

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Elettronica

Tesi di Laurea

**Analisi e metodi per l'allocazione
automatica dei laboratori e la
pianificazione dell'orario accademico**



Relatori

prof. Marco Mezzalama

prof.ssa Elisa Ficarra

Corelatore:

prof. Federico Della Croce

Candidata

Beatrice Scarselli

A.A. 2017-2018

Sommario

La pianificazione dell'orario accademico presso il *Politecnico di Torino* è un'operazione che, ancora oggi, viene svolta manualmente e richiede un notevole impegno di tempo e di energie da parte del personale docente e del personale tecnico amministrativo. Purtroppo però i risultati ottenuti spesso non sono ottimi e non contemplano un'ottimizzazione delle risorse di ateneo; automatizzando il processo si possono invece raggiungere soluzioni di alta qualità e in tempi ridotti.

In questa tesi verrà descritta una soluzione per automatizzare l'assegnazione dei laboratori di informatica dell'Area IT (Information Technology) ai corsi di Laurea Triennale e Magistrale di Ingegneria e di Architettura, fornendo un punto di partenza per sopperire al fatto che ad oggi non esista un processo per automatizzare e centralizzare la pianificazione dell'orario. Inoltre verranno indicati i pregi e i difetti dell'applicativo web in dotazione per la pubblicazione dell'orario e ne verranno illustrati i possibili miglioramenti a seguito della valutazione da parte del personale coinvolto nella pianificazione, e della consultazione dei software disponibili in rete, sia commerciali che open source.

Dagli anni sessanta un'intera comunità scientifica, nell'ambito della Ricerca Operativa e dell'Intelligenza Artificiale, studia il problema dell'orario (in inglese *timetabling problem*) proponendo soluzioni ed applicazioni per l'assegnazione automatica delle risorse e la generazione automatica di orari e turni in qualsiasi settore di attività: sanità, trasporti, istituzioni scolastiche e universitarie; questi studi vengono presentati e discussi, ogni due anni, alla Conferenza Internazionale PATAT (*Practice and Theory of Automated Timetabling*). Una definizione formale del timetabling problem ci viene data da Schaerf, ovvero: "*pianificare una sequenza di lezioni tra insegnanti e studenti in un prefissato periodo di tempo (tipicamente una settimana), soddisfacendo un set di vincoli di vario tipo*". Ma la difficoltà principale è la definizione di un *problema standard* che racchiuda un set di vincoli che lo renda portatile e adattabile alle esigenze di enti differenti. Infatti, allo stato attuale, un programma scritto per creare l'orario di un'università difficilmente può essere utilizzato in un'altra università. Per raggiungere questo scopo tuttavia non si è deciso di testare i software commerciali (per esempio *Hyperplanning*) ed open source (per esempio *UniTime*) trovati in rete perché presentano un insieme di dati, di vincoli e di funzioni che non possono soddisfare tutte le esigenze dei corsi del nostro Ateneo, ma possono solo dare indicazioni e suggerimenti su come scrivere ed implementare il nostro modello.

Anche se un primo tentativo di implementazione di orario automatizzato fu fatto da un tesista nel 2011 per la ex IV Facoltà del Politecnico di Torino, tuttavia in questi anni non si è deciso di estendere il risultato ottenuto per l'intero Ateneo, perché il *Collegio di*

Ingegneria Gestionale presenta un caso semplificato rispetto agli altri Collegi che hanno un orario eterogeneo, e quindi creare un orario per tutti i corsi richiede uno studio molto dettagliato e un procedimento per gradi.

Attualmente la pianificazione dell'orario viene svolta al *Politecnico di Torino* da una *Commissione Orari*, suddivisa in 14 *Collegi di Corsi di Studio* di Ingegneria e di Architettura; per ogni Collegio alcuni docenti e personale amministrativo vengono incaricati come *Referenti* per la creazione dell'orario. La pianificazione è suddivisa in tre fasi: la prima fase è legata alla creazione vera e propria dell'orario dei corsi per ogni semestre, la seconda è legata all'assegnazione delle risorse alle attività didattiche e la terza fase corrisponde alla pubblicazione dell'orario sull'applicativo web.

Durante la prima fase, tutti i docenti sono tenuti (ma non obbligati) a compilare una scheda online per fornire informazioni circa i requisiti del corso in termini di attrezzature, risorse informatiche e di personale, numerosità, incompatibilità d'orario con altri corsi e richieste personali. In seguito il Referente Orario di ogni Collegio, mediante una procedura automatica raccoglie tutti i dati e costruisce una "bozza" orario. Purtroppo però alcune informazioni tecnico-gestionali fondamentali per la costruzione dell'orario non sono fruibili consultando un'unica fonte o non sono distribuite in modo completo al Referente e quindi il Referente deve consultare altre aree della Didattica e ricontattare singolarmente i docenti per ottenere quelle informazioni necessarie per costruire l'orario. Inoltre il Referente deve tenere conto della concomitanza di diversi fattori, quali per esempio: il docente può essere titolare di più corsi; i corsi hanno più docenti che possono essere impegnati in altri corsi differenti fra loro; molti di questi corsi possono essere condivisi tra diversi piani di studi secondo combinazioni differenti (ad esempio corso A offerto a piano di studi 1 e 2, corso B offerto a piano di studi 1, 3 e 4, corso C offerto a piano di studi 2 e 3, corso D offerto a piano di studi 1 e 4); gli studenti non devono avere conflitti di orari tra corsi presenti nel loro piano di studi; ogni corso deve essere collocato in una struttura adeguata (aula o laboratorio) in funzione della capienza e delle attrezzature disponibili (software, lavagne, videoregistrazioni). Purtroppo anche queste informazioni spesso non sono ben chiare e non si possono stabilire linee guida ben definite perché ogni Collegio presenta regole diverse.

Attualmente, la seconda fase corrisponde all'assegnazione delle risorse, operazione svolta manualmente sempre dal Referente Orario in collaborazione con i Referenti delle risorse (aule e laboratori). Il problema fondamentale, evidenziato in questi ultimi anni, è che il numero delle risorse a disposizione è scarsamente sufficiente per accogliere tutti gli studenti iscritti al Politecnico di Torino. A questo si aggiunge il problema della decentralizzazione nell'assegnazione delle aule, per quanto riguarda la sede centrale dove: un parco aule è destinato solo ai corsi del primo anno in comune, uno è destinato al *Collegio di Ingegneria Gestionale*, un altro al *Collegio di Ingegneria Informatica, Cinema e Meccatronica (ICM)* e al *Collegio di Elettronica, Telecomunicazioni e Fisica (ETF)* e il restante a tutti gli altri *Collegi*, suddivisione che non tiene conto del crescente aumento della numerosità dei corsi degli anni successivi al primo. L'assegnazione dei laboratori è al momento l'unica operazione che, sebbene sia svolta in modo manuale, è già centralizzata poiché non dipende da altre attività.

Si è quindi pensato di cominciare con l'automatizzazione dell'allocazione dei laboratori di informatica dell'Area IT condivisi fra tutti i Collegi. Il nostro problema reale può

essere descritto in modo semplificato, nell'ambito della Ricerca operativa, da un insieme di relazioni matematiche che costituiscono il nostro modello matematico. In particolare è un modello di *Programmazione lineare intera booleana* (PLI 0/1) dove viene definita una *variabile decisionale* x che può assumere valore 1 se l'evento si è verificato, 0 altrimenti. Quindi nel nostro caso occorre verificare se il *corso è assegnato al laboratorio oppure no in un determinato modulo orario* (stabilito dai Referenti orari); come modulo orario si considera un'ora e trenta minuti. Per costruire un modello matematico si devono definire i *vincoli* che, tramite disequazioni o equazioni matematiche, stabiliscono i valori assunti dalle variabili, come dettagliato nella tesi. Queste variabili sono state pesate in modo da attribuire priorità differenti ai vari corsi. Questo è stato fatto in virtù del fatto che alcuni piani di studio in alcuni anni presentano maggiori criticità di altri avendo vincoli di orario più stringenti. I relativi corsi quindi devono avere la precedenza nell'assegnazione delle risorse. Infine viene definita come *funzione obiettivo*, la *somma dei corsi assegnati ai laboratori* da massimizzare. Per le nostre istanze sono stati presi in esame tutti i corsi della Laurea Triennale e Magistrale che hanno richiesto l'utilizzo dei laboratori nell'anno accademico 2017 – 2018 e nel primo periodo didattico perché più critico, e solo i laboratori della sede Centrale e Cittadella perché si verificano maggiormente le sovrapposizioni delle richieste.

In letteratura c'è una certa confusione su quanto sia difficile il problema dell'assegnazione delle aule, perché la difficoltà non dipende solo dall'obiettivo o dalla dimensione del problema ma anche dalle preferenze dei corsi per le aule (per esempio, nel nostro caso, le tre squadre dei Corsi di Informatica del primo anno devono essere allocate in moduli orari consecutivi e nello stesso laboratorio); per molte università questo problema è facile per altre risulta quasi impossibile. La *complessità computazionale* del problema è ancora oggetto di studio perché per alcuni autori è possibile sviluppare un *algoritmo polinomiale*, per altri è possibile dimostrare che il problema generale è *NP-completo*; quindi per alcuni si risolve in tempo polinomiale, per altri invece si sviluppano approcci euristici. Ma lo scopo della nostra tesi non è stabilire quale algoritmo debba essere implementato per risolvere il nostro problema, ma è definire un modello matematico generale come punto di partenza per la costruzione di un modello completo, che includa tutti i parametri e i vincoli dell'orario dei corsi del nostro Ateneo.

Per verificare se esiste o meno una soluzione al nostro problema si è utilizzato il software risolutore *Xpress-IVE*, che implementa una serie di algoritmi (come per esempio il metodo *Branch and Bound* o il *Branch and cut*) scrivendo il nostro modello matematico nel linguaggio di programmazione *Mosel*. Si è quindi dimostrato che non è possibile assegnare i laboratori in base alle richieste fornite dai Referenti, ma che si può solo massimizzare l'assegnazione. Solo utilizzando orari di "riserva" per i corsi che sono rimasti "fuori", si possono allocare tutti i corsi. Sono stati assegnati 192 su 202 moduli orari richiesti, conoscendo anche quali corsi sono stati allocati e quali no e i laboratori e le fasce orarie rimaste libere; infine si è completata l'assegnazione, utilizzando gli orari di riserva dei corsi rimasti "fuori". I risultati sono stati raggiunti in pochissimo tempo, dell'ordine del secondo.

L'obiettivo futuro è estendere questa procedura automatizzata anche alle aule per "centralizzare" l'assegnazione di tutti i corsi a tutte le risorse disponibili, consentendo alle

persone coinvolte di riacquistare del tempo da dedicare ad altre attività lavorative. Inoltre visto che la nuova squadra di governo del Politecnico di Torino intende automatizzare l'intero processo, proprio sfruttando gli algoritmi risolutivi della Ricerca operativa, si spera che questa tesi possa essere consultata per conoscere lo stato attuale, il lavoro che c'è "dietro le quinte" e la complessità del problema.

Ringraziamenti

Ringrazio *Cineca* per avermi fatto partecipare ad un workshop sull'evoluzione dell'applicativo *University Planner* e per avermi dato la disponibilità di un ambiente web per effettuare delle prove. Ringrazio il professor Andrea Schaerf, del Dipartimento Politecnico di Ingegneria e Architettura *dell'Università di Udine*, per avermi concesso l'utilizzo dei suoi articoli. Ringrazio il ricercatore Fabio Salassa, del Dipartimento di Ingegneria Gestionale e della Produzione del *Politecnico di Torino* per avermi guidata, con pazienza e disponibilità, nella costruzione del mio primo modello matematico di Ricerca Operativa. Ringrazio i docenti relatori di questa *tesi* che, senza riserva, hanno accettato questo argomento come lavoro da svolgere per l'ultima fase del mio lungo percorso di studi.

Indice

1	Introduzione	1
1.1	Oggetto della tesi	1
1.2	Struttura della tesi	2
1.3	Obiettivi e utilità della tesi	4
2	Stato attuale della pianificazione dell'orario accademico	5
2.1	Generazione dell'orario accademico	5
2.1.1	Problematiche nella generazione dell'orario accademico	7
2.2	Assegnazione delle risorse di Ateneo	12
2.2.1	Problematiche nell'assegnazione delle risorse di Ateneo	13
2.3	Applicativo web per la pubblicazione dell'orario	15
2.3.1	Problematiche nell'applicativo web	19
3	Software disponibili per la generazione automatica dell'orario	21
3.1	Software per l'orario scolastico	21
3.2	Software per l'orario universitario	27
3.3	Orario accademico in altri atenei italiani	34
4	Descrizione del problema e sviluppo del modello	37
4.1	Perché la Ricerca Operativa?	37
4.1.1	Timetabling problem	38
4.1.2	Altre possibili soluzioni al timetabling problem	42
4.2	Dati del problema e i vincoli	46
4.2.1	Le risorse	47
4.2.2	L'Orario	48
4.2.3	I corsi	48
4.2.4	I vincoli	50
4.3	Costruzione del modello matematico	51
4.3.1	Un po' di teoria	51
4.3.2	Variabile decisionale	54
4.3.3	Vincoli	55
4.3.4	Vincoli aggiuntivi	57
4.3.5	Funzione obiettivo	58
4.4	Complessità computazionale del nostro problema	59

4.4.1	Algoritmo polinomiale	59
4.4.2	La classe P e la classe NP	59
4.4.3	Problemi NP-completi e NP-hard	60
5	Implementazione del modello con il solver	61
5.1	Metodo Branch and Bound	61
5.2	Dati di input	62
5.3	Prima istanza	63
5.4	Seconda istanza	66
5.5	Terza istanza	70
6	Possibili miglioramenti sull'applicativo web di Ateneo	71
6.1	Miglioramenti oggettivi	71
6.1.1	Modulo di predisposizione orari e file dei dati estrapolati	71
6.1.2	Software Gestione Orari	74
6.2	Miglioramenti soggettivi	76
6.2.1	Proposte suggerite da altri software	76
6.2.2	Eventuali modifiche dell'applicativo web	79
7	Conclusioni	83
8	Appendici	85
8.1	Appendice A: listato del modello in linguaggio <i>Mosel</i>	85
8.2	Appendice B: risultato ottenuto	85
	Bibliografia	87

Capitolo 1

Introduzione

1.1 Oggetto della tesi

La pianificazione dell'Orario accademico presso il *Politecnico di Torino* è un'operazione che coinvolge personale docente e personale tecnico amministrativo che dedicano molto tempo e molte energie per poter accontentare tutti i fruitori finali dell'orario, ovvero gli studenti, i docenti e anche il personale di supporto. Purtroppo quest'operazione viene svolta manualmente esattamente nello stesso modo da diversi anni.

È suddivisa in tre parti: una prima parte riguarda la generazione vera e propria dell'orario accademico per lo svolgimento di tutte le attività didattiche, quali lezioni ed esercitazioni in aula e in laboratorio; una seconda parte legata all'assegnazione delle risorse dove svolgere queste attività; una terza parte, solo operativa, riguarda la pubblicazione dell'orario su un applicativo web.

Generare un orario accademico significa collocare nel tempo, generalmente una settimana, le attività didattiche che vengono svolte dai docenti e vengono seguite dagli studenti, e questo comporta la concomitanza di diversi fattori, che al *Politecnico di Torino* sono: 1) il docente può essere titolare di più corsi ma può svolgerne uno solo per volta, 2) i corsi hanno più docenti che possono essere impegnati in altri corsi differenti fra loro, 3) ogni corso deve essere collocato nella struttura adeguata (aula o laboratorio) in funzione della capienza e delle attrezzature disponibili (software, lavagne, videoregistrazioni), 4) gli studenti non devono avere conflitti di orari tra corsi presenti nel loro piano di studi, 5) diversi corsi sono offerti a studenti che afferiscono a piani di studi differenti, quindi non sono corsi specifici di un unico piano di studi, 6) molti dei corsi citati al punto 5) non sono condivisi tra gli stessi molteplici piani di studio, ma possono essere condivisi tra diversi piani di studio secondo combinazioni differenti (ad es. corso A offerto a piano di studi 1 e 2, corso B offerto a piano di studi 1,3 e 4, corso C offerto a piano di studi 2 e 3, corso D offerto a piano di studi 1 e 4).

L'assegnazione delle risorse viene fatta manualmente, ancora oggi, come quando esistevano le Facoltà: l'Amministrazione centrale ha assegnato un parco aule destinate solo ed esclusivamente a tutti i corsi del Primo anno di Ingegneria, un parco aule destinate al *Collegio di Ingegneria Gestionale* (la ex *IV Facoltà*), un parco aule al *Collegio di Ingegneria Informatica, Cinema e Meccatronica (ICM)* e al *Collegio di Elettronica, Telecomunicazioni*

e *Fisica (ETF)* (la ex *III Facoltà*), e il restante a tutti gli altri Collegi (la ex *I Facoltà*). Quindi non esiste ancora un processo per risolvere il problema in modo centralizzato: ottimizzare e automatizzare la generazione dell'Orario accademico di tutta la didattica per tutti gli studenti del *Politecnico di Torino* considerando tutte le risorse a disposizione.

La pubblicazione dell'Orario accademico del *Politecnico di Torino* viene svolta utilizzando un applicativo web [3] creato dall'Ufficio Sviluppo e gestione applicazioni dell'*Area Information Technology*, nel 2011, in collaborazione con la ditta esterna *Betacom*.

Il problema fondamentale evidenziato in questi ultimi anni, è che il numero delle risorse a disposizione è scarsamente sufficiente per accogliere tutti gli studenti iscritti al *Politecnico di Torino*. Infatti il numero degli aspiranti studenti ai corsi del primo anno di Ingegneria e di Architettura, in questi anni si è ridotto da 5000 a 4500, contro una domanda raggiunta, nel 2017, di 10000 partecipanti ai Test online di immatricolazione.

Una prima implementazione di orario automatizzato fu fatta da un tesista [2] per la ex *IV Facoltà*. L'estensione ad un orario di ateneo richiede tuttavia uno studio per gradi poiché il *Collegio di Ingegneria Gestionale* presenta un caso semplificato rispetto ad altri Collegi. Si è quindi pensato di cominciare a livello di ateneo con l'automatizzazione dell'assegnazione dei laboratori di informatica dell'*Area Information Technology* condivisi fra tutti i Collegi. Inoltre l'assegnazione dei laboratori è al momento l'unica operazione che, sebbene sia svolta in modo manuale, è già centralizzata poiché non dipende da altre attività.

Questa tesi ha quindi i seguenti scopi: automatizzare l'assegnazione di alcune risorse ad alcune attività didattiche, nello specifico l'assegnazione dei laboratori dell'*Area IT* alle attività di laboratorio programmate per tutta la didattica del *Politecnico di Torino* e ad analizzare pregi e difetti dell'applicativo web per la pubblicazione dell'orario.

1.2 Struttura della tesi

Questa tesi è composta dai seguenti capitoli.

Nel secondo capitolo verrà analizzato lo stato attuale della pianificazione dell'Orario accademico presso il *Politecnico di Torino*, evidenziando sia per la generazione dell'orario vero e proprio, sia per l'assegnazione delle risorse e sia per la pubblicazione dell'orario:

- cosa c'è
- lati positivi
- mancanze

Nel terzo capitolo verrà presentato un elenco di software, commerciali e open source, per automatizzare e semplificare tutta la generazione dell'orario per le scuole medie e superiori e per l'università. Mostriamo alcune caratteristiche per dare un'idea dello stato attuale dell'offerta che c'è in Italia e nel mondo: da semplici file in excel a software con interfacce grafiche che permettono facilmente di inserire tutti i dati relativi ai corsi, agli studenti, ai docenti e alle risorse (aule, laboratori e sedi) e di inserire i vincoli per elaborare automaticamente l'orario grazie a potenti algoritmi risolutivi.

Daremo anche uno sguardo agli orari pubblicati di altre Università: come il *Politecnico di Milano*, l'*Università di Torino* e l'*Università di Bologna*.

Nel quarto capitolo, vista la complessità del problema dell'orario (*timetabling problem* in inglese), verranno riportati alcuni articoli presentati alla Conferenza Internazionale PATAT (*Practice and Theory of Automated Timetabling*) che, ogni due anni, raccoglie un gran numero di studi ed applicazioni sulla generazione automatica di orari e turni in qualsiasi settore di attività (per esempio turni in ospedale, turni nei trasporti o organizzazioni sportive). La necessità di automatizzare questi tipi di problemi nasce dal fatto che questi richiedono un dispendio di energie e di tempo da parte delle persone coinvolte in questo lavoro e spesso le soluzioni raggiunte non sono soddisfacenti; automatizzando il processo invece si possono raggiungere soluzioni di alta qualità e in tempi ridotti. Nel secondo e terzo paragrafo del capitolo verrà spiegato come si è trasformato il nostro problema reale, ovvero automatizzare l'assegnazione dei corsi ai laboratori, in un modello matematico nell'ambito della Ricerca Operativa. In particolare è un modello di *programmazione lineare intera booleana* (PLI 0/1) dove viene definita una variabile decisionale x che può assumere valore 1 se l'evento si è verificato, 0 altrimenti. Nel nostro caso occorre verificare se *il corso è assegnato al laboratorio oppure no in un determinato modulo orario* (stabilito dai Referenti orari); dove per modulo orario si considera un'ora e trenta minuti. Inoltre vengono definiti i *vincoli* che, tramite disequazioni o equazioni matematiche, stabiliscono i valori assunti dalle variabili. Nel nostro caso si deve tenere presente che: tutti i laboratori sono chiusi nel primo modulo orario del lunedì mattina per manutenzione e al sabato per l'intera giornata; in ogni laboratorio il numero degli studenti all'interno non deve superare la capienza massima e possono stare solo due studenti per postazione; in ogni laboratorio è presente un solo corso, tranne in uno dove possono essere allocati due corsi nello stesso momento.

