiu (ETT, PE-CTT, HSTT).

La documentació de FET no és tan bona com la de Drools o UniTime, però és completa tant per usuaris com per desenvolupadors

Tot i ser molt potent, la interfície de FET no és molt intuïtiva i no inclou una interfície Web (només per visualitzar).

5.2 Paquets comercials

5.2.1 Lantiv Scheduling Studio

Lantiv Timetabler[92] és un dels clàssics paquets de programari de generació d'horaris comercial.

Lantiv només funciona en plataformes Windows. Escrit en C++, permet trobar solucions a problemes de timetabling educatiu (ETT, PE-CTT, HSTT).

Destaca la seva interfície(no web) molt intuïtiva i usable, molt semblant a un full Excel.

La documentació d'usuari és molt completa i ben estructurada.

5.2.2 Eventmap Limited

Eventmap[93] és un dels paquets de programari punters en quant a timetabling educatiu.

És un sistema multi-plataforma escrit en C++ que permet resoldre qualsevol problema de timetabling universitari (ETT, CB-CTT, PE-CTT), així com problemes de particionament d'estudiants (**Sectioning Timetabling**).

Eventmap està basat en un motor de resolució de grafs que es complementa per un mòdul totalment configurable de meta-heurístics (Tabú Search, Simulated Annealing, Iterative Search,

Great Deluge,..). Això el fa un dels paquets més complets i més avançats (va incorporant en cada versió noves tècniques).

Un dels principals punts a favor de EventMap és que l'empresa que el desenvolupa és una spin-off del grup **ASAP**[94] de les universitats **Queen's de Belfast** i **Nottingham**. ASAP és possiblement el grup de recerca amb més prestigi en l'àmbit de Automtated Timetabling del món. La gran majoria de referències d'aquest treball són d'investigadors associats a ASAP. Autors com Burke, Qu i McCollum són clars exemples.

5.3 Anàlisi i resultats

La següent taula DAFO ens mostra els punts forts i febles dels paquets analitzats. Per realitzar l'anàlisi s'han emprat les demos interactives dels paquets per veure aspectes com la usabilitat. Per altres aspectes s'ha estudiat la documentació de cada paquet.

Taula 3 Anàlisi DAFO d'eines

Paquet		DAFO			
		Anàlisi Intern		Anàlisi Extern	
		Fortaleses	Debilitats	Oportunitats	Amenaces
Open Source	UniTime	<p>El més complet en quant a tipus de problemes que abasta (HSTT, ETT, CB-CTT, PE-CTT) a més de Sectioning i Interactive.</p> <p>Algorismes d'optimització potents (SA, TS) i configurables</p> <p>Interfície Web senzilla i intuïtiva.</p> <p>Multi-plataforma i multi-idioma.</p> <p>Documentació excel·lent</p> <p>Diferents visions segons els usuaris (planificador, professor, alumne).</p> <p>Testejat en casos reals.</p> <p>Guanyador ITC2007</p> <p>Desenvolupament molt actiu.</p>	<p>El fet d'estar centrat en un motor de Constraint Programming no permet altres algorismes de creació.</p>	<p>Diverses universitats estan interessades o ja han provat UniTime.</p> <p>En ser obert pot millorar amb una bona gestió de la comunitat.</p> <p>Grups de recerca sobre timetabling educatiu al darrere.</p>	<p>Fork del projecte (poc probable)</p>

Comercial	FET	<p>Permet solucionar diferents problemes de timetabling educatiu(ETT, PE-CTT, HSTT).</p> <p>Algorismes de optimització basats en algorismes genètics (GA)</p> <p>Àmpliament utilitzat(va ser un dels primers paquets).</p> <p>Testejat en casos reals</p>	<p>Documentació poc estructurada.</p> <p>Poc intuïtiu.</p> <p>Només té un web viewer, l'aplicació és d'escriptori.</p>	Comunitat activa	Fork del projecte , alguns centres han desenvolupat aproximacions ad-hoc.
	Drools Solver	<p>Pot resoldre qualsevol problema de timetabling universitari-</p> <p>Flexibilitat: permet modelar el problema fàcilment i afegir nous optimitzadors.</p> <p>Multi-plataforma.</p> <p>Gran documentació.</p> <p>Algorismes d'optimització potents (SA, TS) i configurables</p> <p>Testejat com a Motor.</p> <p>Desenvolupament molt actiu</p>	<p>Només és un motor, no té interfície.</p> <p>No multi-idioma.</p>	<p>Desenvolupament</p> <p>Pot arribar a ser el nucli de molts altres paquets.</p> <p>Gran comunitat recolzada per Apache Foundation.</p>	Massa generalista, pot perdre comunitat.
	Lantiv Scheduling Studio	<p>Pot resoldre problemes de timetabling educatiu (ETT, PE-CTT, HSTT)</p> <p>Interfície intuïtiva i usable.</p> <p>Documentació molt completa.</p>	<p>Cost acceptable</p> <p>Poc flexible: no permet incorporar nous algorismes</p> <p>No té interfície web (es pot accedir remotament)</p> <p>No multi-plataforma</p>	Emprat en molts centres de secundària.	Temps de alliberament de versions alt.
	EventMap	<p>Gran varietat de problemes que abasta (ETT, CB-CTT, PE-CTT), i Sectioning.</p> <p>Algorismes d'optimització potents (SA, TS , GD) i configurables.</p> <p>Diferents visions segons l'usuari.</p> <p>Multi-plataforma i multi-idioma.</p> <p>Permet crear planificacions per fer simulacions.</p> <p>Interfície intuïtiva i usable.</p>	Cost	L'equip de Eventmap està format per els líders de la recerca en timetabling educatiu.	S'han utilitzat tant que alguns estudis no els tenen en compte.

Un dels objectius d'aquest treball és descriure les funcionalitats que ofereixen els paquets de software per veure si són les requerides per els usuaris de la UPC.

Un anàlisi qualitatiu sobre la documentació dels diferents paquets de programari ens ha permès veure quines són les funcionalitats comunes que tenen aquests, i quines no són comunes però són interessants. Ens em centrat en els requeriments funcionals, que són els que entenen millor els usuaris del sistema, i no pas en els requeriments no funcionals.

- **Funcionalitats per la gestió d'horaris**
 - Generació automàtica de taules d'horaris
 - Generació automàtica de calendari d'exàmens.
 - Modificació manual de taules d'horaris (resolent conflictes)
 - Gestió d'espais (aules i laboratoris), les seves característiques (equipaments) i disponibilitat.
 - Avís de col·lisions/incoherències (assignatures/professors/espais).
 - Proposta de solucions a col·lisions/incoherències.
 - Gestió de la disponibilitat de professors (per exemple: per hores de consulta, hores màximes setmanals,...).
 - Visualització d'horaris des de diferents perspectives (nivell i grup, alumne, professor).
- **Planificació:**
 - Gestió de l'encàrrec docent(Determina les hores de docència que els departaments oferiran)
- **Explotació i eines per l'usuari**
 - Accés al la informació via Web.
 - Simulació d'horaris per alumnes (abans de la matrícula)
 - Impressió de les taules d'horaris en diferents formats(PDF, EXCEL)
 - Generació d'informes ad-hoc de les dades (generador de consultes per fer informes personalitzats).
 - Informes de la qualitat de les taules generades.
 - Anàlisi de l'activitat docent en funció de les taules horàries (permet generar informes de l'activitat docent de professors i departaments en hores i punts a partir dels horaris)
- **Interoperabilitat**
 - Exportació de dades a altres sistemes

La següent taula relaciona les funcionalitats anteriors amb les eines estudiades (UniTime i EventMap són les que compleixen més funcionalitats)

Taula 4 Funcionalitats i eines estudiades

Funcionalitats		Paquets de programari				
		UniTime	FET	Drools ⁵	Lantiv	EventMap
Funcionalitats per la gestió d'horaris	Generació automàtica de taules d'horaris					
	Generació automàtica de calendari d'exàmens.					
	Modificació manual de taules d'horaris.					
	Gestió de espais (aules i laboratoris), les seves característiques (equipaments) i disponibilitat.					
	Avis de col·lisions/incoherències					
	Proposta de solucions a col·lisions/incoherències.					
	Gestió de la disponibilitat de professors					
	Visualització d'horaris des de diferents perspectives					
Planificació	Gestió de l'encàrrec docent					
Explotació i eines per l'usuari	Accés al la informació via Web.					
	Simulació d'horaris per alumnes					
	Impressió de les taules d'horaris en diferents formats(PDF, EXCEL)					
	Generació d'informes ad-hoc de les dades					
	Informes de la qualitat de les taules generades.					
	Anàlisi de l'activitat docent en funció de les taules horàries					
Interop	Exportació de dades a altres sistemes					

5.4 Conclusions

En un primer anàlisi els paquets **EventMap** i **Unitime** semblen els més complets en cada àmbit. Tots dos són configurables (Unitime, a més, es pot estendre degut a la seva natura lliure). Tots dos tenen darrere investigadors reconeguts i molt actius, que apliquen els seus coneixements en el programari. A més són els que compleixen amb més funcionalitats.