Nel quinto capitolo verranno mostrate le varie istanze necessarie per verificare se esiste o meno una soluzione al nostro problema, utilizzando il software risolutore Xpress-IVE e scrivendo il nostro modello matematico nel linguaggio di programmazione *Mosel*. Considereremo solo i corsi della Laurea Triennale e della Laurea Magistrale del primo periodo didattico (dell'anno accademico 2017 – 2018) perché più critico e solo i laboratori della sede Centrale/Cittadella di corso Duca e di via Boggio perché sono quelli dove si verificano maggiormente le sovrapposizioni delle richieste. Nella prima istanza, con i soli corsi della Laurea Triennale, e nella seconda istanza, con tutti i corsi della Laurea Triennale e Magistrale, verrà verificato quali e quanti corsi riusciremo ad assegnare ai laboratori, quali corsi non riusciremo ad assegnare, e quali laboratori rimarranno liberi e in quali fasce orarie. Nella terza istanza, a seguito dei risultati ottenuti nella seconda istanza, proveremo a completare l'assegnazione di tutti i corsi ai laboratori, considerando l'orario di riserva dei corsi rimasti "fuori".

Nel sesto capitolo verranno proposti i possibili miglioramenti sull'applicativo web in dotazione presso il *Politecnico di Torino* dal 2011, il software *Gestione Orari* [3] per una maggiore facilità nell'inserimento dell'orario e dei dati relativi ai corsi di studio, agli studenti, alle risorse, alle preferenze dei docenti e ai vincoli; frutto dell'esperienza personale di tutte le parti coinvolte, dal Referente orario al Referente per l'assegnazione delle risorse e frutto della consultazione dei manuali e delle interfacce web dei software, sia commerciali

che open source, disponibili in internet.

1.3 Obiettivi e utilità della tesi

L'obiettivo principale di questa tesi è realizzare un programma per automatizzare l'assegnazione dei laboratori dell'*Area IT* ai corsi di Ingegneria e Architettura che ne fanno richiesta. Operazione che fino ad oggi viene svolta in modo manuale e per molto tempo (quasi due mesi), ma che se svolta in modo automatico (ottenendo il risultato nel minor tempo possibile), consentirà alle persone coinvolte di riacquistare del tempo da dedicare ad altre attività lavorative.

In particolare vorremmo dimostrare che non è possibile assegnare i laboratori in base alle richieste programmate e definite dalla generazione degli orari decentrata da parte dei Referenti Orari, ma che si può solo cercare di assegnare il massimo numero di richieste ad ogni laboratorio; e solo utilizzando orari di "riserva" potremmo completare l'assegnazione.

L'obiettivo futuro è estendere questa procedura automatizzata a tutte le risorse di Ateneo per "centralizzare" l'assegnazione di tutte le attività didattiche di Ateneo, riducendo notevolmente parte della pianificazione dell'orario e rendendo meno oneroso il lavoro svolto dal personale docente e dal personale amministrativo.

Per quanto riguarda l'applicativo web l'obiettivo è quello di mostrare le possibili migliorie per una maggiore fruibilità nel suo utilizzo e nella pianificazione "manuale" dell'assegnazione delle risorse come supporto a quella "automatica".

Oltre alle motivazioni già indicate, si è scelto di svolgere questa tesi anche perché:

- si vuole rendere più visibile all'intero corpo docente la complessità del problema e il lavoro "dietro le quinte" che viene svolto dal personale docente e dal personale tecnico amministrativo;
- si vuole sopperire al fatto che, ad oggi, al *Politecnico di Torino* non esista un processo per centralizzare la pianificazione dell'orario, fornendo almeno un punto di partenza che invero corrisponde alla parte finale del problema: assegnamento delle attività didattiche alle aule e ai laboratori. Si è scelto questo, come punto di partenza, perché il problema maggiore è che le risorse non sono sufficienti e adeguate per tutti i corsi che vengono erogati, e perché l'allocazione dei laboratori è l'unica operazione svolta centralmente;
- molti docenti non conoscono ancora bene l'applicativo web dell'orario delle lezioni, malgrado esista già da otto anni, e non conoscono l'utilizzo della parte di "prenotazione aula" che hanno a disposizione sulla loro home page.

Capitolo 2

Stato attuale della pianificazione dell'orario accademico

Ad oggi la pianificazione dell'Orario accademico presso il *Politecnico di Torino* è un'operazione che viene svolta da una *Commissione Orari*, composta dal personale docente e dal personale tecnico amministrativo, e suddivisa per i *Collegi dei Corsi di Studio* di Ingegneria e di Architettura. La pianificazione è suddivisa in tre parti: la prima parte è legata alla generazione dell'Orario accademico per ogni semestre dell'anno accademico, la seconda parte è legata all'assegnazione delle risorse dove svolgere le attività accademiche e la terza parte corrisponde alla pubblicazione dell'Orario su un applicativo web.

2.1 Generazione dell'orario accademico

Secondo il Regolamento Didattico del *Politecnico di Torino* [24] l'organizzazione e la gestione dei Corsi di Laurea e Laurea Magistrale sono realizzate dai *Collegi dei Corsi di Studio*, e attualmente ce ne sono 14. Mentre i corsi del primo anno sono comuni a tutti i corsi di studio e sono quindi gestiti e pianificati da una Commissione preposta.

Per ogni *Collegio* viene assegnato uno o più *Referente Orario*, che ha il compito di generare il proprio Orario accademico, seguendo alcune linee guida generiche stabilite dall'Ateneo e secondo vincoli organizzativi legati ai singoli percorsi di studio del proprio Collegio.

Ogni anno, indicativamente nel mese di maggio, viene programmata una riunione ufficiale alla quale partecipano il *Vice Rettore per la Didattica*, tutti i *Referenti Orari/Esami*, la *Commissione preposta* per i primi anni in comune, e il personale tecnico Amministrativo dell'*Area Information Technology* (IT) e dell'*Area Edilizia e Logistica* (EDILOG). Durante questo incontro vengono indicati:

- Calendario accademico
- Piano di lavoro
- Comunicazioni per gli orari

- Sostenibilità
- Esami

Calendario accademico

Il *calendario accademico* è composto da:

- 2 semestri: ogni semestre è composto da 14 settimane di lezione
- 6 settimane per la sessione esami invernale/estiva e 3 settimane per la sessione esami autunnale (settembre)

Vengono quindi indicate le date di inizio e di fine di ogni semestre, le pause per le festività, e le date di inizio e di fine delle sessioni di esame e delle lauree di Ingegneria e di Architettura.

Piano di lavoro

Il *piano di lavoro* descrive il flusso temporale di lavoro che i Referenti orari dovrebbero seguire per la pianificazione dell'orario dell'intero anno accademico:

- fine MAGGIO: raccolta dati per gli orari del I e II semestre tramite la compilazione della scheda online, **Modulo di predisposizione orari**, da parte del docente titolare del corso;
- GIUGNO/LUGLIO: generazione degli schemi orari del I e II semestre;
- LUGLIO: pubblicazione orari del I semestre da parte di alcuni Referenti e dal personale tecnico amministrativo dell'Area IT e EDILOG
- SETTEMBRE: pubblicazione orari del II semestre.

Comunicazioni per gli orari

Nelle *comunicazioni* vengono indicate:

- la non disponibilità di alcune aule a causa di lavori di ristrutturazione;
- eventuali risorse aggiuntive;
- azioni per risolvere criticità legate alla numerosità dei corsi;
- l'assegnazione di alcune aule per i corsi del primo anno comune Laurea Triennale di Ingegneria (I semestre), per i corsi di Tutorato e per i corsi dei Talenti;
- giorni e fasce orarie per i corsi a scelta della Laurea Triennale e Laurea Magistrale, per alcuni corsi dei Talenti e per i Tutorati;
- eventuali vincoli sull'orario (per esempio per il corso di Fisica II/talenti del secondo anno di Ingegneria per il I semestre);

- Sovra allocazione aule e laboratori:
 - risorse associate ad ogni corso: non allocare più ore di quelle assegnate (per esempio per un corso da 6 CFU, senza sdoppiamento squadre deve avere 3 slot a settimana), non è sostenibile la doppia allocazione "aula-laboratorio" per i corsi che utilizzano prevalentemente il laboratorio (a partire da 9 o 10 settimane su 14)
 - le risorse che non vengono utilizzate devono essere liberate a beneficio di eventuali recuperi o altre necessità.

Tutte informazioni che il Referente Orario deve tenere presente durante la generazione dell'Orario per garantire la compatibilità di orario, evitando le sovrapposizioni con tutti i corsi.

Sostenibilità

In questa sezione vengono presentate una serie di percentuali che indicano il livello di saturazione delle aule, a seconda delle taglie, su stime basate sulla numerosità degli studenti dell'anno accademico precedente e poi le stesse con un aumento di una percentuale sulla numerosità dell'anno accademico successivo (che viene stabilita ogni anno); tenendo conto del modello di 51 ore a settimana per aula e considerando che il *modulo orario* è stato fissato per *un'ora e trenta minuti*. Quindi la settimana è composta da 34 moduli orari: dal lunedì al giovedì dalle 8.30 alle 19, e il venerdì dalle 8.30 alle 17.30; l'ultimo modulo orario del venerdì dalle 17.30 alle 19.00 e il sabato mattina devono essere utilizzati solo nel caso di eventuali recuperi di lezioni qualora non ci fossero altre possibilità durante la settimana.

Il taglio delle aule è suddiviso in base al numero degli studenti che possono stare all'interno: XS (0 – 50), S (51 – 100), MS (101 – 150), ML (151 – 200), L(201 – 250), XL (> 250).

Esami

Nella sezione *Esami* vengono indicate delle regole e alcuni vincoli per la programmazione degli esami per tutte le sessioni, viene presentato il calendario degli esami del primo anno, degli esami comuni del secondo anno, degli esami a scelta della laurea Triennale e proposte di Ateneo per il secondo anno. Viene anche indicato il flusso di lavoro per la pianificazione e pubblicazione da parte dei Referenti Esami. Questa sezione però viene solo accennata in questo contesto e non sviluppata successivamente perché la pianificazione viene fatta diversamente da quella dell'orario, ma è necessario sottolineare che per alcuni Collegi il Referente Orario è lo stesso anche per gli esami.

2.1.1 Problematiche nella generazione dell'orario accademico

In questo paragrafo cerchiamo di evidenziare una serie di problematiche che si incontrano già nel flusso di lavoro della generazione dell'Orario.

Il primo problema è dato dal fatto che non esiste un manuale di istruzione per la compilazione della scheda online, il **Modulo di predisposizione orari**, e non esistono dei tasti *help in linea* per guidare l'utente, passo dopo passo, nei vari campi della scheda che tutti i docenti titolari dei corsi del I e II semestre sono invitati, ma non "obbligati", a compilare a seguito di una comunicazione via mail da parte del Vice Rettore della Didattica. Il giorno dopo e per circa una decina di giorni rimane attiva la possibilità di compilare questa scheda.

In questa scheda online, alla pagina iniziale Figura 2.1, è presente l'elenco dei corsi/incarichi di cui il docente è titolare, con la sola indicazione dell'anno accademico e del periodo didattico di riferimento.

Selezione Corso			
Anno Accademico:	2016		
Docente Titolare:	BORCHIELLINI ROMANO		
E-mail:	romano.borchellini@polito.it		
Elenco Corsi:	Nome	Periodo	Stato
	➔ Numerical design of thermal systems	1	Da Compilare
	➔ Termodinamica applicata e trasmissione del calore	2	Da Compilare

Figura 2.1. Scheda online pagina iniziale

Il campo "Stato" indica la situazione del modulo se è *da compilare* oppure se è stato *inviato*. Il docente dopo aver selezionato il corso si ritrova nella scheda Passo 1, Figura 2.2, dove sono presenti due sezioni: a sinistra i **dati incarico** a destra le **informazioni generali sul corso** e le **informazioni aggiuntive**.

DATI INCARICO

Tipologia Lezione	Num. squadre	Num. ore
Lezione	1	60
Esercitazione laboratorio	2	40
Totale		100

Tot. blocchi settimanali: 4,77 - Max blocchi: 5

Per richiesta di informazioni o segnalazione di eventuali problemi sulla struttura dell'insegnamento o nella compilazione del modulo contattare servizio.intranet.swa@polito.it

Passo 1/3

INFORMAZIONI GENERALI SUL CORSO

Tipologia Aula:

Banchi Disponibilità alimentazione (Fresche elettriche per PC)

Tavoli da disegno

Modalità Lezione Prevalente:

Lavagna

Proiezione da PC

Altro

Preferenza sull'organizzazione dei blocchi in aula:
(Blocchi da 1,5h/ 3h)

Blocchi da 1,5h

Un blocco da 3h e gli altri da 1,5h

Tutti i blocchi da 3h

INFORMAZIONI AGGIUNTIVE

Eventuali e motivate incompatibilità di orario dei docenti coinvolti:
(ad esempio attività didattica in altri corsi nello stesso semestre)

Info aggiuntive

Ulteriori comunicazioni sull'organizzazione del corso che possono essere utili in fase di predisposizione degli orari

Per le lezioni/esercitazioni in aula, possibilmente due giorni adiacenti ad inizio o fine settimana (LU+MA, oppure GI+VE o LU+VE).

Continua >>

Figura 2.2. Scheda online PASSO 1

Nella sezione **dati incarico** sono presenti le informazioni relative al numero di squadre e al monte ore delle lezioni e delle esercitazioni di laboratorio per l'intero periodo didattico. Poi viene indicato il numero totale di blocchi settimanali che si possono allocare in una settimana, dove per blocco si intende il modulo orario di un'ora e trenta minuti. Queste informazioni arrivano direttamente dalla Gestione della Didattica, quindi non possono essere modificate.

Ma mancano i seguenti dati:

- il numero degli studenti iscritti nell'anno accademico precedente,
- il *credito formativo universitario* (CFU) dove ad ogni credito corrispondono 25 ore,
- la suddivisione dei blocchi settimanali tra lezione ed esercitazione.

Nella sezione **informazioni generali sul corso** si richiede al docente di "selezionare" alcuni campi come:

- *tipologia di aula*: banchi, tavoli da disegno o disponibilità di alimentazione (prese elettriche per PC),
- *modalità lezione prevalente*: lavagna, proiezione da PC, altro
- *preferenza sull'organizzazione dei blocchi in aula*: blocchi da 1.5h o da 3h o entrambi.

Nella sezione **informazioni aggiuntive** si richiede di compilare dei campi non pre-strutturati quali:

- *eventuali e motivate incompatibilità di orario dei docenti coinvolti*;
- *ulteriori comunicazioni sull'organizzazione del corso che possono essere utili in fase di predisposizione degli orari*.

Avere questi due ultimi campi non pre-strutturati porta i seguenti svantaggi:

- le informazioni che vengono descritte sono troppo lunghe e ricopiate in entrambi i campi oppure sono completamente vuote;
- vengono aggiunte ulteriori informazioni sulla suddivisione dei moduli orari (diversa da quella fornita dalla Didattica);
- manca la richiesta del numero di crediti che il docente deve coprire e quelli assegnati invece agli assegnisti;
- manca la distribuzione dei crediti nel corso del semestre (perché a volte non sono distribuiti uniformemente lungo il semestre).

Nella scheda Passo 2, Figura 2.3 pag. 10, è presente la sezione **Organizzazione dei laboratori** dove si richiede al docente di indicare la **Tipologia di laboratorio** "per le attività di lezione/esercitazione o per le attività di laboratorio, utilizzando blocchi settimanali per l'intero semestre o per un numero di blocchi limitato".

Il docente può quindi "scegliere":

Passo 2/3

DATI INCARICO

Tipologia Lezione	Num. squadre	Num. ore
Lezione	1	60
Esercitazione laboratorio	2	40
Totale		100

Tot. blocchi settimanali: 4,77 - Max blocchi: 5

Per richiesta di informazioni o segnalazione di eventuali problemi sulla struttura dell'insegnamento del modulo cont...

ORGANIZZAZIONE DEI LABORATORI

In questa sezione è possibile indicare i laboratori utilizzati sia per attività di lezione/esercitazione che per attività di laboratorio con utilizzo di blocchi settimanali per l'intero periodo o per un numero di blocchi più limitato

Tipologia di Laboratorio:

Aggiungi Laboratorio

	Laboratorio	Sede	Posti a sedere	Tipo di utilizzo	Modulo Attrezzature
✘	LEP	SEDE CENTRALE	78 / 78	Prevalente settimanale (>10)	
✘	LAB1D	SEDE CENTRALE	156 / 156	Numero limitato di settimane	

<< Indietro Continua >>

Selezione Laboratorio

Nome Laboratorio: LEP

Num. posti a sedere utilizzati: 78 / 78 Num. postazioni utilizzate: 0

Tipo di utilizzo: Prevalente settimanale (>10) Numero limitato di settimane

Software/Attrezzature: Wireshark, XAMPP

Contatto di riferimento per utilizzo lab: contatto laib

Note: (Es. squadre ecc.)
Un blocco da 3h ogni settimana (usato 1.5h per ciascuna delle due squadre).

*L'assegnazione dello specifico laboratorio indicato in preferenza è comunque soggetto a disponibilità (è possibile la sovrastituzione con altro laboratorio della stessa tipologia)

Salva Chiudi

Figura 2.3. Scheda online PASSO 2

- *il nome del laboratorio*, in base ad un elenco dei laboratori presenti nell'Ateneo;
- *il numero di posti a sedere da utilizzare*; che corrisponde al numero massimo degli studenti che può contenere, ovvero la *capienza massima*;
- *il numero di postazioni utilizzate*, che corrisponde al numero di computer di cui necessita;
- *il tipo di tipo di utilizzo*: se "prevalente settimanale (> 10)" o se per un "numero limitato di settimane".

E poi sono presenti altri tre campi pre-strutturati per indicare: il *software/attrezzature*, il *contatto di riferimento per utilizzo lab*, e un ulteriore campo *note*.

La schermata del PASSO 3, che non mostriamo, richiede i dati relativi per la pianificazione degli esami.

Tutti questi campi però "*non sono obbligatori*", quindi anche se il docente apre la scheda è libero di compilarla o meno ed inviarla con informazioni incomplete. Scaduto il termine di compilazione, tutte queste schede vengono estrapolate in un **file**, in formato excel, composto da 33 colonne relative al corso e 6 colonne relative agli esami.

Alcune di queste colonne contengono i dati forniti dall'Area Gestione Didattica, non modificabili, quali: *matricola titolare*, *cognome*, *nome*, *collaboratori*, *codice insegnamento*, *numcor*, *tipologia - tipo laurea - anno - alfabetica - nome sede*, *titolo materia*, *periodo*

didattico, docente titolare, email; e ancora: *numero blocchi lezione, numero squadre esercitazione, numero blocchi esercitazione per squadra, numero squadre laboratorio, numero blocchi laboratorio per squadra*. Poi sono riportate delle colonne che indicano i dati logistici (sede e indirizzo) del laboratorio selezionato; infine le colonne corrispondenti ai campi compilati e selezionati dal docente.

In una di queste colonne viene indicato lo "stato del modulo":

- **modulo inviato**: il docente ha inviato il modulo ma non è detto che abbia compilato tutti i campi presenti nel modulo;
- **modulo in bozza**: il docente ha iniziato la compilazione e non l'ha portata a termine e non l'ha inviata oppure il docente ha compilato negli anni precedenti e se non la modifica nell'anno in corso allora rimangono i dati relativi all'ultima compilazione.

Se ovviamente non ha mai compilato la scheda non comparirà nessun dato relativo al docente e ai suoi incarichi.

Infatti dai dati raccolti nell'anno accademico 2017 – 2018:

- per il primo semestre: su 694 record, 21 risultano celle incomplete senza dati del docente o del corso, 349 risultano moduli **in bozza** e 324 risultano moduli **inviati**;
- per il secondo semestre: su 630 record, 22 risultano celle incomplete senza dati del docente o del corso, 370 risultano moduli **in bozza** e 238 risultano moduli **inviati**.

Il Referente Orario estrapola i dati e, sulla base dell'orario dell'anno precedente, incomincia a costruire manualmente quello dell'anno seguente. Ma i problemi che riscontra sono:

1. non fruibilità dei dati raccolti: occorre modificare un file excel composto da 40 colonne per rendere leggibili e chiari i dati inseriti;
2. manca il numero degli iscritti dell'anno accademico precedente;
3. manca il CFU;
4. manca il monte ore totale delle lezioni e delle esercitazioni in aula e in laboratorio per il periodo didattico;
5. manca il numero di crediti che deve coprire il docente e quello degli assegnisti;
6. mancano importanti informazioni da parte del docente che possono essere utili per la generazione dell'orario;
7. c'è un'incongruenza nei dati forniti dal docente e dalla Didattica (per esempio la suddivisione dei moduli orari per le lezioni in aula, le esercitazioni in aula e in laboratorio).

Quindi più della metà dei dati raccolti o non sono completi o non sono corretti e questo costringe il Referente a contattare tutti i docenti e a ricercare i dati mancanti nelle altre aree della Gestione Didattica (consultando tutti i corsi erogati dal proprio Collegio), per ottenere quelle informazioni necessarie per generare l'orario; di conseguenza oltre a rifare il lavoro due volte, i tempi richiesti nel flusso di lavoro, indicati nel paragrafo precedente, non vengono mai rispettati. Infatti, nei mesi di giugno e luglio è possibile solo generare l'Orario del I semestre, quello del II semestre è rimandato nei mesi di novembre e dicembre.

Un altro problema importante è la **decentralizzazione della pianificazione** e i **metodi diversi nella fase di generazione dell'orario** che i vari Referenti Orari utilizzano. Questo è dovuto al fatto che il parco aule dell'Ateneo è suddiviso per i corsi del Primo anno in comune e per i corsi di alcuni Collegi secondo criteri che esistono da diversi anni, non tenendo conto dell'aumento sulla numerosità degli studenti nei corsi degli anni successivi al primo e sull'offerta formativa. Il problema maggiore però si verifica presso la sede Centrale/Cittadella perché i Referenti Orari dei Collegi dotati di un parco aule proprie, costruiscono l'orario non solo in funzione dei vincoli legati ai docenti e al percorso di studi degli studenti ma anche in funzione delle aule a loro assegnate ottimizzando sia il tempo che le risorse; mentre i Referenti Orari degli altri Collegi devono costruire l'orario delle lezioni solo in funzione dei vincoli legati ai docenti e al percorso di studi, nella speranza di avere assegnata un'aula sufficientemente adeguata.