Drools Solver té un potencial enorme per construir programari a mida, és un motor molt senzill i orientat a objectes, que permet entrades en formats estàndard XML.

Les funcionalitats ens permetran classificar altres eines, així com desenvolupaments a mida.

⁵ Drools no és una eina d'usuari, tot i que les funcionalitats es poden aconseguir.

6 Estudi del timetabling universitari a la UPC

6.1 La UPC

La Universitat Politècnica de Catalunya és una universitat pública amb orientació als ensenyaments tècnics i enginyeries. Va ser creada al 1971, tot i que alguns dels seus centres havien estat fundats a mitjans del segle XIX, per aglutinar els estudis tècnics a Catalunya.

Com a dades generals està formada per **23 escoles i facultats** distribuïdes en **8 ciutats** catalanes. En aquests s'imparteixen un total de **69 estudis de grau**, **62 programes de màster** i **48 programes de doctorat**. Hi estudien uns **32.000 estudiants** (entre grau, màster i doctorat) amb uns **2.600 professors/investigadors** i **1.500 treballadors d'administració i serveis**. Hi existeixen **42 departaments docents** que abasten un gran ventall d'àmbits de coneixement tècnics.

6.2 Conceptes previs

Les entrevistes a dos gestors de centre i l'enquesta realitzades ens han ajudat a definir conceptes comuns que s'utilitzen als centres estudiats.

La docència del centre es encarregada per la **direcció acadèmica del centre** als **departaments docents**. Aquests són unitats organitzatives per àrea de coneixement on estan assignats els professors. Els departaments docents donen forma a les assignatures i indiquen quins professor les impartiran.

L'**encàrrec docent** és la distribució de la docència (en quant a número de grups, número d'hores totals de les assignatures i la seva distribució en hores de teoria i laboratori) que elabora la direcció acadèmica del centre segons les previsions de matrícula. Els departaments docents han d'assignar recursos (professors) necessaris per cobrir aquest encàrrec.

Els plans d'estudis UPC estructuren les assignatures en **nivells**. Els nivells són conjunts d'assignatures que moltes vegades es corresponen amb espais temporals (en tots els centres tenim un nivell per quadrimestre, el primer i segon nivell es corresponen amb les assignatures del primer i el segon quadrimestre de la carrera). Com a restricció forta les assignatures d'un nivell no han de tenir solapaments entre elles.

Depenent del volum del centre, podem tenir molts alumnes per assignatura, que pot representar un problema de capacitat en les aules o bé fer que el número d'alumnes per professor impliqui deficiències en la docència i seguiment dels alumnes. En aquests casos les assignatures es divideixen en més d'un **grup de matrícula**. Així, podem tenir més d'un grup per nivell i per tant re-formulem la restricció forta anterior de forma que només les assignatures de un nivell i grup concret no han de tenir col·lisions entre elles.

Els caps d'estudis es basen en les dades de matrícula d'anys anteriors per tal de definir en quants grups hauran de dividir una assignatura.

En qualsevol moment poden aparèixer incidències sobre diferents recursos que necessitin introduir canvis en les taules horàries. En aquests casos un sistema de timetabling interactiu pot ajudar a resoldre conflictes que es puguin generar per aquests canvis.

6.3 Anàlisi i resultats

Aquest anàlisi es basa en els resultats obtinguts d'un qüestionari de timetabling acadèmic passat als centres i de dos entrevistes realitzades als tècnics de gestió d'horaris de dos escoles de la UPC: la EUETIB i la EPSEVG.

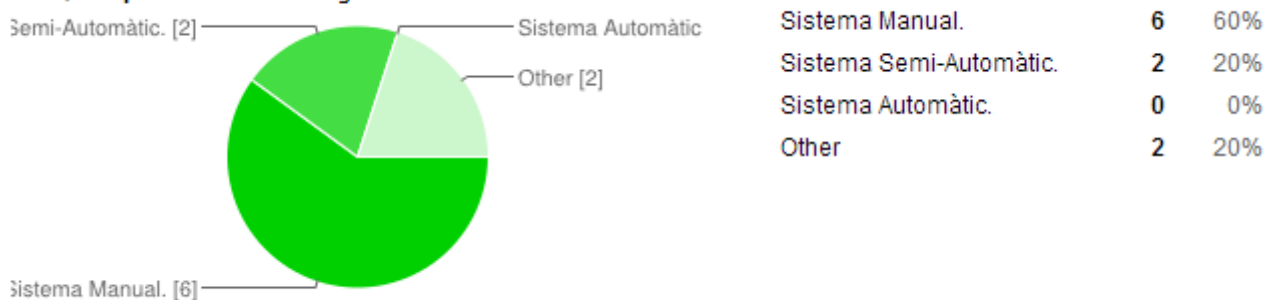
Tots els centres tracten la programació de lectures com un problema **CB-CTTP** de forma que programen les seves taules horàries abans de la matrícula, definint les compatibilitats de les assignatures amb els plans d'estudis de cadascuna de les carreres que imparteixen.

En un centre amb més alumnes s'han d'oferir més grups per nivell, dificultant la cerca de una solució òptima (on es compleixen totes les restriccions fortes i s'intenta complir el màxim de restriccions dèbils).

També tenim **diferències en els recursos** de que disposa cada centre en relació al volum d'alumnes. És normal que un centre amb menys recursos tingui més dificultats en el procés de generació de taules horàries, i hagin de prendre decisions com l'ampliació de l'horari lectiu (en el cas de falta d'aules).

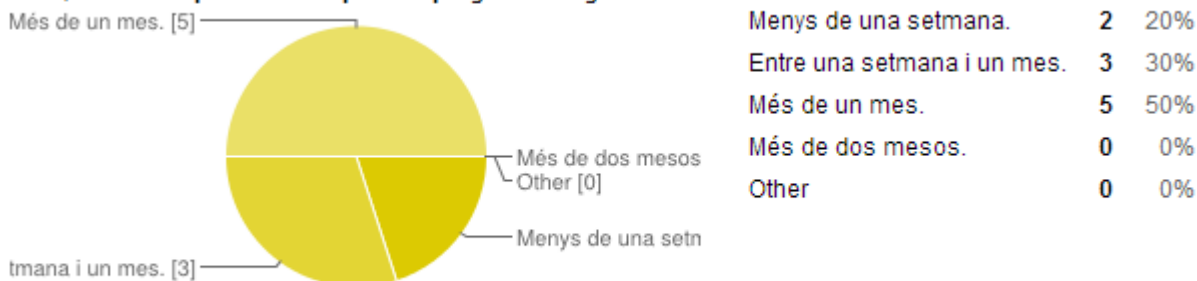
L'enquesta ens revela que cap dels centres enquestats fa servir un sistema automàtic de gestió d'horaris.

3.1 Quin tipus de sistema de generació de taules d'horaris feu servir?



Això contrasta amb el **temps que es dedica a l'elaboració** dels mateixos, on un 50% triga entre un i dos mesos en confeccionar-los.

2.5 Quan de temps feu servir per a la programació/generació de horaris?

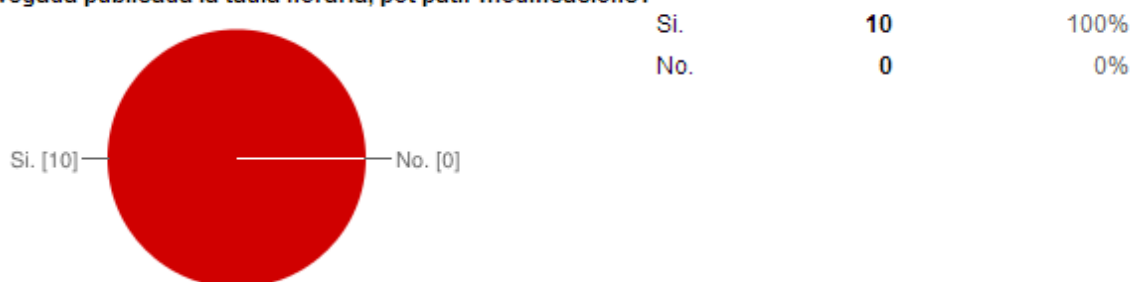


Tot i algunes petites diferències, tots els centres tenen un procés de generació de taules d'horaris similar. Així en general el **procés** d'elaboració d'horaris passa per les següents fases:

- 1 Recopilació d'informació**
 - 1.1 Definició de Calendari Acadèmic:** La direcció del centre decideix dates de matrícula, avaluació, festius..
 - 1.2 Definició de Currículum:** Definició/validació de tipus assignatura, grups que tindrà, capacitats dels grups
 - 1.3 Disponibilitat d'aules:** es comproven els equipaments i disponibilitats de les aules (que no hi hagin indisponibilitat per obres..)
- 2 Elaboració versió preliminar:** els tècnics de gestió d'horaris elaboren una versió preliminar de quadres horaris, normalment en base a una versió d'un curs anterior.
- 3 Assignació Docent:** Els departaments assignen els professors que impartiran les classes sobre la versió preliminar, i indiquen canvis en els horaris per diferents motius.
- 4 Elaboració versió completa:** Els tècnics apliquen les assignacions de professors i intenten complir les restriccions/canvis indicats .En cas de conflicte es deixa que els caps de departament decideixin o bé aplicar criteris com donar més importància a les assignatures que estadísticament tenen un volum major d'alumnes.
- 5 Validació Final:** Els caps departament docent validen els horaris que queden llestos per publicar.
- 6 Publicació:** Finalment els horaris es publiquen