2.2 Assegnazione delle risorse di Ateneo

Presso la sede Centrale, il vice Rettore della didattica e la Commissione Orari dei corsi del primo anno in comune assegna ogni anno un parco aule destinate esclusivamente a tutti i corsi del Primo anno di Ingegneria, un parco aule destinate al *Collegio di Ingegneria Gestionale* (la ex *IV Facoltà*), un parco aule al *Collegio di Ingegneria Informatica, Cinema e Meccatronica (ICM)* e al *Collegio di Elettronica, Telecomunicazioni e Fisica (ETF)* (la ex *III Facoltà*). Di conseguenza i Referenti Orari di questi Collegi generano l'orario in funzione delle aule a loro assegnate.

Il restante parco aule è destinato ai corsi di tutti gli altri Collegi (la ex *I Facoltà*) e vengono assegnate in modo centralizzato dal personale tecnico amministrativo dell'Area Logistica.

I Referenti Orari dei Collegi che non hanno un parco aule proprie, dopo aver generato "manualmente" una bozza orario, si recano dal Responsabile delle aule della Sede Centrale/Cittadella, secondo un calendario definito tra le parti legato alla numerosità degli studenti dei corsi afferenti ai vari Collegi. Durante questi incontri vengono provvisoriamente assegnate delle aule sufficientemente adeguate per capienza. Questo processo viene eseguito manualmente, tramite l'ausilio di una lavagna (tradizionale) suddivisa per righe e per colonne che compongono la "settimana tipo"; dove le righe corrispondono alle aule della sede Centrale/Cittadella, e le colonne corrispondono ai giorni della settimana (da lunedì a venerdì) e ogni giorno è suddiviso dai vari moduli orari (ogni modulo corrisponde ad un'ora e mezza).

In ogni "cella" vengono allocate delle schede relative ai vari corsi, di cartoncino colorato in base ai Collegi afferenti. Il processo di assegnazione di queste aule viene sviluppato all'incirca in tre settimane.

Ma parte delle attività didattiche, quali esercitazioni di laboratorio, vengono svolte o nei laboratori specifici dei singoli Dipartimento, gestiti dai singoli Referenti, o nei laboratori di Informatica di Ateneo, gestiti in modo centralizzato dal personale tecnico dell'Area IT. Una volta che tutti i Referenti Orari di tutti i Collegi hanno una bozza orario con le relative aule assegnate, si recano dal Referente dei laboratori di Informatica dell'Area IT, anche qui secondo un calendario di incontri in base alla numerosità delle richieste e al numero degli studenti dei vari corsi per ogni Collegio.

Durante questi incontri vengono raccolte le richieste in modo manuale e in un foglio, formato A3, dove viene rappresentata la "settimana tipo": le righe indicano i moduli orari da un'ora e trenta minuti, le colonne indicano i giorni della settimana, da lunedì a venerdì, perché al sabato i laboratori sono chiusi.

Quindi in ogni "cella" vengono segnati tutti i corsi che necessitano l'utilizzo del laboratorio; una volta riempite tutte le "celle" si individuano tutte le sovrapposizioni e viene fatta una prima valutazione di quali corsi dovrebbero necessariamente cambiare orario perché i laboratori a disposizione non sono sufficienti per ricoprire tutte le richieste.

Questa prima valutazione viene fatta dal Referente dei laboratori sulla base delle linee guida indicate durante la Riunione con la Commissione Orari e il vice Rettore della Didattica, come:

- dare la precedenza ai corsi del primo anno in comune;
- evitare la sovra allocazione aula e laboratorio nel caso quest'ultimo non venga utilizzato per un numero di settimane almeno maggiore a sette;

e sulla base di alcune caratteristiche dei laboratori dell'Area IT che verranno indicate nel quarto capitolo.

A seguito di questa prima valutazione il Referente dei laboratori cerca "manualmente" di assegnare il massimo numero di corsi ai laboratori; successivamente segnala ai vari Referenti orari dei Collegi le sovrapposizioni riscontrate e i corsi che a suo giudizio dovrebbero cambiare orario, proponendo, dove possibile, eventuali soluzioni di orario "alternativo" per poter soddisfare tutte le richieste. Di conseguenza le modifiche di orario legate all'assegnazione dei laboratori possono comportare ulteriori modifiche di orario per le lezioni che si svolgono in aula.

2.2.1 Problematiche nell'assegnazione delle risorse di Ateneo

In questo paragrafo evidenziamo alcune delle problematiche riscontrate durante l'assegnazione delle risorse.

Mancano le risorse

Il più grande problema dell'assegnazione delle risorse è legato allo scarso numero di aule di capienza adeguata (in particolare quelle di taglia medio grande) ma non è argomento

di questa tesi perché le decisioni vengono prese dal Rettore e dalla squadra di governo dell'Ateneo. Qui lo citiamo soltanto perché più volte sottolineato dallo stesso Rettore: le risorse del *Politecnico di Torino* non sono sufficienti a contenere la numerosità degli studenti iscritti. Infatti, siamo partiti con 5000 studenti iscritti al primo anno nel (2011) per poi scendere ogni anno fino ai 4500 nell'anno accademico 2017 – 2018, contro una domanda di 10000 partecipanti ai Test Online di immatricolazione raggiunta nel 2017.

Suddivisione parco aule dell'Ateneo

Un altro problema, evidenziato più volte dalla Commissione dei Referenti dei vari Collegi, è data dalla suddivisione del parco aule che viene fatta secondo criteri che esistono da diversi anni e che non tengono conto dei cambiamenti sulle numerosità degli studenti e sull'offerta formativa, che non riguarda solo i corsi del primo anno in comune ma anche quella degli anni successivi.

Al momento solo la pianificazione dei corsi del primo anno in comune è centralizzata con un parco aule assegnate, e un numero di laboratori dell'Area IT garantiti per le esercitazioni di laboratorio dei Corsi di Informatica (solo il tipo di laboratorio può variare di anno in anno). Mentre i corsi con un numero di studenti elevato degli anni successivi si ritrovano a non avere delle aule sufficientemente capienti per le loro esigenze didattiche.

Ma oltre alle lezioni ed esercitazioni in aula esistono le attività didattiche che devono essere necessariamente svolte nei laboratori specifici, situati nei singoli Dipartimenti, e nei laboratori di Informatica dell'Area IT. Mentre l'assegnazione dei laboratori specifici viene gestita dai Referenti dei vari Collegi, l'assegnazione dei laboratori di Informatica dell'Area IT (considerati da diversi anni come risorse di Ateneo) viene pianificata in modo unico e centralizzato dal Referente dei laboratori.

Quindi tutti i corsi successivi al primo anno in comune, di Ingegneria e di Architettura, costruiscono l'orario completo delle lezioni con l'aula assegnata ma l'orario delle esercitazioni di laboratorio viene pianificato nella speranza di avere successivamente assegnato un laboratorio; e anche se si è creato il migliore orario possibile questo potrà essere ulteriormente modificato durante l'assegnazione dei laboratori.

Perché se la generazione dell'orario è decentralizzata, la probabilità di sovrapposizioni dei corsi nello stesso modulo orario è alta, e il Referente dei laboratori, una volta raccolti tutti i dati, cerca "manualmente" di assegnare il massimo numero di corsi ai laboratori. Le sovrapposizioni riscontrate verranno indicate e discusse con i vari Referenti dei Collegi fino a completare l'assegnazione; e le modifiche di orario legate all'assegnazione dei laboratori possono comportare ulteriori modifiche di orario per le lezioni che vengono svolte in aula. Quindi l'intero processo della generazione dell'orario con l'assegnazione delle risorse richiede quasi un mese e mezzo di lavorazione.

Numerosità variabile dei corsi

É un problema che si riscontra in prossimità dell'inizio del semestre e a semestre avviato (il termine di iscrizione varia o novembre o marzo) ed è legato alla numerosità degli studenti dei corsi che cambia rispetto alle previsioni e questo comporta un'ulteriore lavorazione e

modifica dell'Orario e dell'assegnazione delle risorse.

Quindi lavorare alla generazione dell'orario di ogni semestre comporta un impegno quasi a tempo pieno per circa un mese e mezzo, e saltuariamente per tutto l'anno a causa delle modifiche; e un notevole dispendio di energie da parte dei Referenti Orari (la maggior parte sono docenti) che non vengono utilizzate per la propria attività didattica.

Lo scopo di questa tesi è mostrare che almeno una parte di questo lavoro potrebbe essere svolto automatizzando l'assegnazione delle risorse, in particolare ci occupiamo dell'allocatione dei laboratori dell'Area IT.

2.3 Applicativo web per la pubblicazione dell'orario

Dal 2011 la pubblicazione dell'Orario accademico del *Politecnico di Torino* viene svolta utilizzando un applicativo web, il software **Gestione Orari** [3] creato dall'Ufficio Sviluppo e gestione applicazioni dell'*Area Information Technology*, nel 2011, in collaborazione con una ditta esterna *Betacom*.

Questo software permette di comporre e pubblicare l'orario delle lezioni di tutti i corsi erogati presso il *Politecnico di Torino*, ed è anche uno strumento di consultazione per tutti gli utenti (docenti, studenti e personale tecnico amministrativo).

Il software **Gestione Orari** si basa su una quantità di dati forniti dall'*area Gestione Didattica* che è l'unico servizio che può modificarli tramite uno strumento esterno al software; con cadenza giornaliera un altro servizio preposto si occupa di allineare i dati registrati all'interno del software con i dati didattici.

L'utente abilitato all'utilizzo del software non può modificare questi dati ma può utilizzarli per costruire l'orario; solo per alcuni dati si possono effettuare delle modifiche.

Alcuni dati importanti forniti dall'*Area Gestione Didattica* sono: Anni accademici, Periodi didattici, Area, Corso di Laurea, Docenti, Docenti titolari, Materie, Alfabetiche, Orientamenti, Squadre, Aule, Laib.

L'utente abilitato all'accesso del software, utilizzando le proprie credenziali (username e password) ha a disposizione un elenco delle diverse aree dove poter lavorare: *Gestione template orari*, *Ricerca orario di un docente*, *Gestione Calendario*, *Ricerca eventi puntuali*, *Visualizza i cdl non pubblicati*, *Gestione richieste e conflitti*, *Visualizzazioni Log Operazioni*, *Esportazione preferenze docenti*, *Stampa per bacheca*, *Pubblicazione Eventi Pulizie*.

Indichiamo di seguito le caratteristiche delle aree maggiormente utilizzate.

Nell'area **Gestione template orari** l'utente ha una serie di *Filtri di ricerca* (Anno accademico, Periodo didattico, Sede, Tipo di Laurea, Area, Cdl) per visualizzare il template orario relativo al corso di Laurea per un determinato anno accademico e periodo didattico che deve costruire o modificare.

Successivamente l'utente può costruire l'orario del corso di Laurea selezionato tramite quattro sezioni: *Filtri di Ricerca*, *Materia*, *Elenco blocchi per la materia selezionata*, e un quadro che rappresenta la settimana tipo, Figura 2.4 a pag. 16, dove vengono allocati i vari blocchetti legati alla Materia.

INGEGNERIA AEROSPAZIALE-2011 (L-9) - 1° anno - 1° periodo - TORINO

Filtri di Ricerca

Anno: 1
 Alfabetica: CAPOM -CENS -4*
 Orientamento: Tutti
 Visualizza crediti liberi

Materia

Materia: Informatica*
 Applica filtro materia a visualizzazione
 Vedi altri cdl che hanno questa materia

Elenco Blocchi per la materia selezionata

Informatica Studenti Previsti: 304 Studenti Frequentanti: 0 MEZZALAMA MARCO() CAPOM -CENS -4* Lezione/Esercitazione Ore rimanenti : 4,50	Informatica Studenti Previsti: 101 Studenti Frequentanti: 0 MEZZALAMA MARCO() CAPOM -CENS -4* Lezione/Esercitazione SQUADRA 1 Ore rimanenti : 1,50	Informatica Studenti Previsti: 101 Studenti Frequentanti: 0 MEZZALAMA MARCO() CAPOM -CENS -4* Lezione/Esercitazione SQUADRA 2 Ore rimanenti : 1,50	Informatica Studenti Previsti: 101 Studenti Frequentanti: 0 MEZZALAMA MARCO() CAPOM -CENS -4* Lezione/Esercitazione SQUADRA 3 Ore rimanenti : 1,50
---	--	--	--

Figura 2.4. Gestione con i Filtri

Tramite i *Filtri di Ricerca* l'utente abilitato sceglie: l'Anno per definire l'anno del corso di Laurea; l'Alfabetica nel caso in cui il Corso di Laurea fosse suddiviso in più alfabetiche (per via della numerosità degli studenti iscritti a quel corso), ed eventualmente l'Orientamento nel caso per esempio dei Corsi di Laurea erogati solo in inglese, o per i Talenti.

Tramite il filtro *Materia* si seleziona la materia, ovvero l'Insegnamento, da allocare per costruire il template orario. Questo elenco e la loro appartenenza ai diversi Corsi di Laurea è fornito dall'Area *Gestione Didattica* e quindi non può essere modificato dall'utente. È possibile visualizzare, tramite il collegamento "Vedi altri cdl che hanno questa materia", l'elenco dei Corsi di Laurea che includono la materia selezionata, ed in particolare si può identificare, dal colore blu della scritta, il Corso di Laurea *Capo Aggregazione*.

Nella sezione *Elenco Blocchi per la materia selezionata*, dopo aver selezionato la materia, vengono visualizzati i blocchi relativi alle lezioni/esercitazioni e alle eventuali squadre, che devono essere allocati nel riquadro che rappresenta la settimana tipo, Figura 2.5.

Elenco Blocchi per la materia selezionata

Informatica Studenti Previsti: 98 Studenti Frequentanti: 101 MEZZALAMA MARCO() CAPOM -CENS -4* Lezione/Esercitazione SQUADRA 1 Ore rimanenti : 0,50	Informatica Studenti Previsti: 98 Studenti Frequentanti: 101 MEZZALAMA MARCO() CAPOM -CENS -4* Lezione/Esercitazione SQUADRA 2 Ore rimanenti : 0,50	Informatica Studenti Previsti: 98 Studenti Frequentanti: 101 MEZZALAMA MARCO() CAPOM -CENS -4* Lezione/Esercitazione SQUADRA 3 Ore rimanenti : 0,50	Informatica Studenti Previsti: 216 Studenti Frequentanti: 304 MEZZALAMA MARCO() CAPOM -CENS -4* Lezione/Esercitazione SQUADRA 4 Ore rimanenti : 0,50
---	---	---	--

Shift orario rispetto alla sede: 0 min.

Visualizzazione Gornaliera

	lunedì	martedì	mercoledì	giovedì	venerdì	sabato
8 ⁰⁰						
9 ⁰⁰	Analisi matematica I CODIGIONE MARCO CAPOM -CENS -4 5					
10 ⁰⁰						
11 ⁰⁰						
12 ⁰⁰	Informatica MEZZALAMA MARCO CAPOM -CENS -4 5			Analisi matematica II CODIGIONE MARCO CAPOM -CENS -4 5 SQUADRA A	Chimica FRANCIA CARLOTTA CAPOM -CENS -4 5 SQUADRA B	
13 ⁰⁰		Informatica MEZZALAMA MARCO CAPOM -CENS -4 5		Chimica FRANCIA CARLOTTA CAPOM -CENS -4 5 SQUADRA C	Analisi matematica II CODIGIONE MARCO CAPOM -CENS -4 5 SQUADRA D	
14 ⁰⁰	Informatica MEZZALAMA MARCO CAPOM -CENS -4 LABEL	Analisi matematica I CODIGIONE MARCO CAPOM -CENS -4 5	Chimica FRANCIA CARLOTTA CAPOM -CENS -4 5			
15 ⁰⁰						
16 ⁰⁰	Informatica MEZZALAMA MARCO CAPOM -CENS -4 LABEL	Chimica FRANCIA CARLOTTA CAPOM -CENS -4 5				
17 ⁰⁰	Informatica MEZZALAMA MARCO CAPOM -CENS -4 LABEL					
18 ⁰⁰	Informatica MEZZALAMA MARCO CAPOM -CENS -4 LABEL		Analisi matematica I CODIGIONE MARCO CAPOM -CENS -4 5			

Figura 2.5. Template orario

Ciascun blocco contiene le informazioni relative alla *materia*, al numero degli *studenti previsti* e *studenti frequentanti*, al *docente titolare* del Corso di Laurea, *alfabetica*, eventuale *squadra*, e il numero di *ore rimanenti* da allocare per la lezione o per le esercitazioni delle squadre. Questo numero inizialmente indicherà il numero massimo di blocchi da allocare per la materia selezionata, poi durante la costruzione dell'orario questo numero diminuirà fino a quando sono allocati tutti i blocchi.

Ma mancano l'informazione legata al CFU e il monte ore totale (che però lo ritroviamo in una sottosezione), e il numero degli studenti previsti e frequentanti non è ben chiaro perché non si sa se si riferisce all'anno accademico precedente o a quello in corso.

Solo alcuni utenti sono abilitati (solitamente il personale tecnico amministrativo preposto per la costruzione e pubblicazione dell'Orario) a modificare il numero di blocchi di lezione per tipologia, tramite un pulsante per la *Gestione blocchi - Lezione, Esercitazioni in aula e Esercitazioni in laboratorio* - da allocare, dove ogni blocco corrisponde ad un modulo orario, ovvero ad una durata di un'ora e trenta minuti. In ogni sottosezione viene indicato il numero totale di ore per l'intero semestre, ma manca il numero di blocchi a settimana, Figura 2.6.

Riepilogo ▲

Periodo Didattico: 1 *	Tipo di Laurea: Laurea	Facoltà: INGEGNERIA I
Corso di Studi: INGEGNERIA AEROSPAZIALE-2011 (L-9)	Anno: 1 *	Alfabetica: CAPOM -CENS
Materia: Informatica*	Tipologia periodo: Semestre	Numero Settimane: 14

Lezione

Numero Ore Totali: 50

Docente: MEZZALAMA MARCO

Note:

Numero Blocchi:

Squadra	Numero Blocchi	Docente
-	<input type="text" value="0"/>	<input type="text"/>
-	<input type="text" value="0"/>	<input type="text"/>
-	<input type="text" value="0"/>	<input type="text"/>

Esercitazioni in aula

Numero Ore Totali: -

Docente:

Note:

Numero Blocchi:

Squadre: Non sono previsti blocchi di tipo esercitazione aula

Esercitazioni in laboratorio

Numero Ore Totali: 20

Docente: MEZZALAMA MARCO

Note:

Numero Blocchi:

Squadra	Numero Blocchi	Docente
SQUADRA 1	<input type="text" value="0"/>	<input type="text"/>
SQUADRA 1	<input type="text" value="0"/>	<input type="text"/>
SQUADRA 1	<input type="text" value="0"/>	<input type="text"/>
SQUADRA 2	<input type="text" value="0"/>	<input type="text"/>
SQUADRA 2	<input type="text" value="0"/>	<input type="text"/>
SQUADRA 3	<input type="text" value="0"/>	<input type="text"/>
SQUADRA 3	<input type="text" value="0"/>	<input type="text"/>
SQUADRA 3	<input type="text" value="0"/>	<input type="text"/>

Figura 2.6. Gestione blocchi

Una volta che tutti i blocchi sono allocati e il numero delle ore rimanenti è azzerato, l'utente abilitato dovrà specificare l'aula per ogni blocco e potrà procedere alla validazione dell'Orario sul Calendario Accademico, se non si presenteranno eventuali **warning**, che spiegheremo di seguito, potrà procedere alla pubblicazione.

Un'altra area importante è la **Gestione Calendario** dove l'utente ha la possibilità di visualizzare e modificare il calendario dell'Orario pubblicato e può caricare tutti i corsi che richiedono l'utilizzo di una risorsa per un numero limitato di settimane (e comunque inferiore a cinque settimane) o per altri eventi (per esempio concorsi, workshop, corsi interni per il personale interno). I filtri a disposizione sono: la *Sede*, il *Locale*, la *Dimensione*, l'*Area*, la *Tipologia* e l'*Aula*, come mostrato in Figura 2.7.

Figura 2.7. Gestione Calendario

Un'altra area molto importante è la **Visualizzazione Log e operazioni** dove vengono visualizzate le operazioni di modifica e di cancellazione da Calendario, tramite dei *Filtri di ricerca*: *Titolo materia*, *Aula*, *Tipo di operazione (modifica o cancellazione)*, Figura 2.8.

Data	Ora Inizio	Ora Fine	Aula	Materia	Alfabetica	Docente Titolare	Tipo Operazione	Data Inserimento	Utente
03/05/2018 00.00.00	11:30	14:30	LED 7	Classificazione e interpretazione di dati biomedici	AA - LZ	BALESTRA GABRIELLA	ELIMINAZIONE	24/04/2018 18.47.35	BALESTRA GABRIELLA
03/05/2018 00.00.00	11:30	14:30	LED 8	Classificazione e interpretazione di dati biomedici	MA - ZZ	BALESTRA GABRIELLA	ELIMINAZIONE	24/04/2018 18.47.30	BALESTRA GABRIELLA
26/04/2018 00.00.00	17:30	19:00	29B	Classificazione e interpretazione di dati biomedici	AA - LZ	BALESTRA GABRIELLA	ELIMINAZIONE	23/04/2018 10.54.51	BALESTRA GABRIELLA

Figura 2.8. Visualizzazione Log Operazioni

Vengono però elencate le operazioni di cancellazione/modifica eseguite da parte dei docenti, ma non da parte di tutti gli utenti abilitati all'utilizzo del software, dall'ultima

eseguita alla prima eseguita anche negli anni precedenti.