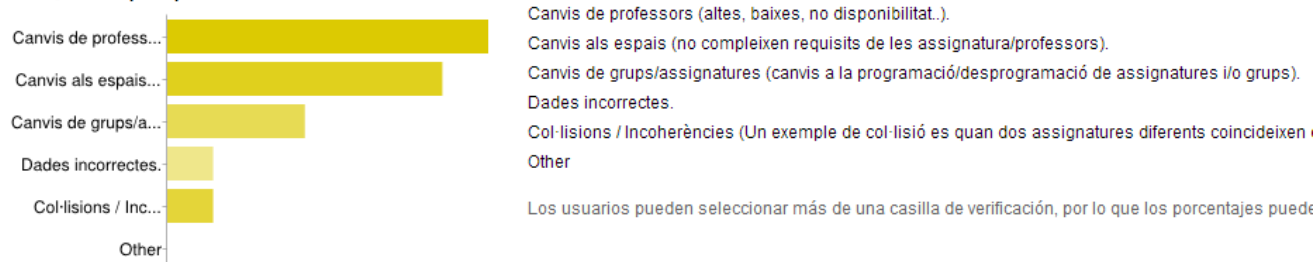
Tots els centres es queixen de que una vegada publicats els horaris arriben **modificacions**.

2.10 Una vegada publicada la taula horària, pot patir modificacions?



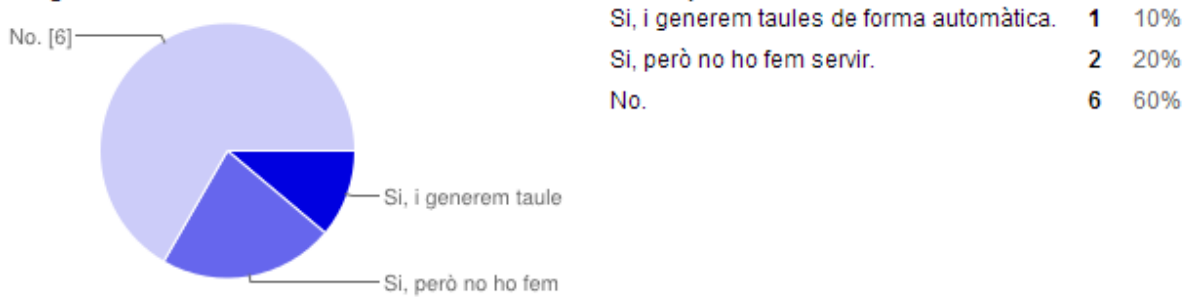
El percentatge de canvis en la taula horària pot arribar al 50% en alguns centres. En preguntar quins són els motius de canvi, la gran majoria són **canvis de professors i d'aula**. Una correcta gestió de la informació podria fer disminuir aquest tipus de canvis.

2.12 Quin és el principal motiu de canvis als horaris?



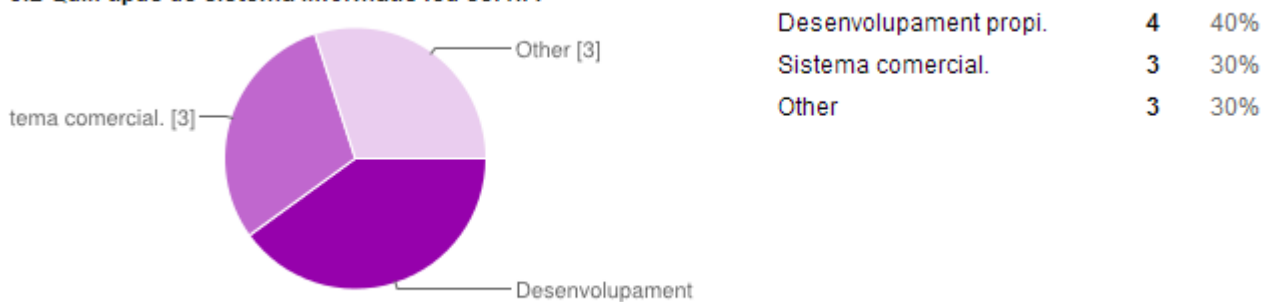
En quant als **sistemes de gestió actuals** veiem que només un 30% dels sistemes permeten generar taules d'horaris de forma automàtica i només un centre fa servir aquesta funcionalitat. Això pot indicar que, o bé el sistema de generació és molt complex, no dona bons resultats o que els usuaris no estan formats per fer-lo servir.

3.4 El sistema genera taules horàries de forma automàtica i feu servir aquesta funcionalitat?



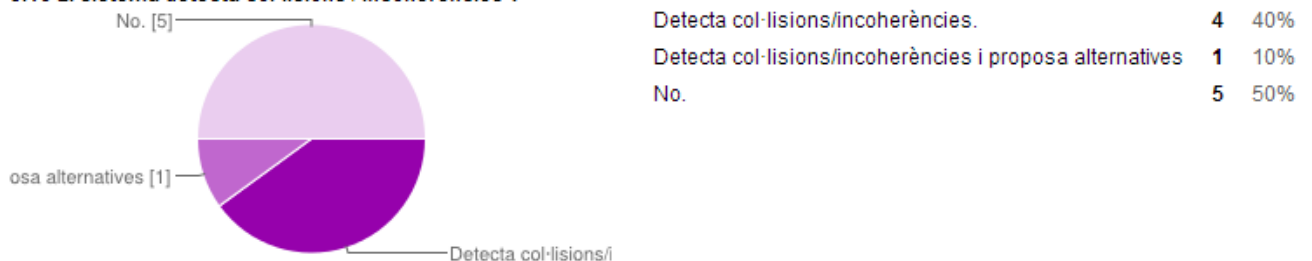
Més de la meitat dels sistemes són desenvolupaments propis (que indiquen que cada cas és específic).

3.2 Quin tipus de sistema informàtic feu servir?



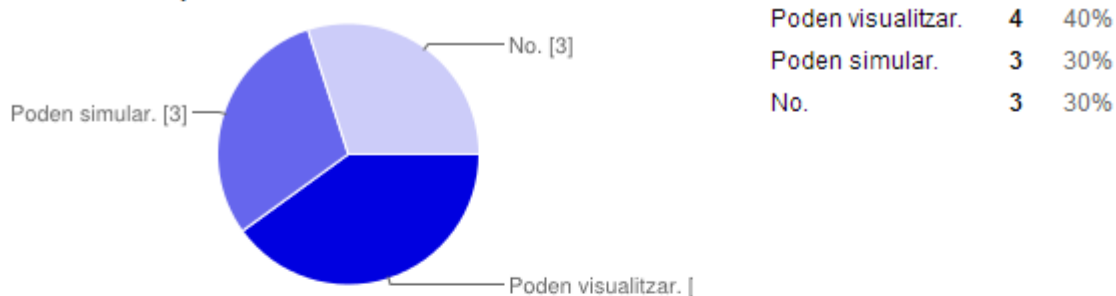
Més de la meitat dels sistemes emprats no detecten col·lisions o incoherències. Això penalitza molt la tasca dels gestors d'horaris.

3.10 El sistema detecta col·lisions / incoherències ?

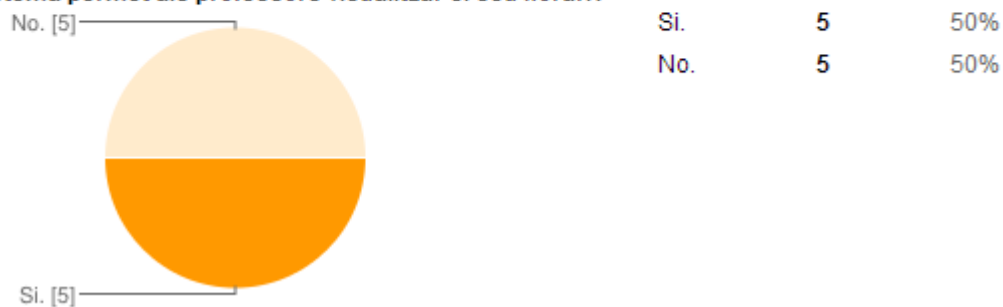


La visualització de dades per la web supera el 50% en els sistemes emprats, tot i que només un 30% també permeten als usuaris simular la seva taula horària. Ara per ara poder accedir als horaris via web és un factor molt important, que evita tenir còpies impreses del horaris penjades i és molt més còmode de consultar per estudiants i alumnes.