L'ultima area abbastanza importante è la **Gestione richieste e conflitti** dove vengono elencati i conflitti tra le materie (ovvero gli Insegnamenti) e gli eventi *puntuali* (per esempio Meeting, Manutenzione, Altri eventi didattici), ma solo dopo la pubblicazione dell'Orario e non sono strettamente legati alla generazione dell'Orario; infatti, non vengono visualizzate le eventuali sovrapposizioni tra le materie comuni dei vari Corsi di Studio, oppure la doppia allocazione aula e laboratorio, per esempio nel caso dell'utilizzo a settimane alterne consentito dalla Commissione oraria. Queste sovrapposizioni non possono essere verificate da nessuna parte a meno che non si vada a controllare la singola materia.

2.3.1 Problematiche nell'applicativo web

A differenza della scheda online, il software **Gestione Orari** è dotato di un manuale utente [3], ma che non riporta tutte le voci e le aree messe a disposizione per gli utenti abilitati e ad oggi non è mai stato aggiornato. Nelle varie schermate l'utente abilitato non dispone di tasti *help in linea* per guidarlo nella compilazione.

Il software, durante l'inserimento dei dati, evidenzia una serie di **messaggi di errore** che però non impediscono all'utente di costruire e pubblicare l'orario.

Il primo warning che si presenta nella fase di costruzione dell'Orario è dato da un messaggio che segnala che un "blocco è andato in conflitto con il template orario di altri Corsi di Laurea già pubblicati sul calendario dell'Ateneo", Figura 2.9.

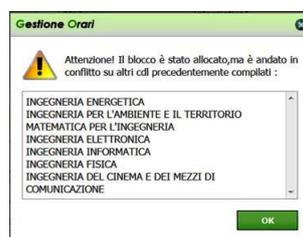


Figura 2.9. segnalazione blocco in conflitto

Questo accade perché, come specificato nel paragrafo precedente, una materia può appartenere a diversi Corsi di Laurea, di conseguenza non si riesce a capire se effettivamente si tratta di una "sovrapposizione" corretta o sbagliata dall'utente.

Un altro problema è legato al messaggio di errore che si ottiene quando si modifica il numero dei blocchi legati alla materia nella sezione *Gestione blocchi*, qualunque sia il numero di blocchi aggiunti o tolti, segnala sempre che si è *superato il 5% dell'offerta formativa*, Figura 2.10.

Una volta caricati tutti i blocchetti delle Materie nel template orario, l'utente abilitato può procedere alla validazione. Il software, in modo automatico, effettua dei controlli prima di validare il template orario, ma gli unici messaggi di warning che si possono presentare sono legati ad un blocco di una materia che si è dimenticato di allocare (**ATTENZIONE! IMPOSSIBILE VALIDARE BLOCCO NON ALLOCATO materia**), oppure ad un blocco

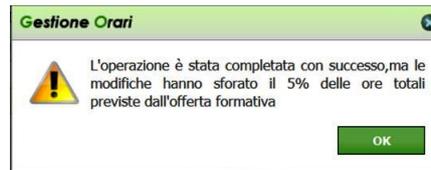


Figura 2.10. Segnalazione sfiorato il 5%

di una materia alla quale si è dimenticato di assegnare un'aula (*ATTENZIONE! IMPOSSIBILE VALIDARE AULA MANCANTE materia*); questo perché il software non crea in modo automatico l'orario ma verifica solo che i dati inseriti siano corretti.

Un'integrazione del software **Gestione Orari** è il software **Orari di lezioni e prenotazioni aule** [4] che è uno strumento messo a disposizione del docente che gli permette la prenotazione (ma al momento è stata disabilitata) e il rilascio degli spazi dedicati alla sua didattica. Attualmente solo alcuni docenti lo utilizzano per cancellare i loro blocchi da Calendario, e la cancellazione ha effetto immediato e non viene inviata nessuna comunicazione (per esempio una mail, che al momento è stata disabilitata) agli uffici di gestione degli spazi dell'Area Logistica e dell'Area IT, ma viene riportata nell'Area **Visualizzazione log operazioni**.

Questo comporta un problema da non sottovalutare perché i Referenti delle risorse si ritrovano con delle modifiche di orario non segnalate nel momento in cui il docente effettua l'operazione di cancellazione; ma solo accedendo all'Area **Visualizzazione log operazioni** può ritrovarle. In quest'area però non vengono visualizzate le modifiche e/o cancellazioni eseguite da tutti gli utenti abilitati al software, e se per errore viene cancellato o modificato un blocchetto da Calendario, da un Referente delle risorse, questo non viene segnalato e riportato da nessuna parte.

Capitolo 3

Software disponibili per la generazione automatica dell'orario

Il problema dell'orario ovviamente non riguarda solo il *Politecnico di Torino*, ma è un problema che viene affrontato dalla maggior parte delle persone coinvolte, di ogni istituto di ordine e grado, come un vero e proprio *rompicapo*.

Ma se digitiamo su un motore di ricerca troviamo, ad oggi, un notevole numero di software per automatizzare e semplificare tutta la generazione dell'orario. Molti dei software disponibili sono di tipo commerciale con versioni di prova gratuite ma limitate, altri invece sono totalmente gratuiti.

Mostriamo qui un elenco di questi software indicando solo alcune caratteristiche per dare un'idea dello stato attuale dell'offerta che c'è in Italia e nel mondo: da semplici file in excel a interfacce web che permettono facilmente l'inserimento dei dati e l'elaborazione automatica dell'orario. Indichiamo prima quelli per l'orario scolastico delle scuole medie e superiori e poi quelli per l'orario universitario.

Le soluzioni proposte sono sicuramente valide dal punto di vista grafico, ma dal punto di vista della qualità delle soluzioni occorre fare delle verifiche, perché le regole e i vincoli proposti per un'università non è detto che siano validi per un'altra università; inoltre gli algoritmi risolutivi spesso sono nascosti e non documentati e quindi non sono confrontabili tra di loro.

3.1 Software per l'orario scolastico

Tra i software per la generazione dell'orario scolastico delle scuole medie e superiori sono presenti dei semplici file in Excel per compilare l'orario scolastico, per gestire le ore in compresenza, per pianificare i consigli di classe, e per controllare velocemente e facilmente i dati inseriti, tra questi abbiamo: **FILEOrario** scritto con Excel 97 (software freeware [40]) e **Orario-EducazioneScuola** [38].

Esiste anche un semplice software gratuito, **WinOR** [54], sviluppato in Visual Basic 6 per l'applicazione e in Excel per la gestione e la creazione dell'orario scolastico; è rilasciato in versione shareware, ma la sua distribuzione è permessa solo se non viene alterato in

alcun modo. Infine indichiamo il software **ORARIO** [39], del 1995, dove nella documentazione relativa al software viene descritto il problema di generazione dell'orario come un problema di "intelligenza artificiale" che può essere risolto utilizzando una "strategia di backtracking" e la "tecnica degli algoritmi genetici". Il programma trova una soluzione e chiede se si desidera cercarne un'altra, quindi permette di confrontare diverse soluzioni e di scegliere quella più idonea. Per produrre un orario diverso, basta cambiare qualche parametro oppure invertire l'ordine dei dati inseriti.

Qui di seguito indichiamo i software che presentano un'interfaccia grafica semplice ed intuitiva per l'inserimento dei dati e per tutte le altre funzioni che servono per generare automaticamente l'orario scolastico.

Orarioweb.it

È un programma [42] che permette di generare automaticamente e completamente **online** l'orario scolastico della scuola primaria, secondaria di primo e secondo grado. Non occorre nessuna installazione e funziona con qualunque sistema operativo. Esiste una versione di prova con delle limitazioni e una versione di base gratuita; in ogni caso occorre registrarsi sul sito. Il software non viene venduto ma concesso in licenza d'uso non esclusiva. Vengono forniti anche i costi per l'attivazione di determinate funzioni, per l'amministrazione di server, per gli aggiornamenti e i backup dei dati. Le istruzioni sono tutte online.

Grazie ad un'interfaccia grafica si inseriscono tutti i dati: il quadro settimanale, le classi, i docenti, le lezioni e i vincoli. Quest'ultimi vengono suddivisi in due gruppi: vincoli "globali" relativi all'istituto (come per esempio l'uscita di tutte le classi all'ultima ora di sabato) e vincoli "relativi ai docenti" (come per esempio il giorno libero o le ore in cui il docente può essere impegnato in altri istituti). Non ci sono però indicazioni in merito alle risorse (aule, laboratori, palestre).

Dopo l'inserimento di tutti i dati si procede alla generazione automatica dell'orario che dura pochi secondi, secondo quanto viene segnalato; c'è la possibilità di correggere manualmente l'orario con una specifica applicazione; e una volta completato l'orario può essere salvato online e stampato in formato pdf.

Il proprietario del software fornisce anche i tempi di elaborazione per la generazione automatica: 1) "circa 36 secondi, per un orario con 13 classi, 37 docenti e 309 lezioni, con vincoli (giorni liberi e docenti in servizio presso altri istituti), con compresenze e una classe articolata; 2) circa 124 secondi per un orario con 51 classi, 97 docenti e 1513 lezioni, con vincoli (giorni liberi e docenti in servizio presso altri istituti), con compresenze e molte lezioni di più di un'ora".

Non è presente nella piattaforma MEPA (Mercato Elettronico della Pubblica Amministrazione) perché i costi sono inferiori ai 1000 Euro.

Orario Facile 10

Questo programma viene proposto da una ditta di Torino, la *Mathema software* [41], frutto di una collaborazione con diverse realtà scolastiche. La prima versione risale a prima del 2002, e c'è la possibilità di scaricare i manuali delle versioni precedenti. È un software

per sistemi operativi Windows a 32 e 64 bit. È presente sulla piattaforma MEPA perché i costi sono superiori ai 1000 Euro. La licenza del software a pagamento è illimitata e può essere utilizzata su una o più postazioni di lavoro, viene data tutta la documentazione e viene fornito un servizio di assistenza per un anno, mentre gli aggiornamenti sono gratuiti.

Esiste una versione di prova gratuita, senza limitazioni e con la possibilità di scaricare dei file di esempio con i dati relativi a due scuole che possono essere modificati.

Il software è stato progettato in base esigenze delle scuole italiane, ma si dichiara adattabile per qualsiasi tipo di istituzione scolastica.

Non ci sono indicazioni su come è stato sviluppato il software, segnala solo che "*sfrutta sofisticate tecnologie algoritmiche appositamente adattate alle complesse esigenze di definizione dell'orario. L'ottimizzazione automatica dell'orario è guidata da un insieme di parametri di ottimizzazione che identificano in modo chiaro e lineare le possibili criticità di un orario (quali le ore buche, l'eccesso di ore giornaliere, distribuzione indesiderata delle lezioni), secondo delle priorità che vengono stabilite dall'utente*" [41].

Tramite l'interfaccia grafica si inseriscono tutti i dati, le caratteristiche e i vincoli non solo delle classi, dei docenti, delle materie ma anche delle aule e della sede scolastica (nel caso di più sedi). Vengono considerate: la classe semplice o le sottoclassi; le lezioni tenute da un singolo docente o da più docenti; le lezioni svolte ad una singola classe o a più classi; più sedi (quindi si possono assegnare i tempi per gli spostamenti del docente da una sede all'altra); le aule, i laboratori e le palestre. Quindi l'utente può passare da una gestione chiamata a "*pool*" (nel caso di molte risorse a disposizione) alla gestione "*singola*" (nel caso voglia assegnare ad ogni lezione un'aula).

Viene introdotto il concetto di "*peso didattico per le materie*" che permette di equilibrare il carico didattico durante la giornata.

Anche con questo software l'utente può modificare e correggere manualmente il risultato ottenuto, visualizzare le tabelle e stampare l'orario.

ZonabitOrario

È un altro software realizzato in Italia dalla ditta Zonabit Sistemi Srl [55], con sede a Bolzano e a Roma, che afferma di essere utilizzato da un numero crescente di scuole italiane, ma sul sito non è presente nessuna indicazione o lista. Dal 2006 si occupa della generazione automatica dell'orario scolastico ed è una distribuzione commerciale con diversi *pacchetti* da scegliere: dal programma completo senza scadenza e senza limitazioni di classi ma con assistenza limitata per un anno scolastico, fino al pacchetto completo con assistenza anche per l'anno successivo. È anche possibile ottenere una licenza con durata limitata fino a 6 mesi e con costo diverso a seconda del numero di classi di cui è composta la scuola.

Esiste una versione di prova gratuita che permette di configurare e di elaborare un orario completo e definitivo, ma con alcune limitazioni: mancano alcuni vincoli (per esempio non consente la sostituzione degli assenti); utilizza una sola CPU per computer che rende più lenta l'elaborazione dell'orario; non dispone della "*ricerca ottimale delle aule*", non ci sono la stampa e l'utilizzo di un server cloud. La versione di prova vale per sistemi operativi Windows e Mac IOS (anche se è una versione beta). La licenza completa del software permette l'elaborazione dell'orario tramite un server remoto, Cloud server, messo

a disposizione dalla ditta. Si può fare richiesta anche di ulteriori vincoli che non sono presenti nella versione proposta. ZonabitOrario è in grado di importare automaticamente i dati da file prodotti da altri software.

Sul sito del software sono presenti numerosi manuali con tutte le istruzioni per ogni esigenza e numerosi video tutorial che spiegano passo per passo come configurare l'orario della propria scuola. Nella guida rapida di questo software vengono descritti i passi da eseguire e tutti i campi da riempire; inoltre nelle varie schermate dell'interfaccia web sono presenti i tasti con il simbolo "?" per l'aiuto in linea.

La configurazione dell'orario avviene in due fasi. Nella prima fase si inseriscono tutti i dati e viene eseguita una prima elaborazione che dura pochi minuti; successivamente si introducono i vincoli, e l'elaborazione continua affrontando un vincolo dopo l'altro. Qui è presente un pannello chiamato "*Problemi critici*" dove vengono mostrati i vincoli problematici che possono essere modificati dall'utente, mediante un "*indice di complessità/frequenza*" che deve diminuire, fino a quando non si ottiene l'orario finale. I tempi di elaborazione, per poter risolvere alcuni vincoli, sono maggiori e variano da mezz'ora a un'ora a seconda della complessità delle scuole.

Una funzione importante di questo software, che non è presente in quelli precedenti, è l'"*assegnazione delle aule*" una volta definito l'orario: si cercano di allocare in modo ottimale le classi nelle aule disponibili. Con due possibili soluzioni: 1) con l'"*assegnazione delle aule ai professori*" e quindi gli alunni si spostano nell'aula dove il professore insegna, 2) con l'"*utilizzo ottimale delle aule disponibili*" per le scuole che hanno un numero inferiore di aule rispetto alle classi. La procedura può essere eseguita mantenendo l'orario trovato all'inizio, oppure modificando l'orario, ma non si possono modificare i vincoli per cercare di ottimizzare l'assegnazione delle aule.

Nella seconda fase si inseriscono gli ulteriori vincoli delle classi, dei professori, e quelli più rari e specifici, ma ad ogni vincolo aggiunto si consiglia di fare sempre un'elaborazione e di controllare il risultato ottenuto.

Si può configurare un orario provvisorio adatto per le prime settimane dell'inizio dell'anno scolastico; gestire gli assenti e visualizzare le sostituzioni.

EDT

Questo software risulta essere uno dei più diffusi in Italia sia per l'organizzazione e la programmazione dell'orario delle lezioni sia per la gestione delle risorse scolastiche. È presente sul mercato dal 1996 e in Italia esiste una ditta proprio in provincia di Torino, a Chieri [43]. In Italia 1400 istituti tra scuole primarie, scuole secondarie di primo e secondo grado lo utilizzano, 114 sono in Piemonte. Esiste anche una versione per la Francia, per il Belgio e per la Svizzera.

Ci sono due tipi di licenze: 1) "*a tempo indeterminato*", è una licenza non-esclusiva e non cedibile, che prevede la sola versione presente in commercio al momento dell'ordine e include la documentazione in formato PDF e un anno di assistenza ma non include gli aggiornamenti alle versioni successive; 2) "*per un anno*" è una licenza con le stesse caratteristiche della precedente ma include gli aggiornamenti disponibili durante il periodo richiesto.

I costi variano a seconda del tipo di licenza scelta e del numero di classi che si vogliono gestire (15, 30 o infinite). È possibile scaricare una versione gratuita del software senza limiti di inserimento delle classi ma con una durata limitata di quattro settimane, valida per i Sistemi Operativi Windows (dal Server 2008 al 10) e per i MAC.

È scaricabile la "*Guida pratica*" relativa alla versione del 2017, molto dettagliata che mette l'utente in condizione di utilizzare facilmente il software; nelle varie schermate per l'inserimento dei dati l'utente dispone anche di due tasti, "F1" e "?", per accedere alla guida in linea; e sul sito sono presenti anche dei video tutorial. La ditta propone agli utenti due tipi di formazione: quella di base per la preparazione dell'orario, la creazione della base dati e i parametri, e per la padronanza degli strumenti di calcolo e di ottimizzazione; quella avanzata per la gestione quotidiana dell'orario, i consigli di classe, le sostituzioni e le comunicazioni.

Non indichiamo ulteriori dettagli del programma perché la ditta propone un altro software per la gestione e la programmazione dell'orario per le scuole superiori e per le università, **HYPERPLANNING** che vedremo nel paragrafo successivo.

UNTIS

Si tratta di un software europeo realizzato dalla ditta fondata dall'ingegnere Bernhard Gruber e dal dottore Heinz Petters che, affascinati dal computer e dalla programmazione, nel gennaio del 1970 programmarono in Assembler, su un grande computer IBM S/360, il primo programma informatico per la creazione di orari scolastici [52]. Nel corso degli anni il software viene continuamente aggiornato rispettando le esigenze delle scuole e tenendo conto dell'evoluzione della tecnologia informatica. Ad oggi conta più di 20.000 installazioni in diversi Paesi Europei (Germania, Svezia, Danimarca, Finlandia, Olanda, Spagna, Portogallo, Francia, Italia) ma anche in America del Sud e Centrale e in Sudafrica. È tradotto in 25 lingue. La distribuzione e l'assistenza in Italia sono garantite dalla ditta PLANSCUOLA srl. È un software in grado di realizzare in meno di tre minuti una proposta di orario anche per gli istituti più complessi.

Sono disponibili tre licenze: 1) la "*UNTIS DEMO*" che è gratuita, è completa e contiene i dati di base (classi, docenti, materie e aule) e i dati degli orari che possono essere modificati e si possono elaborare nuovi orari; ma è limitata alla gestione di 7 classi e la funzione di salvataggio è disabilitata; oppure c'è la versione della durata di soli tre giorni con gestione illimitata delle classi; 2) la "*UNTIS TEST*" che è limitata a tre mesi ma con tutte le funzionalità del programma e un costo fisso; 3) la "*UNTIS*" che è illimitata ma i costi sono in funzione del tipo di scuola e del numero di alunni.

Esistono altri applicativi UNTIS: "*UNTIS EXPRESS*" versione ridotta e semplice; "*WEB UNTIS*" per il registro di classe elettronico, per la prenotazione delle aule, per la gestione degli alunni; "*UNI UNTIS*" per le Università; "*UNI MULTI USER*" per l'utilizzo contemporaneo da parte di più utenti. Può essere installato su PC dotati di Sistema Operativo Windows. L'algoritmo, sviluppato da più di 30 anni e aggiornato ogni anno (in primavera, a Salzburg, viene presentata la nuova versione), calcola più orari e l'utente può scegliere quello che ritiene più adatto.

Sul sito di PLANSCUOLA si possono scaricare diversi documenti informativi, tra i quali il manuale di utilizzo di UNTIS 2018 per la gestione della scuola.

Esiste la versione per l'Università, "**UniUntis**", ma vengono solo riportati i dati dell'Università bavarese (Ostbayrische Technische Hochschule, OTH, Regensburg) dove è stato utilizzato il software, ed è composta: da 8 facoltà, oltre 10.000 studenti e 223 docenti e circa 370 assistenti universitari.

Anche questo software è facile da utilizzare e un aiuto in linea permette l'inserimento di tutti i dati delle classi, delle materie, dei docenti e delle aule, e tutti i desiderata per tutti i dati di base. Qui viene introdotto il concetto di "*parametro ponderato*": l'utente può stabilire le priorità dei desiderata per le classi, per i docenti, per le materie e per le aule. Questo software propone quattro strategie di "*elaborazione dei dati*": 1) "*pianificare tutte le ore in modo automatico*", 2) "*impostare parti dell'orario in modo manuale*", 3) "*apportare modifiche manuali una volta terminata l'elaborazione*", 4) "*intervenire durante l'elaborazione in corso introducendo informazioni che migliorino le aspettative*" [52].

È dotato di uno strumento di "*diagnosi*" per visualizzare gli eventuali errori di immissione dei dati o per segnalare eventuali problemi nell'orario ottenuto con la possibilità di correggerli facilmente. Anche questo software permette di stampare l'orario e dare informazioni statistiche su tutta l'attività didattica.

Tra i "*moduli speciali*" il software propone come "*gestire le sostituzioni*", "*pianificare le cattedre*" (cioè assegnare le materie ai docenti e gli incarichi speciali) e "*pubblicare gli orari*"; e ulteriori moduli adatti per l'Università come la "*pianificazione dei corsi*" per gli studenti che possono scegliere i corsi, l'"*orario alunni*" per gli studenti che hanno poche possibilità di scelta ma molti corsi obbligatori e infine l'"*orario settori*" per l'orario decentralizzato e creato dai vari dipartimenti: una volta elaborati gli orari parziali è possibile accorparli nell'orario generale e quindi fare un'ulteriore elaborazione, ottimizzando l'orario di ogni docente coinvolto nelle attività nei vari settori.