3.11 El sistema permet als alumnes visualitzar i simular els seus horaris?

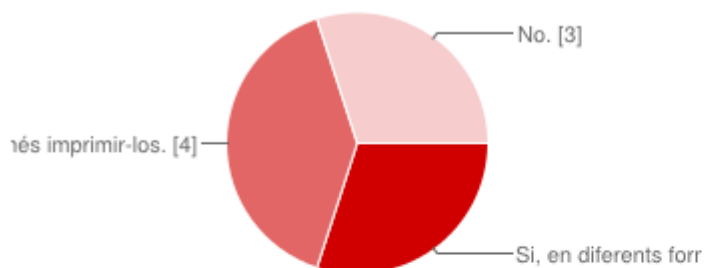


3.12 El sistema permet als professors visualitzar el seu horari?



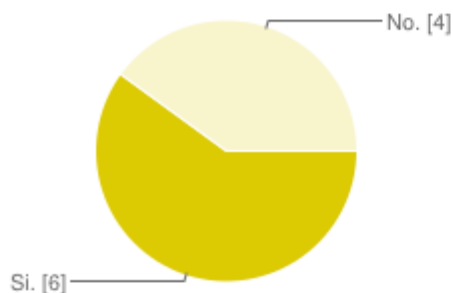
En quant a l'explotació de la informació, veiem que la gran majoria es comporten correctament en quant a la impressió, però no en quant a la generació d'informes ad-hoc i la interoperabilitat

3.13 El sistema permet imprimir els horaris i en diferents formats?



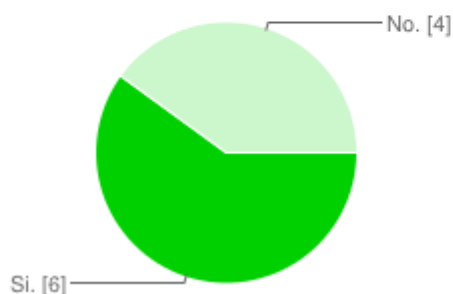
Si, en diferents formats per poder formatejar-
 Si, només imprimir-los.
 No.

3.14 El sistema genera informes estàtics de la informació continguda?



Si.	6	60%
No.	4	40%

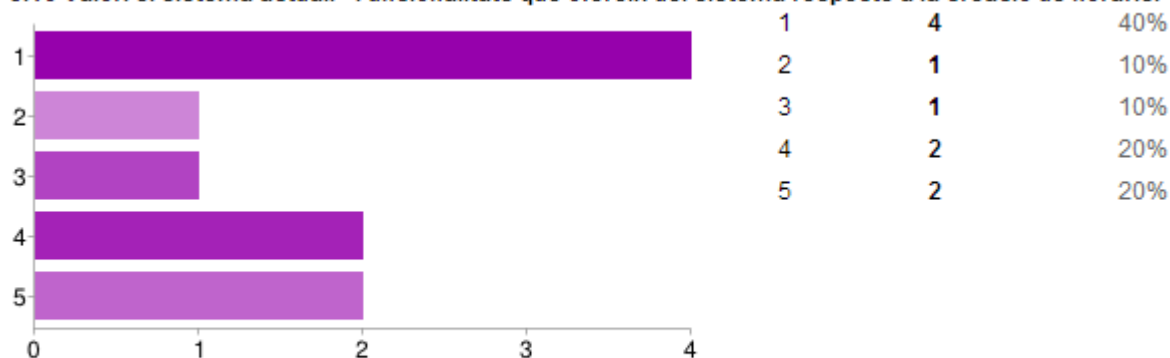
3.15 El sistema permet l'exportació de dades a altres sistemes (PRISMA, ATENEA,..)?



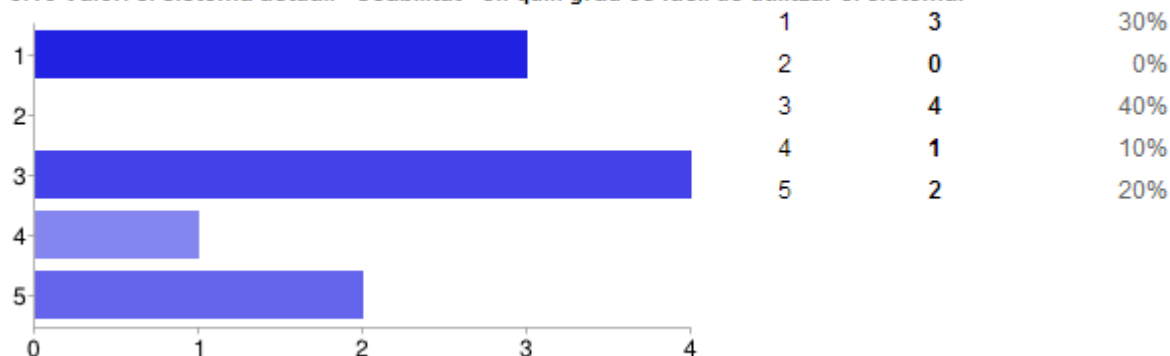
Si.	6	60%
No.	4	40%

La **valoració** dels usuaris respecte al sistema està en el punt mig, menys quan es pregunta per el motor de creació d'horaris on un 40% suspèn el sistema que fan servir actualment.

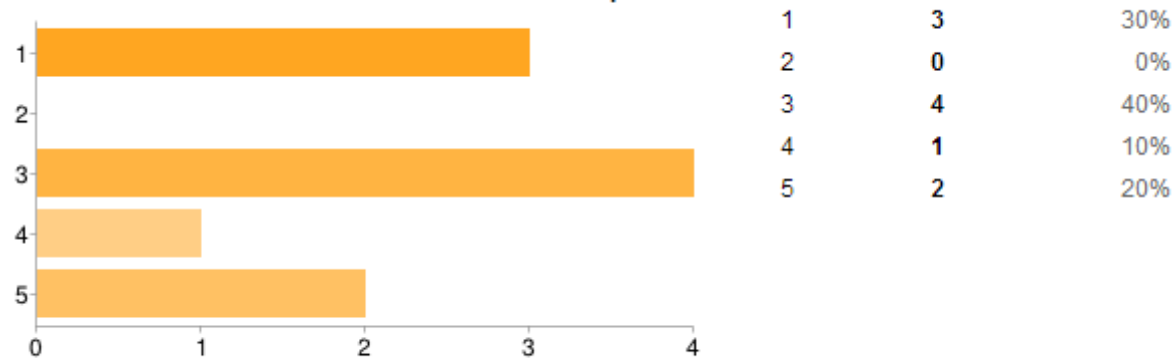
3.16 Valori el sistema actual: - Funcionalitats que ofereix del sistema respecte a la creació de horaris.



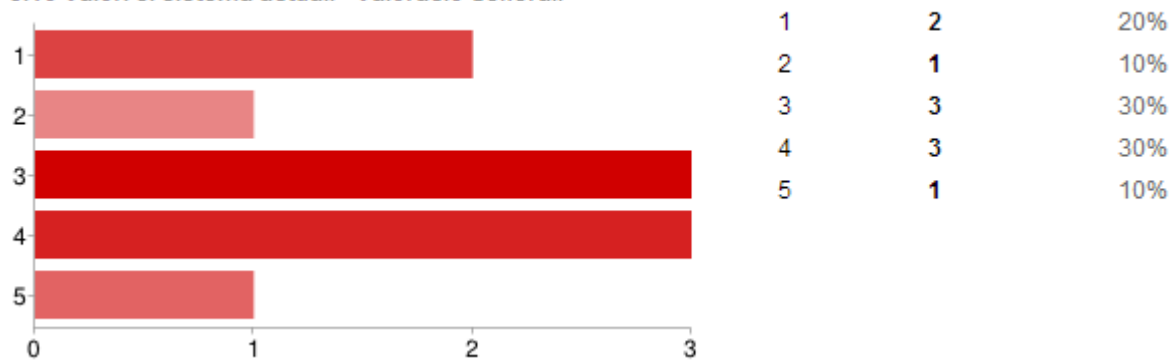
3.16 Valori el sistema actual: - Usabilitat - en quin grau és fàcil de utilitzar el sistema.



3.16 Valori el sistema actual: - Flexibilitat - com s'adapta el sistema als canvis.



3.16 Valori el sistema actual: - Valoració General.



El qüestionari pregunta als tècnics gestors quines són les funcionalitats que consideren més importants i el resultat és que gairebé totes les funcionalitats són considerades importants

Taula 5 Com consideren els centres les funcionalitats d'un sistema

Funcionalitats		Important	Desitjable	Irrellevant
Funcionalitats per la gestió d'horaris	Generació automàtica de taules d'horaris	30%	60%	10%
	Generació automàtica de calendari d'exàmens.	20%	70%	10%
	Modificació manual de taules d'horaris.	100%		
	Gestió de espais (aules i laboratoris), les seves característiques (equipaments) i disponibilitat.	80%	20%	
	Avís de col·lisions/incoherències	100%		
	Proposta de solucions a col·lisions/incoherències.	20%	80%	
	Gestió de la disponibilitat de professors	30%	40%	30%
	Visualització d'horaris des de diferents perspectives	70%	30%	
Planificació	Gestió de l'encàrrec docent	40%	50%	10%
Explotació i eines per l'usuari	Accés al la informació via Web.	80%	10%	10%
	Simulació d'horaris per alumnes	40%	60%	
	Impressió de les taules d'horaris en diferents formats(PDF, EXCEL)	70%	20%	10%
	Generació d'informes ad-hoc de les dades	50%	50%	
	Informes de la qualitat de les taules generades.	20%	30%	50%
	Anàlisi de l'activitat docent en funció de les taules horàries	60%	40%	
Interop	Exportació de dades a altres sistemes	80%	20%	

L'anàlisi detallat de les entrevistes en quant al software que fan servir ens mostra que:

La **EPSEVG** fa servir un software de desenvolupament propi, que han anat millorant curs rere curs i ara es troba en una fase madura. El motor de generació d'horaris es basa en un model híbrid que pot calcular solucions en base a la informació introduïda prèviament (fent servir algorismes de cerca lineal) i que es pot modificar interactivament pels usuaris.

Normalment es reaprofitava un model d'un curs anterior i s'introdueixen modificacions manuals.

Tant professors com alumnes poden accedir via web als horaris una vegada publicats. Els professors poden veure el seu horari setmanal i diari.

Els alumnes poden confeccionar una versió del seu horari prèvia a la matrícula mitjançant un simulador web que els permet veure si tindran col·lisions segons les assignatures i grups de matrícula que escullin.

Al ser un desenvolupament propi, aquest es pot adaptar a canvis en la normativa del centre i en l'estructura dels plans d'estudi amb una certa agilitat.

El grau de satisfacció del sistema és molt alt, i s'adapta a les necessitats dels tècnics i usuaris del sistema.

La **EUETIB** fa servir un sistema comercial que està implantat a altres centres de la UPC. Aquest permet la creació automàtica de taules horàries en base a restriccions basat en un model d'heurístics no ampliable. El sistema permet la generació manual dels horaris, basant-se o no taules horàries d'anteriors quadrimestres. El sistema detecta col·lisions i incoherències en els horaris de forma no intrusiva: avisa l'usuari de la col·lisió però el deixa continuar treballant. El sistema no suggereix solucions a les col·lisions.

Actualment el centre no fa servir el motor de generació automàtica d'horaris i fan servir un model manual basant-se en taules horàries d'anteriors quadrimestres.

A més el sistema ofereix funcionalitats de reserva d'espais i creació de taules de exàmens, que s'utilitzen cada any en el centre.

El sistema permet, mitjançant un mòdul addicional, accedir als horaris via web. Els professors poden consultar el seus horaris així com els alumnes.

Els alumnes poden confeccionar una versió del seu horari prèvia a la matrícula mitjançant un simulador web, aquest no detecta col·lisions.

El grau de satisfacció del sistema és mig, ja que no s'adapta amb facilitat als canvis normatius del centre i d'estructura dels plans d'estudi.

6.4 Conclusions

Als centres petits, degut als seus paràmetres inicials, no té sentit el fet de fer servir un sistema de generació automàtica d'horaris. És millor opció tenir un sistema que permeti la fàcil introducció de les dades com a base de dades per una posterior consulta i explotació (imprimir horaris, publicació).

En els centres grans si que té sentit la creació d'un sistema de generació d'horaris. Tot i que la majoria es basa en copiar les dades de un quadrimestre anterior i afegir les modificacions en molts dels casos s'acaba modificant tota la taula, amb un cost temporal considerable.

El grau de satisfacció fa pensar que els usuaris dels sistemes actuals els accepten, però creuen que haurien de funcionar com eines de suport. Així desitjarien tenir un sistema que pugui gestionar amablement les col·lisions.

Un punt de millora podria ser la creació de uns sistema d'adquisició d'informació entre departaments gestors i els tècnics gestors d'horaris que tingués un temps màxim per introduir modificacions.

Està clar que, degut al gran nombre de modificacions, qualsevol sistema de generació de taules d'horari automàtic ha de permetre que es puguin incorporar canvis de forma manual.

7 Conclusions Generals

En conjunt em vist que la generació de taules d'horaris automàtiques per a centres universitaris és un tema complex. Ara per ara sabem identificar el tipus de problema, les restriccions i els diferents mètodes que ens poden portar a solucionar el problema. Cada institució definirà la seva instància del problema i els resultats d'aquest treball poden ajudar a triar els adients.

En analitzar els centres de la UPC (només una universitat) podem afirmar que cada centre enfoca el procés de forma similar. Aquests imposen restriccions que re-formulen el problema i el fan diferent en funció de requeriments acadèmics, estructurals o de la dimensió del centre. Això no és un inconvenient, és la realitat.

L'aplicació de certs algorismes, o no, dependrà de tots aquests factors, i les taules DAFO poden ajudar a decidir quins mètodes fer servir. Les eines triades hauran de suportar aquests mètodes en els seus motors de resolució i/o optimització, o permetre afegir-los ad-hoc. A més a més, les eines han de oferir les funcionalitats que hem enumerat en l'apartat corresponent. Les eines que compleixin tant amb funcionalitats com amb els mètodes triats seran bones candidates. Aquests criteris són igualment vàlids per desenvolupaments a mida.

Una de les conclusions de l'estudi és que els centres més petits no necessiten una aplicació de generació d'horaris de forma automàtica. Tot i que tenir una base de dades unificada per explotar la informació podria ajudar molt. Podem pensar que els seus paràmetres inicials poden canviar, i llavors necessitar una eina automàtica.

7.1 Línies de Futur

Les línies de futur del projecte passen per la formulació formal del problema a la UPC, l'ús de diversos motors estudiats amb dades reals i analitzar els resultats per poder comparar-los.

8 Apèndix

8.1 Apèndix A: Relació d'algorismes i mètodes

Taula 6 Comparativa de Tècniques i Mètodes emprats en la literatura de UTT

Ref.	Any	Mètodes																
		Math	Graph	Cluster	Constraint	Meta-Heurístic										MultiC	HyperH	
		LP			CP	Poblacions			Local Search									
						GA	MM	ACO	SA	TS	VNS	GD	GRASP	ILS	HC			
[95]	1968																	
[96]	1973																	
[97]	1978																	
[98]	1979																	
[99]	1984																	
[100]	1985																	
[101]	1986																	
[102]	1986																	
[64]	1990																	
[103]	1991																	
[104]	1991																	
[68]	1991																	
[72]	1991																	
[105]	1992																	
[69]	1993																	
[106]	1993																	
[107]	1994																	
[31]	1994																	
[108]	1994																	
[109]	1994																	
[20]	1994																	
[71]	1994																	
[110]	1995																	
[111]	1995																	
[112]	1995																	
[113]	1995																	
[114]	1995																	
[115]	1995																	
[116]	1995																	
[117]	1995																	
[118]	1996																	
[73]	1996																	
[47]	1998																	

Ref.	Any	Mètodes															
		Math	Graph	Cluster	Constraint	Meta-Heuristic										MultiC	HyperH
		LP			CP	Poblacions			Local Search								
						GA	MM	ACO	SA	TS	VNS	GD	GRASP	ILS	HC		
[70]	1998																
[75]	1998																
[79]	1998																
[119]	2000																
[74]	2000																
[82]	2001																
[120]	2001																
[121]	2001																
[50]	2001																
[122]	2002																
[54]	2002																
[77]	2002																
[90]	2003																
[123]	2003																
[44]	2003																
[39]	2003																
[49]	2003																
[78]	2003																
[124]	2004																
[14]	2004																
[125]	2004																
[126]	2004																
[48]	2004																
[51]	2004																
[127]	2005																
[21]	2005																
[76]	2005																
[128]	2005																
[59]	2006																
[62]	2006																
[29]	2007																
[129]	2008																
[45]	2009																
[130]	2009																
[45]	2009																
[7]	2009																
[58]	2010																
[131]	2010																
[132]	2010																
[11]	2011																

Ref.	Any	Mètodes															
		Math	Graph	Cluster	Constraint	Meta-Heuristic										MultiC	HyperH
		LP			CP	Poblacions			Local Search								
						GA	MM	ACO	SA	TS	VNS	GD	GRASP	ILS	HC		
[133]	2011																
[134]	2011																
[135]	2011																
[136]	2012																

Alguns utilitzen diversos mètodes o tècniques alhora, poden ser considerats algorismes híbrids, una tendència habitual en els darrers anys.