FET - Free Timetabling Software

Questo software è l'unico completamente gratuito e open source, il sorgente è disponibile e può essere copiato, utilizzato e modificato con licenza GNU/GPL [44]. La prima versione è nata nel 2002. È scritto con il linguaggio di programmazione C++ usando il framework di applicazioni multi piattaforma Qt. Ha un algoritmo che genera automaticamente l'orario in meno di due minuti e consente assegnazioni semiautomatiche o manuali. Può essere compilato su qualsiasi sistema operativo GNU/Linux, Windows e Mac.

Formato XML modulare e flessibile per il file di ingresso; importa ed esporta file dal formato CSV; gli orari prodotti vengono esportati nei formati HTML, XML e CSV.

Tra le principali caratteristiche di FET è possibile strutturare gli studenti in gruppi o classi, in sottogruppi e anni di corsi, sovrapponibili e non sovrapponibili; è possibile definire i *vincoli di tempo* (come per esempio per il cambio di edificio sia per gli studenti che per i docenti) e i *vincoli di spazio* (come per esempio per l'aula non disponibile, per un'aula preferita per materia o per un'attività con caratteristiche particolari). Tutti questi vincoli hanno un *peso percentuale* così l'utente può decidere la priorità tra quelli che sono obbligatori e tra quelli che si possono rilasciare.

In rete [44] sono disponibili le pubblicazioni dello sviluppatore principale, Liviu Lalescu, le funzioni ufficiali disponibili in varie lingue e le versioni personalizzate.

3.2 Software per l’orario universitario

Per l’orario universitario riportiamo un elenco di software di tipo commerciale che presentano più o meno le stesse caratteristiche, ma ognuno si proclama essere il migliore software per la creazione automatica e la gestione dell’orario; e un unico software gratuito e open source che è il primo che presentiamo.

UniTime

È un software distribuito gratuitamente, con licenza open source, per la pianificazione e la generazione degli orari e per la programmazione degli esami universitari [51]. La piattaforma è implementata utilizzando Java J2EE e il database SQL.

Il sistema è frutto di una collaborazione tra personale e studenti delle Università del Nord America e dell’Europa. In particolare è stato applicato con successo alla Purdue University (dal 2005) strutturata da: 9000 lezioni, 570 aule, 39000 studenti, 190000 richieste di corsi e 2400 esami. Ma può essere personalizzato per una varietà di problemi e di esigenze di orario universitari.

L’inserimento dei dati, da parte dell’utente, è facile ed intuitivo e può essere eseguito in modo manuale o in modo completamente automatico. Possono essere impostati su vari livelli i requisiti e le preferenze per la struttura del corso (orario, classe) e delle lezioni (lezione frontale, esercitazioni, laboratori). Vengono considerati due tipi di vincoli base: i "*vincoli di risorse*" quando una lezione può essere insegnata da un solo docente in un’aula particolare e in un determinato momento e i "*vincoli di gruppo*" quando due sezioni della stessa lezione non possono essere assegnate nello stesso momento, o che alcune lezioni devono essere assegnate una dopo l’altra.

Il software identifica eventuali incongruenze e problemi nei dati di input, fornisce una soluzione di orario completamente automatizzata e consente inoltre di apportare modifiche interattive tramite suggerimenti.

Su internet sono disponibili: il software, la libreria dei vincoli, tutta la documentazione del software e del problema applicato alla Purdue University, diversi set di dati di riferimento (in un formato XML facilmente leggibile) per l’orario del corso, la gestione dello studente e la programmazione degli esami. È presente anche una "*demo online*" che contiene una suite di test sviluppati al *Woebegon College*, creata per risolvere il problema dell’orario su un esempio facilmente gestibile.

La ricerca per migliorare questo software è in continua evoluzione, sono presenti pubblicazioni e presentazioni e anche una tesi; per ulteriori dettagli si possono leggere gli articoli [18], [19], [20] (quest’ultimo presentato alla Conferenza PATAT del 2016).

Nel marzo 2015, UniTime è diventato un progetto sponsorizzato dalla Fondazione *Apereo* e alla Conferenza *Open Apereo* 2018, che si è svolta a Montreal a giugno, sono stati presentati i vari aspetti del software: "*lo stato del progetto*" dalla release corrente

4.2 a quella in uscita 4.3 e a quella futura 4.4 [21], *"l'internazionalizzazione"* [22] e *"la programmazione studentesca alla Pordue University"*.

Tutta la documentazione della versione di **UniTime** 4.3 è disponibile online [51].

HYPERPLANNING

HYPERPLANNING è la versione evoluta del software EDT, dal 1993 è il software di riferimento nella gestione di tutta l'attività didattica per le scuole superiori, per i centri di formazione e per le università [47]. Viene utilizzato in Italia, Francia, Belgio, Svizzera, ma anche in Cameroun, Congo e Tunisia.

Tutti gli applicativi possono essere scaricati gratis e testati senza limiti di inserimento dei dati per un periodo di otto settimane. Può essere utilizzato su Sistemi Operativi Windows da 32 e 64 bit e su Mac.

Sono disponibili due licenze: 1) di *"acquisto"* che comprende l'ultima versione del software con la documentazione in formato elettronico, un anno di assistenza e gli aggiornamenti; 2) di *"un anno"* che comprende le stesse caratteristiche della precedente ma con durata limitata ad un anno a partire dalla data dell'ordine.

I costi variano a seconda della licenza e a seconda dal numero degli utenti (*"monoposto"* o *"rete"*) che sono autorizzati per la gestione, per la consultazione e per la modifica dell'orario (docenti, studenti, personale di Ateneo); e a seconda delle richieste supplementari quali server o Web service e Cloud (*"rete+.net"*).

Infine c'è anche la possibilità di avere, a pagamento, una formazione specifica del software o presso la sede della ditta o presso la propria Università.

Le principali caratteristiche del software sono: *"la gestione degli orari e degli esami, la gestione delle assenze, la gestione delle aule, i resoconti (conteggi delle ore e delle attività annullate), la valutazione degli studenti e il controllo della frequenza, la pubblicazione in internet, l'importazione ed esportazione automatizzate da MySQL, Oracle, SQL, LDAP o foglio elettronico"* [47].

Grazie ad un'interfaccia estremamente intuitiva gli utenti possono compilare i calendari, gli orari, le risorse (studenti, docenti, aule) e i vincoli, in modo automatico oppure facendo una copia da uno esistente e ha sempre la garanzia di non occupare mai due volte una risorsa nella stessa fascia oraria. L'utente può inserire i vincoli che devono essere strettamente rispettati durante un *"piazzamento automatico"*; durante un *"piazzamento manuale"* questi vincoli vengono segnalati, tramite la funzione *"diagnostica di un'attività"*, e l'utente può decidere se ignorarli o meno.

Nell'inserimento delle attività, l'utente può scegliere due modi: 1) se l'orario è già stato realizzato e bisogna solo ricopiarlo, l'utente può riportare in una specie di griglia tutte le attività che vengono create e piazzate nello stesso momento; 2) l'utente può creare delle attività *"sospese"* e piazzarle in un secondo momento, scegliendo la loro posizione con l'aiuto del software oppure lasciando al programma il compito di piazzarle in automatico rispettando i vincoli definiti.

I vincoli che si possono definire con il software sono [47]:

- "*Vincoli legati alle materie*": viene definita l'incompatibilità della materia, ovvero si impedisce di avere più volte la stessa materia nella stessa giornata e si stabilisce un massimo numero di ore giornaliere e di ore settimanali per materia;
- "*Vincoli legati ai docenti/ai corsi*": vengono indicate le indisponibilità dei giorni e delle fasce orarie del docente e del corso e vengono definiti il massimo numero di ore giornaliere e di ore settimanali per i docenti e per i corsi e il massimo di giorni di presenza sulla settimana e sull'anno solo per i docenti;
- "*vincoli legati alle aule, alle sedi*": vengono assegnate le indisponibilità dei giorni e delle fasce orarie nelle quali l'aula non deve essere occupata e vengono fissati i tempi di trasferimento tra le sedi oppure il numero massimo di cambi di sede al giorno;
- "*Vincoli legati al funzionamento dell'istituto*": vengono fissate la pausa pranzo, in una determinata fascia oraria, e le pause tra le attività.

Il software dispone di due "*piazzamenti*": uno automatico e uno manuale.

Il "*piazzamento automatico*" permette all'utente di inserire le attività senza piazzarle in nessuna fascia oraria, sarà poi il software a piazzarle automaticamente rispettando i vincoli definiti in precedenza. Durante la procedura automatica gli utenti sono in fase di consultazione e tutte le attività che non sono bloccate possono essere spostate.

L'utente dispone di varie opzioni per il "*piazzamento manuale*" delle attività: a) può "*interrompere il calcolo al primo scarto*" quindi manualmente cerca di capire il perché dello scarto e cerca di risolverlo; b) può "*trattare le attività fisse solamente sul periodo attivo*" quindi le attività fuori dal periodo attivo non vengono considerate; c) può utilizzare la "*soluzione attività scartate*" ma solo se lavora su poche attività, perché nelle prime fasi dell'elaborazione si rallenta il calcolo; d) infine può scegliere di "*non modificare le attività non estratte*" perché se l'attività non viene selezionata il software di default sposta le attività non estratte e non bloccate.

Dopo il piazzamento automatico l'utente può modificare, se è necessario, i cosiddetti "*criteri di piazzamento*", aumentando il valore assegnato di default: può "*favorire le mezze giornate libere*" al docente; può "*ridurre i buchi*" nell'orario dei docenti e dei corsi; può "*favorire le giornate libere*" ai docenti.

A fine calcolo l'utente avrà un elenco delle attività piazzate e un elenco di quelle scartate e il software permetterà di risolvere quelle scartate (ma solo su un piccolo numero di attività) esplorando nuove combinazioni; e infine l'utente potrà risolvere manualmente l'attività scartata.

Tra le novità dell'ultima versione del software, HYPERPLANNING 2018, c'è la "*gestione delle aule*" che consente: "*una nuova ripartizione automatica delle aule*" dove si possono creare dei gruppi di aule in base alle caratteristiche che hanno in comune; "*un'assegnazione delle aule preferenziali per i docenti e per i corsi*"; "*una ricerca o richiesta di aule*" ovvero non si prendono in considerazione le aule che hanno già in elaborazione una richiesta di prenotazione; e se un utente desidera occupare un'aula alla quale non ha accesso, può effettuare una richiesta di prenotazione (per esempio i docenti dalla loro Area o gli utenti dal Client) e può aspettare finché non gli viene assegnata o rifiutata.

MIMOSA

Questo software è stato sviluppato da una ditta Finlandese, la *Mimosa Software Ltd* [48], che da oltre 15 anni collabora con tutti i tipi di scuole, università e aziende; propone anche soluzioni per la programmazione degli esami o di altri eventi, e propone prenotazioni di sale, attrezzature e altre risorse. Dalle informazioni sul prodotto, risulta essere utilizzato in più di 100 università ma in Italia solo tre istituti lo utilizzano (Bergamo, Merano e Venezia).

La versione commerciale del software ha due licenze: 1) la "*Mimosa Single-User License*" che consente di installare e utilizzare il software solo su un computer, include gli aggiornamenti gratuiti a vita e un supporto gratuito, e ha un costo fisso; 2) la "*Mimosa Site License*" che consente di installare e di utilizzare il software su qualsiasi computer o rete dell'organizzazione e di condividere file creati da utenti diversi, include gli aggiornamenti gratuiti a vita e il supporto gratuito, ma il suo costo si basa sulla dimensione media del numero degli studenti.

Sul sito c'è la possibilità di scaricare (senza registrarsi): 1) una versione gratuita, la "*Mimosa Small School Edition (Freeware)*" che consente di creare e programmare fino a 80 voci e senza limiti di tempo, e i file creati con questa versione consentono di gestire oltre 8.000 risorse ed eventi (circa 300.000 orari, 30 slot in un giorno di una settimana di 7 giorni, 255 settimane); 2) una versione di prova, la "*Mimosa Trial Version*", che può essere testata per trenta giorni senza limitazioni di capacità. In tutte le versioni del software viene installato di default un file di esempio che contiene gli orari settimanali personalizzati per 1.000 studenti. Tutte le versioni sono compatibili su Sistemi Operativi da Windows 98 a Windows 10, su Mac OS e su Linux. Tutte le istruzioni e i tutorial del software si trovano online, oppure si scaricano dei file in formato PDF. Ma è anche possibile avere una formazione per gli utenti per utilizzare al meglio i vari strumenti offerti dal software [48]. Il software ha oltre 800 voci menu e diverse opzioni che consentono di pianificare l'orario manualmente e automaticamente, utilizzando dei sofisticati algoritmi di ottimizzazione e strumenti di pianificazione; ma non vengono indicati quali tipi di algoritmi sono stati implementati.

Se si creano gli orari "*manualmente*", il software guida l'utente a pianificare, a rimuovere e a spostare le lezioni negli orari, ad annullare e a ripetere tutte le modifiche e a tornare in qualsiasi momento alla fase precedente. L'utente quindi è guidato passo dopo passo (tramite un'interfaccia point-and-click) per evitare che commetta errori e può tenere traccia delle azioni fatte e delle modifiche apportate agli orari.

Se si creano gli orari "*automaticamente*", in pochi minuti si ottengono degli orari compatti e operativi. L'utente può sempre scegliere cosa e come ottimizzare e può combinare le attività automatiche e manuali.

Come strumenti di ottimizzazione vengono date queste possibilità: "*ottimizzazione della selezione del corso e dell'evento, soluzione di pianificazione iniziale, assegnazione ottimale delle lezioni a settimane, ottimizzazione dell'uso dell'aula, ottimizzazione rapida, pianificazione dell'ottimizzazione, ottimizzazione della selezione*" [48].

È possibile selezionare e trasferire i dati in entrata e in uscita tramite un'applicazione, chiamata "*Appunti*", oppure come file di testo. Gli orari e i report possono essere stampati

e pubblicati come pagine Web per essere consultati dagli studenti, dai docenti e dal personale. Si possono utilizzare anche i formati di file vCalendar (VCD) e iCalendar (ICS) per le più diffuse applicazioni di calendario web, MS Outlook e per il calendario del proprio cellulare.

Vengono ritrovate con facilità e visualizzate da varie angolazioni tutte le informazioni di pianificazione; una particolare vista, la "*Vista orari*" (che è suddivisa in *vista Risorse*, *vista Eventi*, *vista Settimane* e *Tabella orari*) permette di vedere tutte le interdipendenze tra gli orari, permette di spostare blocchi di eventi e di sostituire le risorse per evitare conflitti, e impedisce di effettuare doppie prenotazioni.

BEST Automated Scheduling and Timetabling

La Bullet Solution [37] è un fornitore di molti programmi di ottimizzazione, non solo in ambito universitario ma anche aziendale e ospedaliero. A differenza degli altri software non ci sono immagini grafiche di quello che il prodotto può offrire e per ottenere tutte le informazioni sul prodotto, i manuali o le brochure, e sapere i costi bisogna scrivere una mail.

Al momento non esistono uffici in Italia, ma solo in Portogallo, negli USA (a San Francisco e a Boston) a Londra, a Istanbul e ad Amsterdam.

Vengono però descritte le caratteristiche delle varie applicazioni [37], e qui ne indichiamo solo alcune:

- "*Automated Scheduling and Timetabling*": genera automaticamente la pianificazione e gli orari; combina e valuta diversi obiettivi e vincoli in base alle esigenze dell'Università, ottimizzando gli orari del personale accademico, degli studenti e delle aule. È basato su potenti algoritmi, ma non indica quali sono; è flessibile e può facilmente comunicare con altri software. Le caratteristiche principali offerte sono: l'*"ottimizzazione degli spazi fisici"* che dà la possibilità di ottenere delle aule di ricambio per allocare degli eventi esterni; l'*"eliminazione di errori"* tipici, come le sovrapposizioni nell'allocazione delle risorse; l'*"assegnazione di priorità"* a ciascun obiettivo in base alle esigenze dell'università, durante la generazione e l'ottimizzazione degli orari.
- "*Event and Resource Management*": guida l'utente nelle modifiche giornaliere delle attività e delle risorse ottimizzando i processi di gestione della propria università, grazie ad un sistema di supporto decisionale intelligente. Presenta queste caratteristiche: le *"informazioni in tempo reale"* ovvero il software monitora in tempo reale gli eventi in corso e l'occupazione delle risorse; le *"regolazioni manuali intelligenti"* ovvero quando l'utente esegue variazioni manuali sugli orari, il software indica automaticamente e immediatamente tutte le possibilità di modifiche di data e ora e delle risorse assegnate, garantendo che non si verifichino dei conflitti; la *"gestione quotidiana"* quando l'utente gestisce gli orari e tutti gli altri eventi (come workshop, esami, riunioni, prenotazioni di aule); infine la *"Pubblicazione istantanea di orari"*

quando l'utente invia, tramite mail, le bozze dell'orario ai docenti in modo che possano convalidarle, e una volta convalidate l'utente pubblica automaticamente gli orari sul web per l'intera comunità accademica.

- "*Automated Classroom Allocation*": questa funzione assegna in modo automatizzato e ottimizzato le aule per gli eventi programmati, in base alle caratteristiche dell'aula richiesta e agli obiettivi dell'università; l'"*allocazione automatica dell'aula*" viene svolta in pochi minuti e consente di alleggerire il lavoro al personale, la "*gestione automatizzata dell'aula*" permette di ottenere aule di ricambio per allocare altri eventi; la "*priorità degli obiettivi*" assegnata in base alle esigenze dell'università, permette di soddisfare sia i docenti che gli studenti; "*soluzioni efficienti*" viene generata rapidamente una soluzione ottimizzata, con l'utilizzo di potenti algoritmi, eliminando gli errori legati alle sovrapposizioni nell'allocazione delle risorse.
- "*Real-time Information Display*" è una funzione che mostra tutte le informazioni utili per il mondo accademico, come gli eventi, le attività, i promemoria, le notizie in tempo reale e le informazioni per il prossimo futuro.

La ditta sta lavorando ad un'altra applicazione, la "*Automated Exam Scheduling*": una funzione avanzata per generare automaticamente la programmazione degli esami in modo completamente ottimizzato. Combina e soddisfa i diversi obiettivi e le regole di programmazione degli esami, in base alle esigenze dell'Università, fornendo la data, l'ora, l'aula e il docente per ogni esame.

SCHEDULING STUDIO 8

La ditta LANTIV [49] propone due software sviluppati con un algoritmo di Intelligenza Artificiale: uno per la scuola superiore, "*TIMETABLING TURBO 8*" e uno per le Università e i centri di formazione, "*SCHEDULING STUDIO 8*". Entrambi adatti per la pianificazione delle lezioni, dei docenti, del personale, delle aule e delle attrezzature; controllano i conflitti e aiutano a decidere "come, quando e dove" pianificare. Possono essere utilizzati sia sui Sistemi operativi Windows sia su Mac e su Linux.

Ma il software è basato su un Cloud e viene offerto con un pagamento mensile; può essere installato su tutti i computer di cui si ha bisogno, ma il numero di utenti connessi simultaneamente al server è limitato e dipende dal costo dell'abbonamento fissato (l'abbonamento può essere cancellato in qualsiasi momento). I file su cui l'utente lavora sono gestiti e sincronizzati dai server della ditta così più utenti possono lavorare insieme sul programma e osservare i cambiamenti degli altri utenti in tempo reale. Il programma contiene diversi file di esempio che possono essere utilizzati.

Esiste una versione di prova gratuita, *Lantiv Scheduling Studio*, disponibile per un tempo limitato ed è completamente funzionante e senza restrizioni.

L'interfaccia utente, anche se presenta un design moderno ed elegante, è un po' sovraccaricata di opzioni, quindi occorre del tempo per imparare ad usare il programma; ma sul sito web l'utente può trovare una guida in linea dettagliata, suddivisa in varie sezioni [49]:

"Come iniziare" , "Lo Spazio di lavoro" , "Istituzione", "Risorse", "Attività" , "Pianificazione", "Conflitti", "Combinazioni", "Modifiche", "Report", "Stampa ed esportazione", "Importazione dei dati".

Tra le caratteristiche principali vediamo come vengono gestiti i "conflitti", ed elenchiamo qui di seguito i vari scenari di conflitto che il programma può segnalare [49].

- "Double Booked Resource": quando due attività che coinvolgono lo stesso studente, o docente o gruppo di persone, sono programmate nello stesso momento.
- "Risorsa non disponibile": quando si cerca di pianificare un'attività che coinvolge una risorsa che è stata precedentemente bloccata in un determinato momento e per determinate ore.
- "La capacità della stanza è stata superata": quando si supera una certa capacità, ovvero il numero di persone, o di gruppi o di attività che una stanza può contenere.
- "Distanza tra le stanze": quando vengono programmate due attività che coinvolgono una persona (o un gruppo di persone) in due luoghi troppo lontani tra loro.
- "Over Booked Activity": quando l'utente pianifica un'attività per più ore o periodi rispetto alla quantità pianificata.

Ogni volta che il programma rileva uno o più conflitti viene visualizzata una finestra, chiamata "Finestra dei conflitti", che elenca tutti i conflitti che si sono presentati. Questa finestra offre all'utente tre opzioni di scelta: la prima è cancellare l'ultima cosa che ha fatto e tornare allo stato precedente; la seconda opzione è quella di ignorare i conflitti, completare l'operazione e risolvere i conflitti in seguito (nel "Pannello Conflitti"); la terza opzione consiste nel risolvere i conflitti "sul posto", direttamente nella "Finestra dei conflitti" scegliendo quali attività in conflitto devono rimanere programmate e quali devono essere eliminate.

Il "Pannello Conflitti" riassume tutti i conflitti esistenti nella pianificazione, dopo che l'utente ha scelto di continuare senza eliminare le attività in conflitto. Nell'elenco dei conflitti, ogni riga rappresenta un conflitto e contiene: il titolo, il tipo di conflitto e la risorsa coinvolta, la descrizione del conflitto, l'intervallo temporale e le attività in conflitto.