8.2 Apèndix B: Referències Bibliogràfiques

Per a les referències bibliogràfiques faig servir la notació IEEE[137]

- [1] E. K. Burke, G. Kendall, M. Mısı́r, E. Özcan, E. K. Burke, G. Kendall, E. Özcan, G. Kendall, and M. Mısı́r, "Applications to timetabling," in *Handbook of Graph Theory, chapter 5.6*, 2004, pp. 445–474.
- [2] A. Schaerf, "A Survey of Automated Timetabling," *ARTIFICIAL INTELLIGENCE REVIEW*, vol. 13, pp. 87–127, 1995.
- [3] M. J. S. Iglesias, *Els Límits de la computació: indecidibilitat i NP-completesa*. Univ. Politèc. de Catalunya, 2010.
- [4] Willemen, RJ, "School timetable construction Algorithms and complexity," 2002.
- [5] E. Burke, K. Jackson, J. Kingston, and R. Weare, "Automated University Timetabling: The State of the Art," *The Computer Journal*, vol. 40, pp. 565–571, 1997.
- [6] Luca Di Gaspero and A. Schaerf, "The Second International Timetabling Competition (ITC–2007): Curriculum-based Course Timetabling (Track 3)," CiteSeerX, 2007.
- [7] M. J. Geiger, "Multi-criteria Curriculum-Based Course Timetabling—A Comparison of a Weighted Sum and a Reference Point Based Approach," in *Evolutionary Multi-Criterion Optimization*, vol. 5467, M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 290–304.
- [8] "University Course Timetabling & Student Sectioning System," presented at the ICAPS, 2007.
- [9] M. Dimopoulou and P. Miliotis, "An automated university course timetabling system developed in a distributed environment: A case study," *European Journal of Operational Research*, vol. 153, no. 1, pp. 136–147, Feb. 2004.
- [10] Y. Xiang and W. Zhang, "Distributed University Timetabling with Multiply Sectioned Constraint Networks," 2008, pp. 567–571.
- [11] P. WANGMAETEEKUL, "Using Distributed Agents to Create University Course Timetables Addressing Essential & Desirable Constraints and Fair Allocation of Resources," 2011. [Online]. Available: <http://etheses.dur.ac.uk/3602/>. [Accessed: 16-Jun-2012].
- [12] H.-J. Goltz and D. Matzke, "Combined Interactive and Automatic Timetabling," in *In Proc. PACLP99*, 1999.
- [13] T. Müller and R. Barták, "Interactive Timetabling: Concepts, Techniques, and Practical Results," in *PATAT 2002 — Proceedings of the 4th international conference on the Practice And Theory of Automated Timetabling*, 2002, pp. 58–72.
- [14] H. Cambazard, F. Demazeau, and N. Jussien, "Interactively solving school timetabling problems using extensions of constraint programming," in *PATAT 2004 — Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling*, 2004, pp. 107–124.
- [15] "International Timetabling Competition - Course Timetabling - Problem Model." [Online]. Available: http://www.cs.qub.ac.uk/itc2007/curriculumcourse/course_curriculum_index_files/problemmodel.htm. [Accessed: 04-Nov-2012].
- [16] G. Post, S. Ahmadi, S. Daskalaki, J. H. Kingston, J. Kyngas, C. Nurmi, and D. Ranson, "An XML format for benchmarks in High School Timetabling," *Annals of Operations Research*, vol. 194, no. 1, pp. 385–397, Feb. 2010.
- [17] "CTT - Instances." [Online]. Available: <http://satt.diegm.uniud.it/ctt/index.php?page=instances>. [Accessed: 02-Nov-2012].

-
- [18] E. K. Burke, D. G. Elliman, P. H. Ford, and R. F. Weare, "Examination Timetabling in British Universities - A Survey," in *The Practice and Theory of Automated Timetabling*, 1996, vol. 1153, pp. 76–92.
- [19] S. Petrovic and E. Burke, "University timetabling," in *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter 45, 2004.
- [20] D. Corne, P. Ross, and H.-L. Fang, "Fast Practical Evolutionary Timetabling," Dept. of Artificial Intelligence, University of Edinburgh, Research Report 708, 1994.
- [21] T. Müller, "Constraint-based Timetabling," 2005.
- [22] S. Piechowiak, J. Ma, and R. Mandiau, "An open interactive timetabling tool," in *Proceedings of the 5th international conference on Practice and Theory of Automated Timetabling*, Berlin, Heidelberg, 2005, pp. 34–50.
- [23] A. Bonutti, F. Cesco, L. Gaspero, and A. Schaerf, "Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results," *Annals of Operations Research*, vol. 194, no. 1, pp. 59–70, Feb. 2010.
- [24] "WATT -- The EURO Working group on Automated TimeTabling." [Online]. Available: <http://www.asap.cs.nott.ac.uk/external/watt/applications/indexEducational.shtml>. [Accessed: 17-Jun-2012].
- [25] "PATAT 2012 - 9th International Conference on the Practice and Theory of Automated Timetabling - Information." [Online]. Available: <http://www.patat2012.com/information.html>. [Accessed: 19-Jun-2012].
- [26] M. Carter and G. Laporte, "Recent developments in practical course timetabling," in *Practice and Theory of Automated Timetabling II*, vol. 1408, E. Burke and M. Carter, Eds. Springer Berlin / Heidelberg, 1998, pp. 3–19.
- [27] E. K. Burke and S. Petrovic, "Recent Research Directions in Automated Timetabling," *European Journal of Operational Research*, vol. 140, pp. 266–280, 2002.
- [28] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee, "A Survey of Search Methodologies and Automated System Development for Examination Timetabling," *Journal of Scheduling*, vol. 12, no. 1, pp. 55–89, 2009.
- [29] K. Schimmelpfeng and S. Helber, "Application of a real-world university-course timetabling model solved by integer programming," *OR Spectrum*, vol. 29, no. 4, pp. 783–803, 2007.
- [30] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, 1979.
- [31] E. Burke, "A University Timetabling System based on Graph Colouring and Constraint Manipulation," *Journal of Research on Computing in Education*, vol. 27, pp. 1–18, 1994.
- [32] E. K. Burke and J. Newall, "Solving Examination Timetabling Problems through Adaptation of Heuristic Orderings," *Annals of operations Research*, vol. 129, pp. 107–134, 2004.
- [33] H. Asmuni, E. K. Burke, and J. M. Garibaldi, "Fuzzy Multiple Heuristic Ordering for Course Timetabling," in *Proceedings of the 5th United Kingdom Workshop on Computational Intelligence (UKCI 2005)*, London, UK, 2005, pp. 302–309.
- [34] L. P. Reis and E. Oliveira, "A Language for Specifying Complete Timetabling Problems," in *PATAT2000 Proceedings, August 2000. Burke and Erben (Eds, 2000*, pp. 322–341.
- [35] O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. Stützle, "A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem," in *Practice and Theory of Automated Timetabling IV*, vol. 2740, E. Burke and P. De Causmaecker, Eds. Springer Berlin / Heidelberg, 2003, pp. 329–351.
- [36] L. D. Gaspero, *Local Search Techniques for Scheduling Problems: Algorithms and Software Tools*. 2003.
- [37] S. Abdullah and E. K. Burke, "A Multi-start Large Neighbourhood Search Approach with Local Search Methods for Examination Timetabling," in *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS 2006)*, Cumbria, UK, 2006, pp. 334–337.
- [38] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. D. Reidel Publishing Company, 1987.
-

- [39] E. Burke, Y. Bykov, J. Newall, and S. Petrovic, *A Time-Predefined Approach to Course Timetabling*. 2003.
- [40] E. K. Burke and Y. Bykov, "Solving Exam Timetabling Problems with the Flex-Deluge Algorithm," in *CONFERENCE PROCEEDINGS*, 2006, pp. 370–372.
- [41] S. Abdullah, E. K. Burke, and B. McCollum, "An Investigation of Variable Neighbourhood Search for University Course Timetabling," in *Proceedings of MISTA 2005: The 2nd Multidisciplinary Conference on Scheduling: Theory and Applications*, NY, USA, 2005, pp. 413–427.
- [42] E. K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic, and R. Qu, "Hybrid Variable Neighbourhood Approaches to University Exam Timetabling," *European Journal of Operational Research*, vol. 206, pp. 46–53, 2010.
- [43] K. Shaker and S. Abdullah, "Incorporating great deluge approach with kempe chain neighbourhood structure for curriculum-based course timetabling problems," in *2nd Conference on Data Mining and Optimization, 2009. DMO '09*, 2009, pp. 149–153.
- [44] S. Casey and J. Thompson, "GRASPing the Examination Scheduling Problem," in *Practice and Theory of Automated Timetabling IV*, E. Burke and P. D. Causmaecker, Eds. Springer Berlin Heidelberg, 2003, pp. 232–244.
- [45] E. K. Burke, R. Qu, and A. Soghier, "Adaptive Selection of Heuristics within a GRASP for Exam Timetabling Problems," in *The 4th Multidisciplinary International Scheduling: Theory and Applications 2009*, Dublin, Ireland, 2009, pp. 409–423.
- [46] E. K. Burke, T. Curtois, M. Hyde, G. Kendall, G. Ochoa, S. Petrovic, J. A. Vazquez-Rodriguez, and M. Gendreau, "Iterated Local Search vs. Hyper-heuristics: Towards General-purpose Search Algorithms," in *IEEE Congress on Evolutionary Computation (CEC 2010)*, Barcelona, Spain, 2010, pp. 3073–3080.
- [47] P. Ross and D. Corne, "Comparing Genetic Algorithms, Simulated Annealing and Stochastic Hillclimbing on Several Real Timetabling Problems," in *Proceedings of the AISB Workshop on Evolutionary Computing, Sheffield, England, 1995*.
- [48] D. A. Pelta and N. Krasnogor, "Recent Advances in Memetic Algorithms," Springer, 2004.
- [49] K. Socha, M. Sampels, and M. Manfrin, "Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art," in *Applications of Evolutionary Computing*, vol. 2611, S. Cagnoni, C. Johnson, J. Cardalda, E. Marchiori, D. Corne, J.-A. Meyer, J. Gottlieb, M. Middendorf, A. Guillot, G. Raidl, and E. Hart, Eds. Springer Berlin / Heidelberg, 2003, pp. 334–345.
- [50] E. K. Burke, Y. Bykov, and S. Petrovic, "A Multicriteria Approach to Examination Timetabling," in *Practice and Theory of Automated Timetabling III: Selected Revised papers from the 3rd international conference*, Konstanz, Germany, 2001, vol. 2079, pp. 118–131.
- [51] G. Kendall and N. M. Hussin, "Tabu Search Hyper-Heuristic Approach to the Examination Timetabling Problem at University Technology MARA," in *Proceedings of the 5th international conference on the Practice and Theory of Automated Timetabling (PATAT)*, Pittsburgh, USA, 2004, pp. 199–217.
- [52] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu, "A Graph-Based Hyper-Heuristic for Educational Timetabling Problems," *European Journal of Operational Research*, vol. 176, pp. 177–192, 2007.
- [53] E. K. Burke, B. L. MacCarthy, S. Petrovic, and R. Qu, "Case-Based Reasoning in Course Timetabling: An Attribute Graph Approach," in *Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR-2001)*, Vancouver, Canada, 2001, vol. 2080, pp. 90–104.
- [54] E. K. Burke, B. L. MacCarthy, S. Petrovic, and R. Qu, "Knowledge Discovery in Hyper-Heuristic Using Case-Based Reasoning on Course Timetabling," in *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2002)*, Gent, Belgium, 2002, vol. 2740, pp. 276–286.
- [55] M. W. Carter, "A Decomposition Algorithm for Practical Timetabling Problems," Industrial Engineering, University of Toronto, Working Paper 83-06, Apr. 1983.