TimetablePlus

Questo software è stato introdotto per la prima volta nel luglio 2007. È presente la versione *Timetable Plus v1.2* [50] facile da usare e potente. Questo software viene utilizzato per la generazione degli orari dagli istituti di istruzione superiore in Malesia dal 2007. L'obiettivo principale è soddisfare le richieste degli istituti di istruzione superiore, College, Università e i Politecnici. Sul sito vengono indicate le caratteristiche principali: "algoritmo di scheduling avanzato; impostazione di vincoli multipli; facile processo di immissione dei dati; più informazioni per l'inserimento dei dati degli utenti; altre opzioni di report di analisi; varie tecniche e opzioni per la generazione automatica; mappatura di database e risultati con sistema studente". Sono disponibili due versioni.

"*Timetable Plus Spring Lite*" è la versione gratuita, facile da installare e può essere usata anche su una pen drive; è flessibile nell'inserimento dei dati e dei vincoli. Il software offre all'utente l'opzione di posizionare manualmente le sezioni dei corsi nell'output dell'orario o di utilizzare la generazione automatica; consente di stampare orari individuali per docenti, per gruppi di studenti, giorno per giorno e per le aule; consente, ad un solo utente, di gestire il database, di inserire i dati e di generare l'uscita dell'orario.

"*Timetable Plus Spring Pro*" è la versione avanzata della versione Lite con funzioni aggiuntive di facile uso. L'utente può acquistare online questa versione con licenza illimitata e con un facile sistema di pagamento e download. Può essere installato sul sistema operativo da Windows XP a Windows 8. Il manuale di utilizzo viene dato solo dopo essersi registrati. Nel mondo però esistono solo due uffici: uno in Malesia e uno a Singapore. A seguito di una richiesta da parte dell'Università Tun Hussein Onn Malesia (UTHM), è stato sviluppato e introdotto il software "*Timetable Plus Exam 2013*" che consente di generare automaticamente un calendario d'esame per quasi 10.000 studenti con l'ottimizzazione dello spazio. E con la versione "*Timetable Plus Exam 2014*", giugno 2014, sono in grado di soddisfare tutti i problemi legati alla programmazione degli esami per tutti i tipi di istituti.

3.3 Orario accademico in altri atenei italiani

Per completezza e confronto abbiamo controllato l'orario pubblicato in alcune università italiane: l'*Università di Torino*, il *Politecnico di Milano* e l'*Università di Bologna*.

Università di Torino

L'*Università di Torino* offre ben 67 corsi di Laurea triennale, 9 corsi di Laurea Magistrale a ciclo unico, 75 corsi di Laurea Magistrale, però distribuiti su molte sedi distaccate a Torino, a Cuneo, ad Asti e a Collegno, quindi non può esserci una pianificazione centralizzata.

Anche perché dagli orari pubblicati per i vari Corsi di studi risultano utilizzati vari applicativi: dai semplici file in formato XLS e PDF fino all'interfaccia web *University Planner Web* (UP) creata da *Cineca* (*Consorzio Interuniversitario del Nord-Est per il Calcolo Automatico*). Quest'ultima ha sviluppato, quest'anno, una versione aggiornata e migliorata dell'interfaccia web, la *UP 2.0* [53], che però non viene ancora utilizzata dall'*Università di Torino*. Per tutta la Gestione didattica risulta essere utilizzato l'applicativo *Campus Net*.

Università di Bologna

Anche l'*Università di Bologna* offre molti Corsi di Laurea distribuiti in varie città: Bologna, Cesena/Cesenatico, Forlì, Ozzano dell'Emilia, Imola, Rimini, Ravenna e Faenza.

Se consideriamo solo i corsi di Ingegneria e di Architettura abbiamo tre sedi: i corsi di Ingegneria a Bologna e Ravenna, i corsi di Architettura e Ingegneria a Cesena, i corsi di Ingegneria aerospaziale e Meccanica a Forlì. Dagli orari pubblicati risulta essere utilizzato il nuovo applicativo *UP 2.0* sviluppato da *Cineca*. Per questa Università è stata sviluppata, in una tesi del 2015, un'ulteriore interfaccia web per l'inserimento di tutti i dati e

come supporto al software di ottimizzazione per la pianificazione dell'orario [11]; ma ne parleremo nel quarto capitolo.

Politecnico di Milano

Il *Politecnico di Milano* offre invece 27 corsi di Laurea e 45 di Laurea Magistrale tra Ingegneria e Architettura.

Ma anche qui abbiamo una distribuzione dei corsi tra varie sedi e in città diverse: 1 corso di Laurea e 1 di Laurea Magistrale a Como e a Mantova; solo 2 corsi di Laurea a Cremona, 2 corsi di Laurea e 4 di Laurea Magistrale a Lecco e a Piacenza; quelli più numerosi si trovano a Milano Bovisa che offre 9 corsi di Laurea e 15 di Laurea Magistrale e a Milano Leonardo che offre 14 corsi di Laurea e 24 di Laurea Magistrale.

In questo caso però l'orario delle lezioni non è visibile pubblicamente, è visibile solo accedendo ai Servizi Online con il proprio *codice persona* e *password*.

Sul sito si possono fare ricerche sull'offerta formativa, su tutta la Didattica, sulla ricerca aule, tramite un'interfaccia web creata internamente dall'*Area Servizi ICT*. E da questa pagina abbiamo trovato che a Milano Bovisa ci sono 140 aule mentre a Milano Leonardo ce ne sono 186.

Se anche una della università menzionate avesse sviluppato un programma per la creazione automatica dell'Orario non potrebbe essere applicabile al nostro caso, visto che al *Politecnico di Torino* vengono erogati 27 Corsi di Laurea Triennale e 29 Corsi di Laurea Magistrale tra Ingegneria e Architettura. Perché il problema principale è proprio la definizione di specifiche regole e vincoli per una pianificazione dell'orario che sia valida per tutte le università, difficilmente un programma scritto per creare l'orario di un'università può essere utilizzato in un'altra università. Sarebbe utile avere un *problema standard* che racchiuda un set di vincoli di un insieme di università e scrivere un programma adattabile a tutti i casi.

Capitolo 4

Descrizione del problema e sviluppo del modello

In questo capitolo, dopo un breve accenno storico, illustriamo i progressi della comunità scientifica di ricerca sul cosiddetto *timetabling problem* che si sono ottenuti in questi anni. E spieghiamo come abbiamo descritto il nostro problema nell'ambito della Ricerca Operativa e come abbiamo trovato una possibile soluzione per automatizzarlo.

4.1 Perché la Ricerca Operativa?

Tra le varie definizioni di Ricerca Operativa (indicata con l'acronimo *RO*) quella più citata risulta essere degli americani Morse e Kimball: "*applicazione del metodo scientifico da parte di gruppi interdisciplinari a sistemi complessi e organizzati per fornire al personale dirigente soluzioni utilizzabili nei processi decisionali*" [17].

Il termine *Ricerca Operativa* deriva dal termine inglese "*Operational Research*", ovvero "ricerca sulle operazioni" in ambito militare perché le sue prime applicazioni risalgono alla Seconda Guerra Mondiale per aumentare l'efficienza di operazioni militari. Anche se negli anni precedenti alla guerra, in particolare in Gran Bretagna, erano state condotte delle ricerche per un nuovo sistema di difesa: il radar.

Esistono però degli esempi di RO risalenti al 1776 da G. Monge che studiò un problema dei trasporti esaminandone gli aspetti economici e al 1885 da F. Taylor che elaborò un metodo per migliorare i processi di produzione.

Nel 1941 nasce una vera e propria sezione di RO, composta da un gruppo di scienziati esperti in materia, che prese il nome di *Operational Research Team*. Dagli anni '60 in poi la diffusione della RO ha avuto un crescente sviluppo in molti settori: industriale, marketing, pianificazione, telecomunicazioni e società di consulenza.

Alcuni problemi risolti per mezzo della RO sono per esempio [10]:

- "*finanza e investimenti*": quanto e come dobbiamo investire? quanto ci costerà un investimento?;
- "*pianificazione della produzione*": come dobbiamo pianificare il livello della produzione e l'utilizzo delle risorse? In particolare esistono problemi di "*allocazione ottima di*

risorse2: come distribuire le risorse in modo tale da minimizzare il costo complessivo oppure massimizzare il profitto, ovvero il guadagno;

- "*allocazione ottima di componenti elettronici (VLSI design)*": come disegnare una piastra madre in modo da ottimizzare le lunghezze del percorso seguito dai segnali elettrici;
- "*localizzazione e dimensionamento di impianti*": dove conviene costruire gli ospedali, o le scuole o i centri commerciali per ottimizzare il servizio che vogliamo fornire?";
- "*sequenziamento*": quale operazione o processo dobbiamo fare prima e quali dopo?;
- "*determinazione dei turni del personale*": come organizzare i turni degli infermieri in un ospedale in modo tale da minimizzare il numero totale degli infermieri da impiegare; come assegnare il personale che viaggia sui treni (i passeggeri, il capotreno, il conducente, ecc.) in modo tale da minimizzare il numero dei viaggi a "vuoto".

4.1.1 Timetabling problem

Il problema dell'orario (in inglese *timetabling problem*) consiste nel "*pianificare una sequenza di lezioni tra insegnanti e studenti in un prefissato periodo di tempo (tipicamente una settimana), soddisfacendo un set di vincoli di vario tipo*". Questa è la definizione formale data da Schaerf [27], professore all'Università di Udine e ricercatore sul timetabling problem da diversi anni. Il maggior numero di applicazioni e studi sull'argomento vengono presentati e discussi, ogni due anni, alla Conferenza Internazionale PATAT (*Practice and Theory of Automated Timetabling*). Studi ed applicazioni che generano automaticamente orari e turni in qualsiasi settore di attività: possono riguardare turni in ospedale, i trasporti, gli eventi sportivi, le istituzioni scolastiche e universitarie. La necessità di automatizzare questi tipi di problemi nasce dal fatto che questi richiedono un impegno notevole di tempo e di energie da parte dell'uomo e spesso le soluzioni raggiunte sono di bassa qualità; automatizzando il processo si possono invece raggiungere soluzioni di alta qualità e in tempi ridotti.

Esiste anche una International Timetabling Competition (ICT) dove vengono proposti soluzioni a problemi di orario [46].

Uno dei primi articoli su questo problema risale al 1963, da Gotlieb [12].

Tra gli articoli sul timetabling problem spicca "*A survey of Automated Timetabling*", del 1999 di Schaerf [27], perché descrive in modo preciso il problema e dà alcune indicazioni sulle tecniche di soluzione che riportiamo qui di seguito.

Il problema dell'orario cambia a seconda del tipo di istituto considerato, università o scuola, e dal tipo di vincoli; Schaerf classifica il problema in tre classi principali [27]:

- "**School Timetabling/Orario scolastico**: *la pianificazione settimanale per tutte le classi di una scuola, evitando che gli insegnanti incontrino due classi allo stesso tempo e viceversa*";

- **"Course Timetabling/Orario del corso:** *la pianificazione settimanale per tutte le lezioni di un insieme di corsi universitari, riducendo al minimo le sovrapposizioni di lezioni dei corsi che hanno studenti comuni*";
- **"Exam Timetabling/Calendario degli esami:** *la pianificazione per gli esami di un insieme di corsi universitari, evitando sovrapposizioni degli esami dei corsi che hanno studenti comuni*".

Il problema può essere formulato in due modi: come un "problema di ricerca" che consiste nel trovare un orario che soddisfi tutti i vincoli oppure come un "problema di ottimizzazione" che consiste nel trovare un orario che soddisfi tutti i vincoli *hard*, e che massimizzi o minimizzi una data *funzione obiettivo* che tiene conto dei vincoli *soft*.

Quindi si definisce un "*underlying problem*": il problema di decidere se esiste una soluzione nel caso di problema di ricerca, o se esiste un determinato valore della funzione obiettivo nel caso di problema di ottimizzazione.

Entrambi i problemi sono del tipo *NP-completo*, e una soluzione esatta la si può ottenere per casi di piccole dimensioni, per esempio qualche decina di corsi. Ma in realtà parliamo di centinaia di corsi e quindi solo i *metodi euristici*, Pearl (1984) [23], risultano essere applicabili a questi problemi per ottenere la soluzione ottimale.

Il "*successo dell'applicazione*" viene misurato a seconda che si tratti di un problema di ricerca o di un problema di ottimizzazione: nel primo caso la misura del successo è legata al numero delle lezioni che si riescono a pianificare, nel secondo caso è legato al valore della funzione obiettivo per la soluzione ottimale, che in alcuni casi ha un significato ben preciso (come per esempio il numero di insoddisfazioni da parte degli studenti per il risultato ottenuto), in altri invece potrebbe non essere ben chiaro (come per esempio la somma ponderata di un numero di funzioni). In entrambi i casi i risultati vengono confrontati con quelli manuali, anche se ovviamente quelli ottenuti al computer sono superiori a quelli prodotti manualmente.

Le prime tecniche di soluzione erano basate su una simulazione del modo umano per risolvere il problema [30]. Queste tecniche vengono chiamate "*euristica diretta*", e sono basate sul "*successivo aumento*": cioè si pianifica un orario, lezione per lezione, fino a quando vengono programmate tutte le lezioni, con l'idea di iniziare a pianificare per prima la lezione "*più vincolata*", quindi il problema si basa sul significato che si dà all'espressione "*più vincolata*".

Nell'articolo [27], Schaerf descrive tutti i dettagli delle tre classi di problemi di orario, ma visto che la nostra tesi si basa sul "*course timetabling*" riportiamo i dettagli solo su questa classe.

La differenza tra l'orario di una scuola e quello di un'università è legata al fatto che nelle scuole la classe è composta da studenti disgiunti mentre nelle università i corsi possono avere studenti comuni e quindi non possono essere pianificati nello stesso momento.

Un altro problema è legato alla "*disponibilità delle aule*" (e alle loro dimensioni) perché mentre nel problema scolastico non vengono quasi prese in considerazione perché si assume che ogni classe abbia la sua aula, nel caso dei corsi universitari questo non è possibile.

Nell'articolo [27] viene descritta prima la "*formulazione di base*" del problema, secondo de Werra (1985) [36], dove vengono definiti il modello matematico del problema dell'orario

(se una lezione di un corso è programmata in un determinato periodo) e i vincoli. Poi viene introdotto il "*problema di ottimizzazione*", dove viene definita la funzione obiettivo secondo de Werra e con la variante proposta da Tripathy (1992) [34] (che *minimizza l'insoddisfazione globale data dalla somma di tutte le insoddisfazioni*, per esempio nel caso in cui due lezioni, seguite da studenti comuni sono pianificate nello stesso momento). Mentre altri ricercatori (Aubin e Ferland 1989 [1]) introducono il concetto di requisiti "*hard e soft*", dove i primi devono essere inclusi nei vincoli mentre i secondi devono essere inclusi nella funzione obiettivo. Si prosegue con la descrizione delle "*varianti*" più comuni del problema dell'orario che riportiamo qui di seguito:

- "*indisponibilità e preassegnazione*": quando un docente o una classe non può essere disponibile, in un determinato momento, e una particolare lezione deve essere fissata per essere pianificata in un determinato momento;
- "*sezioni multiple e raggruppamento in sottoproblemi*": quando ci sono corsi che coinvolgono un gran numero di studenti che seguono un determinato curriculum, questi corsi possono essere ulteriormente suddivisi in sezioni, riducendo così il numero di conflitti. Il problema è chiamato "*raggruppamento di sottoproblemi*" (Laporte e Desroches 1986): è un problema di ottimizzazione che suddivide i requisiti "*hard*" (uno studente non può essere in due lezioni contemporaneamente e non può avere due lezioni consecutive in due sedi distaccate) e "*soft*" (il massimo numero di lezioni in un giorno e lo spostamento tra varie sedi fissato in un'ora libera). Il problema viene risolto in due fasi: l'algoritmo ricerca prima la "*soluzione ammissibile*" e poi ricerca l'"*ottimo locale*";
- "*periodi di lunghezza variabile*": le lezioni non durano solo un periodo ma possono durare più periodi, quindi per ogni lezione bisogna considerare l'ora di inizio e la durata; o più in generale la lezione ha una "*lunghezza variabile*" (Hertz 1992 [15]): se per esempio un corso che ha dodici periodi, può essere suddiviso in lezioni di due o tre periodi, allora il corso può avere varie lunghezze come: (3,3,3,3), (2,2,2,2,2), (3,3,2,2,2);
- "*sottoproblema di assegnazione delle aule*": quando si assegna un'aula ad una classe in un orario fissato. In particolare Carter e Tovey (1989 [5]) analizzano in dettaglio il problema e ne danno varie formulazioni. Mostrano il caso in cui il problema diventa "*NP-completo*" (ovvero quando si impone il vincolo che tutte le lezioni di un corso si devono svolgere nella stessa aula) e il caso in cui il problema è "*polinomiale*" (quando l'assegnazione in ciascun periodo è indipendente dagli altri periodi).

Infine Schaerf [27] elenca le varie tecniche di soluzione e per ciascuna di queste fa riferimento ad altri autori.

- La "*riduzione della colorazione del grafo*": viene indicato il metodo di soluzione proposto da de Werra (1985 [36]) dove ad ogni lezione di un corso si associa un vertice e per ogni corso si associa una "*clique*" di vertici.

- La "*programmazione lineare intera*": vengono elencati un certo numero di autori che hanno risolto il problema con tecniche di programmazione lineare; tra queste anche quella del "*Rilassamento Langragiana*" (Tripathy 1984 [33]).
- Le "*tecniche di flusso di rete*": viene presentata la tecnica di rete di Dinkel et al. [9] che è costituita da tre livelli, il primo livello "*Dipartimento*", il secondo livello "*Faccoltà/Staff*", il terzo livello "*Aula dimensione/tempo*". Il modello di rete può essere risolto in tempo polinomiale e la procedura può assegnare un solo insegnante a più lezioni contemporaneamente. Se la procedura risolve il problema allora il processo si arresta, altrimenti occorre un intervento umano per sbloccare la non fattibilità, per esempio modificando alcuni pesi; la procedura viene eseguita più volte fino a quando non si raggiunge una soluzione fattibile.
- La "*ricerca tabu*": viene proposta la tecnica della ricerca tabu usata da Hertz (1991-1992 [14] [15]) che considera un problema di ottimizzazione nel caso di un corso suddiviso in sezioni e nel caso di un corso con un periodo di lunghezza variabile; ma non vengono considerati i vincoli che riguardano l'allocazione delle lezioni ad un'aula.
- "*Approccio basato su regole*": la tecnica indicata è quella basata su regole (Solotrevsky et al. 1994 [31]), RAPS, come le: regole di assegnazione, regole di vincolo, regole di cambiamento locale, regole di contesto e di regole di priorità.
- "*Approccio alla programmazione con vincoli logici*": viene fatto tramite il sistema "*Constraint logic programming*" (CLP), uno strumento che modella il problema dichiarando le variabili e i loro domini e ponendo dei vincoli. Il metodo di base è una ricerca di "*backtrack*", dove i vincoli consentono al sistema di vedere in anticipo le conseguenze delle decisioni e gli eventuali errori. La tecnica di soluzione del sistema CLP è di tipo "*branch-and-bound*". Alcuni dei linguaggi CLP sono: CHIP, DOMLOG, ECLIPSE.
- Altre tecniche vengono solo accennate: gli "*algoritmi genetici*" (utilizzati anche per la pianificazione degli esami), la "*programmazione logica*" (con l'implementazione di PROLOG), la "*programmazione di obiettivi interi misti*".

Per approfondire questi argomenti si rimanda alla lettura dell'articolo di Schaerf [27] e dell'Appendice (algoritmo della colorazione del grafo, la ricerca tabu, la *simulated annealing*, gli algoritmi genetici) e degli articoli degli autori da lui riportati, e del libro [32].

Tutte queste tecniche vengono confrontate con i risultati ottenuti manualmente, ma l'assenza di una definizione comune per i vari problemi non permette di fare un confronto fra gli algoritmi proposti; quindi la "*complessità computazionale*" dei sistemi proposti viene in generale determinata dal "*tempo di calcolo*", che nella maggior parte dei casi è dell'ordine di alcuni secondi. Inoltre l'hardware utilizzato varia da caso a caso, dal mainframe al PC.

4.1.2 Altre possibili soluzioni al timetabling problem

Nell'articolo di Schaerf, riportato nel paragrafo precedente, si fa principalmente riferimento a soluzioni proposte nel campo della Ricerca Operativa; ma il problema viene affrontato anche nel campo dell'Intelligenza Artificiale, per esempio con gli "*algoritmi genetici*", la "*ricerca tabu*" e la "*programmazione dei vincoli*"; tecniche che risultano essere migliori di quelle proposte dalla RO. In particolare la "*programmazione dei vincoli*" consente una maggiore flessibilità nella formulazione del problema: perché la definizione dei vincoli e delle penalità del problema spesso è confusa, non è facile identificare in modo preciso quali siano i veri vincoli e quali siano le caratteristiche che un buon orario deve avere. Però questo metodo può risultare poco efficaci quando il problema diventa molto elevato computazionalmente.

Un'applicazione di Intelligenza Artificiale è stata nuovamente proposta da Schaerf, nel 2004, in collaborazione con Di Gaspero nell'articolo "*Il problema della generazione automatica dell'orario delle lezioni: teoria e pratica*" [29], dove viene sviluppato un risolutore basato sul "*paradigma della ricerca locale*", e impiegato per la generazione dell'orario della ex *Facoltà di Ingegneria di Udine* dall'anno accademico 2000 – 2001, con il passaggio dal vecchio al nuovo ordinamento (il "*3 + 2*").

Vengono definiti i dati del problema, i vincoli *hard* e *soft*, e la *funzione obiettivo* da minimizzare come "*somma pesata delle violazioni dei vincoli soft*". I pesi da assegnare ai vari vincoli vengono definiti dall'utente, ma hanno un "*coefficiente fisso legato al numero degli studenti coinvolti nella violazione del vincolo*". Il problema così rappresentato risulta essere una generalizzazione del problema della "*colorazione del grafo*" e quindi è un problema NP-completo.