- [56] R. Qu and E. K. Burke, "Adaptive Decomposition and Construction for Examination Timetabling Problems," in *Multidisciplinary International Scheduling: Theory and Applications (MISTA 2007)*, Paris, France, 2007, pp. 418–425.
- [57] R. Qu, F. He, and E. K. Burke, "Hybridizing Integer Programming Models with an Adaptive Decomposition Approach for Exam Timetabling Problems," in *The 4th Multidisciplinary International Scheduling: Theory and Applications 2009*, Dublin, Ireland, 2009, pp. 435–446.
- [58] E. K. Burke, J. M. cek, A. J. Parkes, and H. Rudov' a, "Decomposition, Reformulation, and Diving in University Course Timetabling," *Comput. Oper. Res.*, vol. 37, no. 1, pp. 582–597, 2010.
- [59] M. Oprea, *Multi-Agent System for University Course Timetable Scheduling*. 2006.
- [60] "Metaheuristics Network." [Online]. Available: <http://www.metaheuristics.org/>. [Accessed: 04-Nov-2012].
- [61] R. Lewis, "A survey of metaheuristic-based techniques for University Timetabling problems," *OR Spectrum*, vol. 30, no. 1, pp. 167–190, 2008.
- [62] M. Chiarandini, M. Birattari, K. Socha, and O. Rossi-doria, *An effective hybrid algorithm for university course timetabling*. 2006.
- [63] B. McCollum, A. Schaerf, B. Paechter, P. McMullan, R. Lewis, A. J. Parkes, L. D. Gaspero, R. Qu, and E. K. Burke, "Setting the Research Agenda in Automated Timetabling: The Second International Timetabling Competition," *INFORMS JOURNAL ON COMPUTING*, vol. 22, no. 1, pp. 120–130, 2010.
- [64] A. Colorni, M. Dorigo, and V. Maniezzo, "A Genetic Algorithm to Solve the Timetable Problem," Politecnico di Milano, 90-060, 1990.
- [65] A. Colorni, M. Dorigo, and V. Maniezzo, "Genetic Algorithms and Highly Constrained Problems: The Timetable Case," in *Proceedings of the First International Workshop on Parallel Problem Solving from Nature, Dortmund, Germany. Lecture Notes in Computer Science 496*, Springer-Verlag, 1991, pp. 55–59.
- [66] A. Colorni, M. Dorigo, and V. Maniezzo, "Metaheuristics for High School Timetabling," *Computational Optimization and Applications*, vol. 9, no. 3, pp. 275–298, 1998.
- [67] M. P. Carrasco and M. V. Pato, "A Multiobjective Genetic Algorithm for the Class/Teacher Timetabling Problem," in *Practice and Theory of Automated Timetabling III*, E. Burke and W. Erben, Eds. Springer Berlin Heidelberg, 2001, pp. 3–17.
- [68] D. Abramson, "Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms," *Management Science*, vol. 37, pp. 98–113, 1991.
- [69] D. Abramson and H. Dang, "School Timetables: A Case Study Using Simulated Annealing," in *Applied Simulated Annealing, Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, 1993, pp. 104–124.
- [70] M. A. S. Elmohamed, P. Coddington, and G. Fox, "A comparison of annealing techniques for academic course scheduling," in *Practice and Theory of Automated Timetabling II*, E. Burke and M. Carter, Eds. Springer Berlin Heidelberg, 1998, pp. 92–112.
- [71] D. Costa, "A Tabu Search Algorithm for Computing an Operational Timetable," *European Journal of Operational Research*, vol. 76, pp. 98–110, 1994.
- [72] A. Hertz, "Tabu Search for Large Scale Timetabling Problems," *European Journal of Operational Research*, vol. 54, pp. 39–47, 1991.
- [73] A. Schaerf, "Tabu Search Techniques for Large High-School Timetabling Problems," in *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 1996, pp. 363–368.
- [74] L. D. Gaspero and A. Schaerf, *Tabu Search Techniques for Examination Timetabling*. 2000.
- [75] Thompson J.M. and Dowsland K.A., "A robust simulated annealing based examination timetabling system," *Computers and Operations Research*, vol. 25, no. 7, pp. 637–648, 1998.
- [76] H. Arntzen and A. Løkketangen, "A Tabu Search Heuristic for a University Timetabling Problem," in *Metaheuristics: Progress as Real Problem Solvers*, T. Ibaraki, K. Nonobe, and M. Yagiura, Eds. Springer US, 2005, pp. 65–85.
- [77] K. Socha, J. Knowles, and M. Sampels, "A MAX-MIN Ant System for the University Course Timetabling Problem," in *in Proceedings of the 3rd International Workshop on Ant Algorithm, ANTS 2002, Lecture Notes in Computer Science*, 2002, pp. 1–13.

- [78] L. T. G. Merlot, N. Boland, B. D. Hughes, and P. J. Stuckey, "A Hybrid Algorithm for the Examination Timetabling Problem," in *Practice and Theory of Automated Timetabling IV*, E. Burke and P. D. Causmaecker, Eds. Springer Berlin Heidelberg, 2003, pp. 207–231.
- [79] B. Paechter, R. C. Rankin, A. Cumming, and T. C. Fogarty, "Timetabling the classes of an entire university with an evolutionary algorithm," in *Parallel Problem Solving from Nature — PPSN V*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, 1998, pp. 865–874.
- [80] B. J. Oates, *Researching Information Systems And Computing*. SAGE, 2006.
- [81] A. Lim, J. Ang, W. Ho, and W. Oon, "UTTSExam: A Campus-Wide University Exam-Timetabling System," *Artificial Intelligence*, pp. 838–844, 2002.
- [82] M. Dimopoulou and P. Miliotis, "Implementation of a university course and examination timetabling system," *European Journal of Operational Research*, vol. 130, no. 1, pp. 202–213, Apr. 2001.
- [83] Peter Karich, "gstpl - Timetabling Software Survey." [Online]. Available: <http://gstpl.wikispaces.com/Timetabling+Software+Survey>. [Accessed: 17-Jun-2012].
- [84] T. Müller, K. Murray, and H. Rudová, *System Demonstration of Interactive Course Timetabling*. .
- [85] T. Müller and H. Rudova, *Minimal Perturbation Problem in Course Timetabling*. .
- [86] "UniTime | University Timetabling." [Online]. Available: <http://www.unitime.org/>. [Accessed: 19-Jun-2012].
- [87] "Drools Planner - JBoss Community." [Online]. Available: <http://www.jboss.org/drools/drools-planner.html>. [Accessed: 17-Jan-2013].
- [88] "Rete algorithm - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/Rete_algorithm. [Accessed: 17-Jan-2013].
- [89] Liviu Lalescu and Costin Badica, "AN EVOLUTIONARY APPROACH FOR SCHOOL TIMETABLING," University of Craiova, Faculty of Control, Computers and Electronics, 2002.
- [90] Liviu Lalescu and Costin Badica, "TIMETABLING EXPERIMENTS USING GENETIC ALGORITHMS," University of Craiova, Faculty of Control, Computers and Electronics, 2003.
- [91] "FET - Free Timetabling Software." [Online]. Available: <http://lalescu.ro/liviu/fet/>. [Accessed: 10-Jun-2012].
- [92] "#1 Scheduling Software. | College | University | School | Medical | Employees | Conference." [Online]. Available: <http://schedulingstudio.com/>. [Accessed: 17-Jun-2012].
- [93] "The Timetabling and Scheduling Specialists» EventMAP Limited» Home." [Online]. Available: <http://www.eventmap-uk.com/site/>. [Accessed: 17-Jan-2013].
- [94] "ASAP | Automated Scheduling, Optimisation and Planning Research Group." [Online]. Available: <http://www.asap.cs.nott.ac.uk/>. [Accessed: 17-Jan-2013].
- [95] I. Oliver, "Tree-Searching School Timetables," *The Australian Computer Journal*, vol. 1, pp. 153–157, 1968.
- [96] E. A. Akkoyunlu, "A Linear Algorithm for Computing the Optimum University Timetable," *The Computer Journal*, vol. 16, pp. 347–350, 1973.
- [97] P. W. Chan and G. M. White, "An Algorithm for Examination Scheduling: Theory and Practice," in *Proceedings of the Canadian Information Processing Society Session, Canadian Computer Conf., Edmonton, 1978*, pp. 244–249.
- [98] G. M. White and P. W. Chan, "Towards the Construction of Optimal Examination Schedules," *INFOR*, vol. 17, no. 3, pp. 219–229, 1979.
- [99] A. Tripathy, "School Timetabling - A Case in Large Binary Integer Linear Programming," *Management Science*, vol. 30, pp. 1473–1489, 1984.
- [100] D. de Werra, "An Introduction to Timetabling," *European Journal of Operational Research*, vol. 19, pp. 151–162, 1985.
- [101] M. W. Carter, "A Survey of Practical Applications of Examination Timetabling Algorithms," *Operations Research*, vol. 34, pp. 193–202, 1986.
- [102] K. Gosselin and M. Truchon, "Allocation of Classrooms by Linear Programming," *Journal of the Operational Research Society*, vol. 37, pp. 561–569, 1986.