La *ricerca locale* è un paradigma di ricerca e ottimizzazione, che offre delle tecniche *non esaustive*, perché "*non garantiscono di trovare una soluzione ammissibile o ottima, ma esplorano lo spazio di ricerca fino a quando viene soddisfatto un criterio di arresto*" [29].

Il risolutore è stato realizzato con il linguaggio di programmazione C++ utilizzando il compilatore *gcc* v.3.3 su un sistema operativo Windows e su Linux. Il tempo medio di calcolo su istanze reali è stato di circa 490 secondi per ciascuna prova con un processore Pentium 4 a 3.4 GHz. Il risolutore utilizza il *framework* EASYLOCAL++ che fornisce le strutture di controllo degli algoritmi, mentre l'utente deve solo scrivere il codice specifico nei punti prestabiliti. Per l'input e l'output dei dati sono stati utilizzati dei file di testo, mentre l'orario finale è stato pubblicato su dei file HTML per facilitare la consultazione.

L'utilizzo del risolutore prevede i seguenti passi:

- si inseriscono i dati di ingresso (i vincoli e le preferenze dei docenti) nel sistema informativo della Facoltà tramite un'interfaccia web;
- si esegue un certo numero di prove (qualche centinaia) partendo da stati iniziali casuali distinti, e tra gli stati finali si sceglie quello di costo minimo come "bozza" orario;
- questa "bozza" orario viene presa in visione e valutata dai docenti che evidenziano gli eventuali problemi e possono richiedere ulteriori modifiche; queste però non verranno

eseguite direttamente sull'orario ma verranno utilizzate per formulare nuovi vincoli e saranno aggiunti a quelli precedenti per generare una nuova istanza;

- utilizzando la "bozza" orario come stato iniziale si effettua una nuova istanza per ottenere l'orario "definitivo";
- l'orario "definitivo" viene modificato manualmente in casi eccezionali usando il risolutore solo per validare lo stato raggiunto o per trovare altre soluzioni migliorative.

Per ulteriori dettagli sugli elementi base e sulle tecniche usate si rimanda all'articolo [29]. Ma sottolineiamo che sia il codice sorgente di EASYLOCAL++ che quello dell'applicazione sono disponibili gratuitamente¹. Il risolutore è stato acquisito, modificato ed esteso da una spin-off *EasyStaff* (Zucchetti Group) che ne ha fatto un prodotto industriale ed ha venduto ad una ventina di Università.

All'undicesima edizione della Conferenza Internazionale PATAT 2016, svolta in Italia all'*Università di Udine*, sono state presentate le ricerche condotte dai due coordinatori dell'evento, Schaerf e Di Gaspero, e dal gruppo di lavoro del Dipartimento Politecnico di Ingegneria e Architettura dell'*Università di Udine* e ulteriori articoli sul problema dell'orario. Citiamo quello di Tomáš Müller, "*University Course Timetabling - Solver Evolution*" [20], dove viene presentato lo stato dell'arte e i possibili progressi futuri del software open source *UniTime*, e quello di Lach e altri, "*University course timetabling with Moses: System demonstration*" [16], dove viene illustrato un nuovo sistema IT chiamato *Moses*.

Nell'articolo "*University Course Timetabling - Solver Evolution*" [20] vengono descritti i miglioramenti apportati al software open source *UniTime* per la creazione dell'orario. Partendo dalla base di dati di benchmark già studiata nel 2011 per l'orario della *Purdue University*. I miglioramenti riguardano sia gli "*algoritmi di soluzione*" che la "*gestione dei vincoli*".

Il problema dell'orario della *Purdue University* è stato suddiviso in sotto-problemi: un problema di lezioni pianificato centralmente, otto problemi di Dipartimenti organizzati individualmente, un problema di laboratorio informatico pianificato centralmente; per un totale di 2500 classi, con 32000 studenti e 200 aule. Ogni problema ha anche una serie specifica di aule in cui le classi dovrebbero essere pianificate, e ogni classe ha un numero definito di posti.

La capacità di *UniTime* di modellare la maggior parte dei vincoli del mondo reale abbinata alla capacità di inserire diverse tecniche di soluzione nel risolutore, fa sì che il software risulti essere un valido strumento per risolvere problemi di dimensioni e complessità reali. La speranza degli autori è che *UniTime* possa essere utilizzato da altri ricercatori per ampliare il set di dati e possa essere utilizzato da più istituzioni sparse in tutto il mondo.

Nell'articolo il problema dell'orario viene definito come un "*problema di allocazione delle risorse allo scopo di assegnare classi a tempi e spazi in modo tale che due classi*

¹visitare la pagina <http://satt.diegm.uniud.it/projects/>

non vengano collocate nella stessa aula o insegnate dallo stesso insegnante in un dato momento" [20] e devono essere raggiunti quattro obiettivi che riportiamo qui di seguito:

1. "*preferenze di data*", ogni settimana può esserci una lezione a *tempo pieno*, oppure a settimane alterne *pari/dispari*, oppure solo per le prime N settimane; e le "*preferenze di tempo*", una lezione può durare 50 minuti per due volte a settimana. A queste preferenze vengono assegnate delle "*penalità*" che sono *negative* se la preferenza è fortemente preferita o preferita, oppure sono *positive* se la preferenza è scoraggiata o vietata, invece è *neutra* quando non viene indicata nessuna preferenza. "*L'obiettivo di preferenza temporale è quello di minimizzare la penalità temporale complessiva*";
2. "*preferenze di aula*": ogni corso può chiedere di utilizzare un determinato numero di aule insieme alle relative penalità, inserite dall'utente, in base alla dimensione del corso e alle preferenze del corso. Un'aula può essere disponibile solo in determinati orari oppure può essere assegnata ad un determinato dipartimento per un determinato tempo. Ovviamente un'aula non può essere occupata da due corsi nello stesso momento;
3. "*preferenze dei docenti*": un corso può avere uno o più docenti. Ma non è consentito che un docente possa insegnare due o più lezioni contemporaneamente. Se un docente insegna due lezioni consecutive e collocate in aule troppo distanti tra loro allora l'assegnazione è vietata. Viene assegnata una diversa *penalità* a seconda della distanza tra le due aule: per esempio è scoraggiato (penalità +1) che un docente insegni due lezioni consecutive che si trovano a una distanza di 50 metri, mentre è fortemente scoraggiato (penalità +4) se si trovano ad una distanza superiore ai 50 metri ed è vietato se si trovano ad una distanza oltre i 200 metri;
4. "*preferenze degli studenti*": ogni studente ha un elenco di corsi che deve seguire, sulla base dei curriculum. Il problema è complicato dal fatto che i corsi tendono ad avere più sezioni. Il conflitto nasce quando uno studente è iscritto a due sezioni che si sovrappongono nel tempo oppure che sono collocate in momenti consecutivi e in aule troppo distanti tra di loro. Ad ogni tipo di conflitto può essere assegnato una *penalità*.

A questi si aggiungono gli "*ulteriori vincoli di distribuzione*": "*back-to-back*" (le lezioni devono essere inserite nello stesso giorno una dietro l'altra), "*precedenza*" (le lezioni devono essere inserite nell'ordine dato), "*tempi diversi*" (le lezioni non devono sovrapporsi), "*distribuzione nel tempo*" (le lezioni devono essere distribuite nel tempo), "*condivisione dell'aula*" (le lezioni possono condividere un'aula se questa è sufficientemente capiente). Spesso questi vincoli di distribuzione sono assegnati dalla struttura del corso. Vengono assegnati le penalità ai vincoli a seconda che la preferenza sia fortemente preferita, preferita, scoraggiata, fortemente scoraggiata e vietata.

Il risolutore si basa su un algoritmo iterativo di "*ricerca in avanti*" (Rudová et al 2011 [25]), che è simile ai metodi di ricerca locale, ma opera su soluzioni fattibili che non sono necessariamente complete; quindi ci possono essere delle lezioni non assegnate, ma tutti i vincoli sulle classi devono essere rispettati. La ricerca viene elaborata in modo

iterativo partendo da un orario iniziale e termina quando viene trovato l'orario desiderato o quando c'è un time out.

Il solver framework, chiamato CPSolver (Constraint Solver Library), è scritto in Java ed è disponibile sotto licenza GNU Lesser General Public License. Sono stati utilizzati in questi anni varie versioni: il CPSolver 1.1 (nel 2011) fino al CPSolver 1.3 del dicembre 2015. Agli inizi con solo una CPU fino ad arrivare a 4 core CPU. Il tempo medio per le esecuzioni è di circa 30 minuti, utilizzando un Mac Pro con due processori Intel Xeon a 6 core con 3,06 GHz e 64 GB di memoria, OS X 10.11 e Java 8.

Il risolutore, CPSolver 1.3, è arrivato tra i finalisti della International Timetabling Competition - ITC2007 [46] - dove bisognava risolvere tre tracce di problemi di orario: 1) *Examination Timetabling* - calendario esami, 2) *Post Enrollment based Course Timetabling* - Orario corsi post iscrizione, 3) *Curriculum Based Course Timetabling* - Orario corsi basato sul curriculum; arrivando primo nella prima e terza traccia e quinto nella seconda traccia.

Nel secondo articolo, "*University course timetabling with Moses: System demonstration* [16], viene brevemente presentato il progetto realizzato nel 2002 da *innoCampus*, un Dipartimento dell'*Università Tecnica di Berlino* che ha svolto ricerche sul problema della generazione dell'orario e della programmazione degli esami implementando un sistema IT chiamato *Moses*. *Moses* è stato utilizzato per la prima volta nella primavera del 2003 per distribuire dei tutorial ai 32000 studenti. Poi è stato esteso per risolvere il problema della pianificazione degli esami, in particolare dal 2015 presso l'*Università tecnica di Monaco* con 39000 studenti.

Nel 2012 *innoCampus* e il dipartimento di Ricerca Operativa della *RWTH Aachen University* hanno avviato un progetto che ha portato all'estensione di *Moses* per risolvere il problema dell'orario per la *RWTH Aachen University*, con oltre 42000 studenti, 1200 docenti e 500 aule. Il modello di ottimizzazione di questo sistema è un modello di programmazione lineare intera mista (*mixed-integer programming* (MIP)). Il modello del problema dell'orario viene poi risolto utilizzando un risolutore MIP commerciale; il sistema si presenta come un'applicazione web Java EE.

Il problema dell'orario è stato anche argomento di una tesi di Laurea, nel 2011, di uno studente del *Politecnico di Torino*, "*Ottimizzazione dell'orario universitario*" [2] e di una tesi di Laurea Magistrale in Informatica di uno studente dell'*Università di Bologna*, "*Un software per la gestione dell'orario delle lezioni*" [11]. In entrambi i casi viene analizzato e sviluppato un software per la generazione automatica degli orari universitari.

Nella prima tesi viene descritto un modello matematico per la generazione automatica dell'orario delle lezioni dell'ex *IV* Facoltà del Politecnico di Torino, suddividendo il problema in due sotto-problemi: la collocazione delle attività nel tempo e l'assegnazione delle attività in aule e laboratori adeguati. Però in questo modello vengono definiti dei vincoli che garantiscono sempre la presenza di un numero adeguato di aule con le caratteristiche richieste, così il secondo problema risulta avere sempre una soluzione ammissibile.

Per l'assegnazione delle attività alle aule viene data solo una formulazione tramite due

funzioni obiettivo: "minimizzare la strada percorsa da tutti gli studenti in uno qualsiasi dei giorni dell'orario" oppure "minimizzare la massima strada percorsa da uno qualsiasi degli studenti in qualsiasi giorno dell'orario" [2]; considerando il cosiddetto problema del *Vehicle routing*, che è una ulteriore generalizzazione del problema del *Commesso viaggiatore* (*Traveling Salesman Problem*, *TSP*, [26], [32], [35]).

Come soluzione al problema dell'assegnazione delle attività alle aule propone di assegnare prima i corsi con numerosità grande alle aule di grande dimensione, poi di assegnare i corsi di media numerosità alle aule di dimensione media e una volta esaurite, assegnare i corsi rimasti alle aule grandi (che ovviamente sono adeguate) ed infine assegnare i corsi con numerosità ridotta alle aule di piccole dimensioni, esaurite queste assegnare i corsi piccoli rimasti ad una qualsiasi delle aule ancora disponibili tra quelle grandi. Questo problema è risolvibile in "tempo polinomiale" [2]. Per l'implementazione del modello matematico per la creazione dell'orario è stato utilizzato il software *Xpress-IVE* con il linguaggio di programmazione *Mosel*, lo stesso che abbiamo usato noi per il nostro modello e che descriveremo nel quinto capitolo.

Nella seconda tesi viene sviluppata un'interfaccia grafica come supporto al software di ottimizzazione per la pianificazione dell'orario dei corsi di Informatica dell'*Università di Bologna* per l'anno accademico 2014 – 2015 [11]. Il software per l'orario risulta essere un adattamento del modello matematico del *timetabling problem* proposto in un articolo di Daskalaki e altri [7]. Il modello matematico è stato implementato utilizzando: il risolutore open source *GLPSOL*, per la programmazione lineare intera, che fa parte del pacchetto *GLPK* (*GNU Linear Programming Kit*); una libreria software scritta in *ANSI C* che implementa i metodi di soluzione per i problemi di programmazione lineare. L'interfaccia grafica è stata realizzata per facilitare l'inserimento di tutti i dati di input utili per il risolutore, che devono essere inseriti dalla segreteria (come i dati degli studenti, dei corsi, delle aule, dei docenti) e dai docenti (come i dati relativi al proprio corso e alle preferenze di orario).

4.2 Dati del problema e i vincoli

I corsi della Laurea Triennale e Magistrale di Ingegneria e di Architettura, richiedono l'utilizzo dei laboratori di informatica dell'Area IT per poter svolgere le esercitazioni di laboratorio durante la settimana e per due periodi didattici, ciascuno composto da quattordici settimane. I laboratori dell'Area IT sono distribuiti in varie sedi della città di Torino.

La sede del Lingotto (dotata di un solo laboratorio) viene utilizzata gestita separatamente al momento dai corsi del Collegio di Architettura e dai corsi Master, quindi l'assegnazione di questo laboratorio non influisce quella delle altre sedi.

Anche la sede di Mirafiori utilizzata dal Collegio di Design e di Ingegneria dell'Autoveicolo, e la sede del Castello del Valentino utilizzata dai corsi del Collegio di Architettura sono entrambe dotate di tre laboratori e in entrambi i casi l'assegnazione dei corsi ai laboratori viene gestita separatamente.

La sede Centrale/Cittadella è dotata di cinque laboratori e di due aule attrezzate con computer portatili (in corso Duca) e di tre laboratori (in via Boggio) che vengono utilizzate dai corsi di tutti i Collegi.

La sede delle ex Tornerie, dotata di un laboratorio e un'aula attrezzata con computer fissi, viene utilizzata principalmente dai corsi del Collegio ICM, gli eventuali spazi liberi rimasti possono essere assegnati ai corsi degli altri Collegi.

Esamineremo solo il primo periodo didattico, perché più critico, e considereremo solo i laboratori della sede Centrale di corso Duca e di via Boggio perché sono quelle dove si verificano maggiormente le sovrapposizioni delle richieste dei corsi. I corsi che utilizzano il laboratorio per poche date (minore di cinque settimane) vengono gestiti successivamente in base alla disponibilità rimasta e quindi non rientrano nel nostro problema.

Ora descriviamo tutti i dati del nostro problema: i **parametri** (le risorse, il modulo orario e i corsi) e i **vincoli**.

4.2.1 Le risorse

Presso la sede Centrale/Cittadella sono disponibili cinque laboratori e due aule attrezzate con computer portatili (in corso Duca), e tre laboratori (in via Boggio). Ma per entrambe le sedi un laboratorio è esclusivamente utilizzato per l'accesso libero agli studenti, di conseguenza non è possibile assegnarlo a nessun corso; e le due aule attrezzate con i computer (di recente costruzione) sono utilizzate da determinati corsi che richiedono il doppio utilizzo come aula da disegno e come aula con PC, e queste non vengono assegnate dal Referente dei laboratori bensì da quello delle aule in collaborazione con il Referente di questi corsi.

Per il nostro modello consideriamo solo i quattro laboratori di corso Duca, descritti nella tabella 4.1.

Tabella 4.1. Caratteristiche LAIB - Duca

Nome Laib	Numero postazioni	Capienza massima
LAIB1	76	156 studenti
LAIB3	62	117 studenti
LAIB4	59	115 studenti
LAIB5	39	80 studenti

E i due laboratori di via Boggio, descritti nella tabella 4.2.

Tabella 4.2. Caratteristiche LAIB - Boggio

Nome Laib	Numero postazioni	Capienza massima
LAIB2B	42	84 studenti
LAIB3B	36	70 studenti

Bisogna tener conto che un laboratorio, il LAIB1, può essere utilizzato da due corsi nello stesso momento perché è diviso in due sale, come indicato nella tabella 4.3.

Tabella 4.3. LAIB1 - Duca

Nome Laib	Numero postazioni	Capienza massima
LAIB1a	42	84 studenti
LAIB1b	34	70 studenti

Ogni laboratorio è dotato di un *numero di postazioni*, ovvero numero di computer, e di una *capienza massima*, ovvero il massimo numero di persone che possono stare all'interno. Tutti i laboratori/aule sono dotati degli stessi software. Ogni singolo software può avere un numero illimitato o limitato di licenze, ma questo non impedisce l'assegnazione del laboratorio al corso.

4.2.2 L'Orario

L'attività didattica, presso il *Politecnico di Torino*, si svolge nei giorni dal lunedì al giovedì dalle ore 8,30 alle ore 19,00 e il sabato dalle ore 8,30 alle ore 13,00.

Ad ogni Insegnamento (che qui chiameremo per semplicità corso) viene assegnato un monte ore, che viene determinato dal prodotto dei crediti formativi universitari (CFU) per 25 ore. Questo monte ore viene suddiviso in moduli orari distribuiti nella settimana e poi replicato per le 14 settimane del periodo didattico di riferimento.

Il *Politecnico di Torino*, ha stabilito come modulo orario: un'ora e trenta minuti. Quindi la settimana è composta da 34 moduli orari: dal lunedì al giovedì dalle 8.30 alle 19, e il venerdì dalle 8.30 alle 17.30; l'ultimo modulo orario del venerdì dalle 17.30 alle 19.00 e il sabato mattina devono essere utilizzati solo per eventuali recuperi di lezioni nel caso non ci fossero altre possibilità durante la settimana.

Ma nei laboratori dell'Area IT bisogna tener presente che:

- nessuna attività didattica può essere assegnata nel primo modulo del lunedì (dalle ore 8,30 alle 10,00) perché i laboratori sono chiusi per manutenzione,
- l'attività didattica al sabato non viene svolta perché i laboratori sono chiusi.

Quindi l'attività didattica, si svolge il lunedì dalle ore 10,00 alle ore 19,00, e dal martedì al venerdì dalle ore 8,30 alle ore 19,00 per un totale di 34 moduli.

Per il nostro modello quindi considereremo i moduli orari, in sequenza, dalle ore 10,00 del lunedì fino alle ore 19,00 del venerdì, $\{1, \dots, 34\}$.

4.2.3 I corsi

Come descritto nel secondo capitolo, i Referenti Orari di tutti i Collegi dopo aver generato una bozza orario con le relative aule assegnate, si recano dal Referente dei laboratori di Informatica dell'Area IT, secondo un calendario di incontri in base alla numerosità delle richieste e alla numerosità degli studenti dei corsi per ogni Collegio.

Le principali informazioni che i Referenti Orari dei Collegi devono comunicare al Referente dei laboratori, oltre ai dati relativi al corso (come nome e titolare), sono:

- moduli orari richiesti;
- numerosità stimata degli studenti;
- software di utilizzo.

Per quanto riguarda la numerosità degli studenti dobbiamo considerare che c'è una differenza tra i laboratori e le aule: nelle aule possono accedervi anche tutti gli studenti iscritti mentre nei laboratori possono accedervi solo gli studenti di prima frequenza nell'anno accademico corrente.

Nel primo periodo didattico bisogna tenere conto dei 20 corsi di *Informatica I* del primo anno in comune che devono svolgere le esercitazioni presso i laboratori dell'Area IT. Ogni corso è suddiviso in tre squadre, e ognuna deve svolgere un modulo orario a settimana, per un totale di tre moduli consecutivi per corso. L'orario è già stato definito e consolidato dalla Commissione preposta per il primo anno in comune, e da alcuni anni sono riservati 60 moduli orari a settimana e questo comporta l'utilizzo in parallelo di 2 laboratori dell'Area IT in determinate fasce orarie dal lunedì al venerdì.

Riportiamo nella tabella 4.4, a pag.49, il numero dei Corsi di Laurea Triennale e Magistrale dei Collegi che richiedono l'utilizzo dei laboratori, solo in sede Centrale/Cittadella, e il numero dei moduli orari richiesti.

Tabella 4.4. Corsi di Ingegneria e di Architettura

Collegio dei Corsi di Laurea	Numero Corsi	Numero Slot	Numero Corsi	Numero Slot
	L. Triennale		L. Magistrale	
1°anno-INFORMATICA I	20	60	-	-
AEROSPAZIALE	2	2	8	16
AMBIENTE E TERRITORIO	1	2	6	11
ARCHITETTURA	4	7	-	-
BIOMEDICA	-	-	4	5
CHIMICA	1	1	2	4
CIVILE	1	2	10	17
EDILE	-	-	3	6
ELETTRICA	-	-	2	3
ELN-TLC-INF-CINEMA	5	11	4	7
ENERGETICA	-	-	4	16
MATEMATICA	1	1	2	2
MECCANICA	5	10	7	14
PRODUZIONE	2	3	1	2

Per *slot* indichiamo il modulo orario di un'ora e trenta minuti.