-
- [103] V. Lotfi and R. Cervený, "A Final Exam Scheduling Package," *Journal of the Operational Research Society*, vol. 42, pp. 205–216, 1991.
- [104] D. Abramson and J. Abela, "A Parallel Genetic Algorithm for Solving the School Timetabling Problem," Division of Information Technology, Commonwealth Scientific and Industrial Scientific Organisation (CSIRO), Australia, Apr. 1991.
- [105] S. E. Ling, "Integrating Genetic Algorithms with a Prolog Assignment Program as a Hybrid Solution for a Polytechnic Timetable Problem," *Parallel Problem Solving from Nature*, vol. 2, pp. 321–329, 1992.
- [106] P. Dige, C. Lund, and H. F. Ravn, "Timetabling by Simulated Annealing," in *Applied Simulated Annealing, Lecture Notes in Economics and Mathematical Systems 396*, Springer-Verlag, 1993, pp. 151–174.
- [107] F. Azevedo and P. Barahona, "Timetabling in Constraint Logic Programming," in *Proceedings of World Congress on Expert Systems'94*, 1994.
- [108] E. K. Burke, D. G. Elliman, and R. F. Weare, "A Genetic Algorithm based University Timetabling System," in *Proceedings of the 2nd East-West International Conference on Computer Technologies in Education*, Crimea, Ukraine, 1994, vol. 1, pp. 35–40.
- [109] E. K. Burke, D. G. Elliman, and R. F. Weare, "A Genetic Algorithm for University Timetabling," in *Proceedings of the AISB workshop on Evolutionary Computing (University of Leeds, UK, 11th-13th April 1994)*, 1994.
- [110] P. Boizumault, Y. Delon, and L. Peridy, "Constraint Logic Programming for Examination Timetabling," *Journal of Logic Programming*, vol. 19, pp. 1–17, 1995.
- [111] C. Cheng, L. Kang, N. Leun, and G. M. White, "Investigations of a constraint logic programming approach to university timetabling," in *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling (ICPTAT '95)*, 1995, pp. 82–93.
- [112] G. Lajos, "Complete university modular timetabling using constraint logic programming," in *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling (ICPTAT '95)*, 1995, pp. 364–375.
- [113] C. Gueret, N. Jussien, P. Boizumault, and C. Prins, "Building university timetables using constraint logic programming," in *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling (ICPTAT '95)*, 1995, pp. 393–408.
- [114] E. K. Burke, D. G. Elliman, and R. F. Weare, "A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems," in *Proceedings of the 6th International Conference on Genetic Algorithms (ICGA'95)*, Pittsburgh, USA, 1995, pp. 605–610.
- [115] W. Erben and J. Keppler, "A genetic algorithm solving an weekly course-timetabling problem," in *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling (ICPTAT '95)*, 1995, pp. 21–32.
- [116] H. Luchian, C. Ungureanasu, B. Paechter, and M. Petruic, "Fine-tuning a genetic algorithm for the general timetable problem," in *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling (ICPTAT '95)*, 1995, pp. 435–442.
- [117] R. C. Rankin, "Memetic timetabling in practice," in *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling (ICPTAT '95)*, 1995, pp. 45–56.
- [118] E. K. Burke, J. Newall, and R. F. Weare, "A Memetic Algorithm for University Exam Timetabling," in *The Practice and Theory of Automated Timetabling*, 1996, vol. 1153, pp. 241–250.
- [119] S. Abdennadher and M. Marte, "University course timetabling using constraint handling rules," *Journal of Applied Artificial Intelligence*, vol. 14, p. 2000.
- [120] "Zervoudakis y Stamatopoulos - 2001 - A Generic Object-Oriented Constraint-Based Model f.pdf." .
- [121] E. K. Burke, P. Cowling, and R. Keuthen, "Effective Local and Guided Variable Neighborhood Search Methods for the Asymmetric Travelling Salesman Problem," in *Proceedings of the First European Workshop on Evolutionary Computation in Combinatorial Optimization (EvoCOP2001)*, Lake Como (Milan), Italy, 2001, vol. 2037, pp. 203–212.
-

-
- [122] M. Marte, "Models and algorithms for school timetabling: a constraint programming approach /," 2002.
- [123] E. K. Burke, G. Kendall, and E. Soubeiga, "A Tabu Search Hyper-Heuristic for Timetabling and Rostering," *Journal of Heuristics*, vol. 9, no. 6, pp. 451–470, Dec. 2003.
- [124] E. K. Burke, D. de Werra, and J. Kingston, "Handbook of Graph Theory," J. Gross and J. Yellen, Eds. Chapman Hall/CRC Press, 2004, pp. 445–474.
- [125] "An integer programming formulation for a case study.pdf." .
- [126] E. K. Burke, D. De Werra, and J. H. Kingston, "Applications to Timetabling," 2004, pp. 445–473.
- [127] S. Daskalaki and T. Birbas, "Efficient solutions for a university timetabling problem through integer programming," *European Journal of Operational Research*, vol. 160, no. 1, pp. 106–120, Jan. 2005.
- [128] R. Santiago-Mozos, S. Salcedo-Sanz, M. DePrado-Cumplido, and C. Bousoño-Calzón, "A two-phase heuristic evolutionary algorithm for personalizing course timetables: a case study in a Spanish university," *Computers & Operations Research*, vol. 32, no. 7, pp. 1761–1776, Jul. 2005.
- [129] E. K. Burke, J. M. cek, A. J. Parkes, and H. Rudov' a, "Penalising Patterns in Timetables: Novel Integer Programming Formulations," in *Operations Research Proceedings: Selected Papers of the of the Annual International Conference of the German Operations Research Society (GOR)*, Saarbrücken, Germany, 2007, pp. 409–414.
- [130] Z. Bratković, T. Herman, V. Omrčen, M. Čupić, and D. Jakobović, "University Course Timetabling with Genetic Algorithm: A Laboratory Exercises Case Study," *Evolutionary Computation in Combinatorial Optimization*, vol. 5482, pp. 240–251, 2009.
- [131] Z. Lü and J.-K. Hao, "Adaptive Tabu Search for course timetabling," *European Journal of Operational Research*, vol. 200, no. 1, pp. 235–244, Jan. 2010.
- [132] D. Hadjidj and H. Drias, "Grasp and Guided Local Search for the examination timetabling problem," *International Journal of Artificial Intelligence and Soft Computing*, vol. 2, no. 1, pp. 103–114, Jan. 2010.
- [133] E. K. Burke, J. Mare\v cek, A. J. Parkes, and H. Rudov' a, "A Branch-and-cut Procedure for the Udine Course Timetabling Problem," *Annals of Operations Research*, vol. 194, pp. 71–87, 2011.
- [134] N. R. Sabar, M. Ayob, R. Qu, and G. Kendall, "A Graph Coloring Constructive Hyper-Heuristic for Examination Timetabling Problems," *Applied Intelligence*, 2011.
- [135] S. Yang and S. N. Jat, "Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 1, pp. 93–106, Jan. 2011.
- [136] G. H. G. Fonseca, S. S. Brito, and H. G. Santos, "A Simulated Annealing Based Approach to the High School Timetabling Problem," in *Intelligent Data Engineering and Automated Learning - IDEAL 2012*, vol. 7435, H. Yin, J. A. F. Costa, and G. Barreto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 540–549.
- [137] "IEEE - Author Digital Tool Box." [Online]. Available: http://www.ieee.org/publications_standards/publications/authors/authors_journals.html. [Accessed: 19-Jun-2012].