Quindi abbiamo 42 corsi della Laurea triennale che richiedono 99 moduli orari e 53 corsi della Laurea Magistrale che ne richiedono 103. Se consideriamo tutti i corsi della Triennale e Magistrale abbiamo 202 moduli orari e visto che ogni laboratorio dispone di 34 moduli orari a settimana, vuol dire che occorrono almeno 6 laboratori per soddisfare tutte queste richieste.

4.2.4 I vincoli

Come indicato nel secondo capitolo, durante l'assegnazione manuale dei laboratori, le richieste dei Referenti orari vengono raccolte in un foglio, formato A3, che rappresenta la "settimana tipo" come una griglia: dove le righe indicano i moduli orari da un'ora e trenta minuti, le colonne i giorni della settimana, da lunedì a venerdì (al sabato i laboratori sono chiusi).

Quindi in ogni "cella", corrispondente ad un modulo orario in un determinato giorno, vengono inseriti tutti i corsi che richiedono l'utilizzo del laboratorio. Una volta riempite tutte le "celle" si individuano tutte le sovrapposizioni e si fa una prima valutazione di quali corsi dovrebbero cambiare orario; perché essendo decentralizzata la generazione dell'orario la probabilità che si verifichino le sovrapposizioni dei corsi nella stessa fascia oraria è alta e quindi i laboratori a disposizione non sono sufficienti per ricoprire tutte le richieste.

Questa prima valutazione viene fatta dal Referente dei laboratori sulla base delle linee guida indicate durante la Riunione con la Commissione Orari e il vice Rettore della Didattica, come:

- i corsi del primo anno in comune hanno la precedenza rispetto ai corsi degli anni successivi;
- le tre squadre dei 20 corsi di Informatica sono collocate in tre moduli orari consecutivi ed è richiesto che siano il più possibile nello stesso laboratorio della stessa sede, o almeno nei laboratori vicini ma della stessa sede;
- bisogna evitare la sovra allocazione aula e laboratorio nel caso quest'ultimo non venga utilizzato per un numero di settimane almeno maggiore a sette;

e sulla base delle caratteristiche dei laboratori dell'Area IT:

- tutti i laboratori nel primo modulo del lunedì (8,30 – 10,00) sono chiusi per manutenzione;
- tutti i laboratori al sabato sono chiusi;
- in ogni laboratorio/aula attrezzata con PC ci possono stare al massimo due studenti per PC, ma in ogni caso non bisogna superare la capienza massima;
- in ogni laboratorio/aula attrezzata è presente un solo corso, tranne al LAIB1 della sede centrale di corso Duca che può avere due corsi nello stesso momento;

- tutti i laboratori sono dotati dello stesso software. Ogni singolo software può però avere un numero illimitato o limitato di licenze, ma questo non impedisce l'assegnazione del laboratorio al corso

Durante la generazione dell'orario occorre sapere quale sia il software richiesto; in particolare per quale piattaforma:

1. la piattaforma Xen App: l'utilizzo contemporaneo di alcuni software (al momento solo Solidworks e Geartrax), in tutti i PC di tutte le sedi, non deve superare un certo numero di licenze (al momento 200 licenze);
2. la piattaforma VDI (VMWARE): la somma delle macchine virtuali utilizzate in tutti i laboratori di tutte le sedi deve essere minore di un certo numero (al momento 300).

Il primo caso però è presto risolvibile perché i corsi che utilizzano questi software vengono svolti nelle due aule attrezzate presso la sede centrale di corso Duca, quindi spetta al Referente orario fare attenzione durante la pianificazione non aggiungendo altri corsi in parallelo a quelli assegnati nelle aule.

Anche il secondo caso, al momento, non impedisce la pianificazione perché i corsi che utilizzano il software con la piattaforma VDI sono molto pochi e la numerosità degli studenti è bassa, quindi non è possibile superare la somma delle macchine virtuali utilizzabili.

Un possibile problema che invece può presentarsi è avere la disponibilità in un solo laboratorio di un determinato software, che per problemi di tempo è stato installato solo lì, oppure di particolari attrezzature; allora se un corso richiede quelle caratteristiche specifiche deve necessariamente essere assegnato in quel laboratorio.

In questa tesi i vincoli sul software non verranno considerati, ma daremo solo una formulazione per il primo caso se si dovesse presentare questo problema in futuro; e per completezza daremo una formulazione sull'ultimo vincolo.

4.3 Costruzione del modello matematico

4.3.1 Un po' di teoria

Qui di seguito riportiamo alcuni aspetti teorici di base di *Ricerca Operativa* utili per capire il nostro problema e i passi seguiti per la costruzione del modello matematico, che fanno ampiamente riferimento ai testi [10], [6], [8].

Con un *modello* possiamo mettere in evidenza le caratteristiche di un oggetto reale; con un *modello matematico*, costituito da un insieme di relazioni matematiche, descriviamo in modo semplificato i problemi reali.

Un *modello di programmazione matematica* è formato dai seguenti elementi:

- **parametri**: sono i dati del problema, i dati di input;
- **variabili decisionali** che sono le incognite sulle quali dobbiamo agire e sono associate alle decisioni da prendere;

- **vincoli:** sono delle disequazioni o equazioni matematiche che stabiliscono i valori assunti dalle variabili e definiscono geometricamente una regione chiamata *regione di ammissibilità*;
- **funzione obiettivo:** è la funzione delle variabili decisionali ed è la quantità che dobbiamo massimizzare o minimizzare.

Un *problema di ottimizzazione* è un particolare modello matematico ed è formulato con un *modello di programmazione matematica* come:

$$\min f(x_1, \dots, x_n) \quad (\text{oppure} \quad \max f(x_1, \dots, x_n))$$

$$g_i(x_1, \dots, x_n) = \begin{cases} \geq b_i \\ = b_i \\ \leq b_i \end{cases} \quad \forall i = 1, \dots, m$$

$$x_1, \dots, x_n \in S \subseteq \mathbb{R}$$

La "risoluzione di un problema di ottimizzazione, consiste nel determinare una tra le scelte migliori, da un insieme ammissibile S , che soddisfa tutti i vincoli g_i e massimizza o minimizza il valore della funzione obiettivo f ".

Le variabili del problema:

- possono assumere tutti i valori reali e sono variabili "continue" $x_i \in \mathbb{R} \quad \forall i = 1, \dots, n$
- possono essere numeri interi e sono:
 - variabili "binarie" $x_i \in \{0,1\} \quad \forall i = 1, \dots, n$
 - variabili "discrete" $x_i \in \mathbb{Z} \quad \forall i = 1, \dots, n$

Ora richiamiamo la definizione di *funzione lineare* [10]: "una funzione reale di n variabili reali $f: \mathbb{R}^n \rightarrow \mathbb{R}$ si dice lineare se può essere scritta nella forma:

$$c_1x_1 + c_2x_2 + \dots + c_nx_n$$

con c_1, \dots, c_n costanti reali".

Quindi un problema di *Programmazione Lineare (PL)* è caratterizzato da: una *funzione obiettivo lineare* f da minimizzare (*min*) o da massimizzare (*max*):

$$f(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n = \sum_{j=1}^n c_jx_j$$

e da un numero finito di *vincoli lineari* scritti nella forma:

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &\leq (=, \geq) b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n &\leq (=, \geq) b_2 \\ &\cdot \\ &\cdot \\ &\cdot \\ a_{m1}x_1 + \dots + a_{mn}x_n &\leq (=, \geq) b_m \end{aligned}$$

ovvero:

$$\sum_{j=1}^n a_{ij}x_j = \begin{cases} \geq b_i \\ = b_i \\ \leq b_i \end{cases} \quad \forall i = 1, \dots, m$$

dove:

- $x_j \quad \forall j = 1, \dots, n$ sono le variabili decisionali (le incognite) che possono essere *reali*, *interi* o *binarie* ($x_j \in \{0,1\}$);
- $c_j \quad \forall j = 1, \dots, n$ sono i coefficienti di costo (*min*) o di profitto (*max*) e sono costanti note del problema (parametri);
- $a_{ij} \quad \forall i = 1, \dots, m$ e $\forall j = 1, \dots, n$ sono i coefficienti tecnologici (parametri);
- $b_i \quad \forall i = 1, \dots, m$ sono i termini noti (parametri).

I modelli di programmazione matematica si possono classificare in modelli di:

- *Programmazione Lineare (PL)* se la funzione obiettivo e i vincoli sono funzioni *lineari* e se tutte le variabili possono assumere valori reali;
- *Programmazione Lineare Intera (PLI)* se almeno una delle variabili è intera (o binaria);
- *Programmazione Lineare Intera Mista (PLIM)* se alcune variabili assumono valori reali e altre valori interi;
- *Programmazione Non Lineare (PNL)* se almeno una delle funzione non è lineare.

I passi da seguire per trasformare un problema reale in un modello di programmazione lineare sono:

1. individuare quali sono le decisioni da prendere, quindi definire le "*variabili decisionali (le incognite)*";
2. determinare qual è l'obiettivo, quindi dobbiamo formulare la "*funzione obiettivo*" da ottimizzare;

3. descrivere le "condizioni di ammissibilità delle soluzioni", quindi costruire un sistema di disequazioni e/o equazioni lineari che rappresentano i "vincoli" che le variabili devono soddisfare per essere le soluzioni ammissibili del problema.

Visto che noi trattiamo solo modelli *lineari* dobbiamo fare attenzione a non moltiplicare fra loro le variabili, possono essere solamente sommate fra di loro o moltiplicate per una costante.

Dopo aver descritto i dati e i vincoli del nostro problema, costruiamo ora il nostro vero e proprio modello matematico.

È un modello di *Programmazione Lineare Intera (PLI)* con variabili intere (binarie) e bisogna modellare il fatto che un evento possa verificarsi oppure no; in particolare è un modello di *Programmazione lineare intera booleana (PLI 0/1)* dove definiamo una variabile x che può assumere valore 1 se l'evento si è verificato, 0 altrimenti.

Il nostro problema è un problema di **assegnamento** quindi: "Ogni corso deve essere assegnato al laib e al laib ci può stare al più un corso" e la *funzione obiettivo* che dobbiamo massimizzare è data dalla *somma dei corsi assegnati ai laib*.

Per costruire il modello dobbiamo: individuare le *variabili decisionali (le incognite)*, scrivere i *vincoli*, tramite equazioni e disequazioni lineari, e definire la *funzione obiettivo* da ottimizzare.

4.3.2 Variabile decisionale

Per il nostro modello dobbiamo verificare se il corso è assegnato al laboratorio oppure no in un determinato modulo orario (stabilito dai Referenti orari).

Quindi la variabile decisionale è:

$$x_{ijk} = \begin{cases} 1 & \text{se il corso } i \text{ è assegnato al LAIB } j \text{ nel modulo orario } k \\ 0 & \text{altrimenti} \end{cases} \quad (4.1)$$

Dove i pedici:

- i appartiene all'insieme I dei corsi che hanno richiesto l'utilizzo dei laboratori;
- j appartiene all'insieme J dei laboratori informatici dell'Area IT che prenderemo in considerazione;
- k appartiene all'insieme $K = \{1...34\}$ dei moduli orari di cui è composta la settimana nei laboratori.

Quindi:

$$x_{ijk} \in \{0,1\} \quad \forall i \in I \quad \forall j \in J \quad \forall k \in K$$

Definiamo come *termini noti*:

- N_i : la numerosità degli studenti del corso i ;

- C_j : la capienza massima di un laboratorio j , che corrisponde al numero massimo di persone all'interno del laboratorio;
- p_i : il peso che viene assegnato ad ogni corso e serve, in questo caso, per dare una priorità ai corsi a seconda delle esigenze fornite dalla Commissione orario.

Se dovessimo considerare anche il numero massimo di licenze relativo a particolari software (per esempio Solidworks e Geartrax) potremmo indicarlo come: SG_k numero massimo di licenze disponibili nel modulo orario k .

In alcuni casi è opportuno definire una *variabile logica*, il cui dominio è limitato ai soli valori binari 0 e 1 che deve essere messa in relazione con il valore delle altre variabili decisionali.

Quindi definiamo la variabile "ausiliaria" binaria (o sì o no):

$$y_i = \begin{cases} 1 & \text{se il corso } i \text{ è assegnato} \\ 0 & \text{altrimenti} \end{cases} \quad (4.2)$$

E la relazione tra y_i e x_{ijk} è data dal vincolo di "attivazione":

$$\forall_i \quad \forall_j \quad \forall_k \quad x_{ijk} \leq y_i \quad (4.3)$$

4.3.3 Vincoli

Ora dobbiamo descrivere le *condizioni di ammissibilità delle soluzioni*, quindi costruiamo un sistema di disequazioni e/o equazioni lineari.

Assegnamento

Ogni corso i deve essere assegnato ad un laboratorio j in un determinato modulo orario k , quindi scriviamo:

$$\forall_i \quad \sum_{jk} x_{ijk} = 1 \quad (4.4)$$

il vincolo così descritto impone che per ogni corso i la somma dei corsi assegnati ad un laboratorio j nel modulo orario k deve essere uguale a 1, quindi se il risultato della somma fosse 0 allora vuol dire che non è stato assegnato nessun corso ad un laboratorio; ovvero il problema non ha soluzione.

Ma noi vogliamo che un corso possa essere assegnato al laboratorio nel modulo orario fissato, allora il vincolo diventa:

$$\forall_i \quad \sum_{jk} x_{ijk} \leq 1 \quad (4.5)$$

E in ogni laboratorio j , nel modulo orario k , ci può essere al più un corso, quindi aggiungiamo la condizione:

$$\forall_j \quad \forall_k \quad \sum_i x_{ijk} \leq 1 \quad (4.6)$$

Caso particolare del LAIB1

Abbiamo appena scritto che in ogni laboratorio ci può stare al più un corso (condizione valida anche nel caso di assegnamento di aule); ma nella sede centrale di corso Duca c'è un laboratorio costruito in modo tale da poter allocare contemporaneamente due lezioni, perché il laboratorio è suddiviso in due sale, ognuna dotata di un numero di computer e capienza, come indicato nel paragrafo precedente.

Di conseguenza se indichiamo la prima sala come LAIB1a ($j = 1$), la seconda sala come LAIB1b ($j = 2$) e il laboratorio intero come LAIB1 ($j = 3$), dobbiamo scrivere la condizione che, in ogni modulo orario k , ci possono stare al massimo due corsi, un corso nella prima sala e uno nella seconda sala ($j = 1$ e $j = 2$):

$$\forall_k \quad \sum_i x_{i1k} + \sum_i x_{i2k} \leq 2 \quad (4.7)$$

Ma dobbiamo anche aggiungere la condizione che non possiamo mettere un corso al LAIB1a ($j = 1$) e un corso al LAIB1 ($j = 3$), quindi:

$$\forall_k \quad \sum_i x_{i1k} + \sum_i x_{i3k} \leq 1 \quad (4.8)$$

Ovviamente anche la condizione che non possiamo mettere un corso al LAIB1b ($j = 2$) e un corso al LAIB1 ($j = 3$):

$$\forall_k \quad \sum_i x_{i2k} + \sum_i x_{i3k} \leq 1 \quad (4.9)$$

Queste tre disequazioni, (4.7) (4.8) (4.9), possiamo scriverle in un'unica disequazione in questo modo:

$$\forall_k \quad \sum_i x_{i1k} + \sum_i x_{i2k} \leq 2 \left(1 - \sum_i x_{i3k} \right) \quad (4.10)$$

Capienza del LAIB

Per ogni laboratorio j e per ogni modulo orario richiesto k , il corso i non deve superare la capienza massima del laboratorio, ovvero il numero degli studenti del corso (N_i) non deve essere superiore alla capienza massima (C_j) del laboratorio:

$$\forall_j \quad \forall_k \quad \sum_i N_i x_{ijk} \leq C_j \quad (4.11)$$

Corsi di Informatica del primo anno e corsi da tre moduli orari consecutivi

Abbiamo detto che nel primo periodo didattico ci sono i 20 corsi di *Informatica I* che devono svolgere le esercitazioni presso i laboratori dell'Area IT. L'orario dei corsi del primo anno in comune viene definito da una Commissione preposta e da alcuni anni viene consolidato lo stesso orario; e visto l'elevato numero degli studenti iscritti al primo anno, la Commissione ha stabilito di dar loro la priorità rispetto ai corsi degli anni successivi.

Ogni corso è suddiviso in tre squadre e ognuna deve svolgere un modulo orario a settimana, per un totale di tre moduli consecutivi per corso.

Quindi dobbiamo indicare che le tre squadre dei corsi di Informatica (o i corsi che hanno comunque tre moduli orari consecutivi) devono effettivamente essere assegnati nei tre moduli orari consecutivi richiesti:

$$\forall i \in I_3 \quad \sum_j (x_{ijk} + x_{i+1jk+1} + x_{i+2jk+2}) = y_i + y_{i+1} + y_{i+2} \quad (4.12)$$

Dove con I_3 indichiamo solo i corsi da tre moduli orari consecutivi.

A questa condizione bisogna aggiungere la richiesta della Commissione che i corsi devono essere nello stesso laboratorio j per evitare spostamenti da parte del docente.

$$\forall i \in I_3 \quad \forall_j \quad x_{ijk} = x_{i+1jk+1} \quad \text{e} \quad x_{i+1jk+1} = x_{i+2jk+2} \quad (4.13)$$

La priorità dei corsi di Informatica viene data assegnando il valore più alto al *peso* p_i , che troveremo nella *funzione obiettivo*.

Corsi da due moduli orari consecutivi

Anche per i corsi che hanno due moduli orari consecutivi, dobbiamo imporre la condizione che devono effettivamente essere assegnati nei due moduli orari consecutivi richiesti, quindi:

$$\forall i \in I_2 \quad \sum_j (x_{ijk} + x_{i+1jk+1}) = y_i + y_{i+1} \quad (4.14)$$

Dove con I_2 indichiamo solo i corsi da due moduli orari consecutivi.

A questa condizione bisogna aggiungere la richiesta della Commissione di assegnare lo stesso laboratorio anche per i corsi che hanno due moduli orari consecutivi perché si vuole evitare che il docente perda tempo nel trasferimento da un laboratorio all'altro nel caso in cui il corso sia suddiviso in due squadre con un modulo orario per squadra.

$$\forall i \in I_2 \quad \forall_j \quad x_{ijk} = x_{i+1jk+1} \quad (4.15)$$

4.3.4 Vincoli aggiuntivi

Numero di licenze limitato per alcuni software

Se dovessimo considerare anche il numero limitato di licenze per esempio di software che utilizzano la piattaforma Xen App, che al momento è fissato a 200 allora dobbiamo scrivere

che per ogni modulo orario richiesto k la numerosità degli studenti N_j diviso due (perché per ogni postazione PC ci possono stare due studenti), del corso i non deve superare il numero massimo di licenze SG_k :

$$\forall_k \sum_{ij} \frac{N_j}{2} x_{ijk} \leq SG_k \quad (4.16)$$

oppure se indico con: P_i = numero postazioni PC nel laib i , possiamo scrivere in questo modo:

$$\forall_k \sum_{ij} P_i x_{ijk} \leq SG_k \quad (4.17)$$

Software/attrezzatura disponibile solo in un laboratorio

Un possibile problema che potrebbe presentarsi è che un solo laboratorio sia dotato di un determinato software, che per problemi di tempo è stato installato e verificato solo lì, oppure sia dotato di particolari attrezzature; allora il corso con queste esigenze deve necessariamente essere assegnato a quel laboratorio.

Se per esempio il corso $i = 9$ che utilizza un determinato software, disponibile solo al LAIB5 ($j = 6$), richiede l'utilizzo del laboratorio il lunedì dalle ore 10,00 alle 11,30 ($k = 1$) allora dobbiamo imporre la condizione:

$$x_{961} = 1 \quad (4.18)$$

Orario del LAIB1T e aula 5T

Se volessimo considerare anche le due aule utilizzate inizialmente dai corsi del *Collegio ICM* ma successivamente disponibili, nei moduli orari rimasti liberi, per i corsi degli altri *Collegi*, possiamo assegnare una "chiusura" nei moduli orari delle due aule occupati dai corsi del *Collegio ICM* così nei moduli rimasti liberi possiamo assegnare gli altri corsi.

Per esempio se al lunedì il LAIB1T (mettiamo $j = 7$) è occupato dai corsi per tutto il giorno ($k \in \{1..6\}$) scriviamo:

$$\forall_i \quad \forall_k \in \{1..6\} \quad \sum_i x_{i7k} = 0 \quad (4.19)$$

4.3.5 Funzione obiettivo

Definiamo la funzione obiettivo obj come la somma di tutti i corsi i assegnati nel laboratorio j e nel modulo orario richiesto k moltiplicato per il peso p che assegniamo ad ogni corso i :

$$obj = \sum_{ijk} p_i x_{ijk} \quad (4.20)$$

Visto che dobbiamo assegnare il massimo numero di corsi i al laboratorio j nello slot k allora scriviamo:

$$\max \text{obj} \tag{4.21}$$

Ma se consideriamo la variabile ausiliaria y_i allora scriviamo:

$$\text{obj1} = \sum_i p_i y_i \tag{4.22}$$

da massimizzare:

$$\max \text{obj1} \tag{4.23}$$

Dove p_i è il *peso* che assegniamo ad ogni corso e definisce la *priorità* in base alle esigenze volute dalla Commissione orario.

4.4 Complessità computazionale del nostro problema

In letteratura [5] c'è una certa confusione su quanto sia difficile il problema dell'assegnazione delle aule, perché la difficoltà non dipende solo dall'obiettivo o dalla dimensione del problema ma anche dalle preferenze dei corsi per le aule (per esempio, nel nostro caso, le tre squadre dei Corsi di Informatica del primo anno devono essere allocate in moduli orari consecutivi e nello stesso laboratorio); per molte università questo problema è facile per altre risulta quasi impossibile. La complessità computazionale del problema è ancora oggetto di studio perché per alcuni autori è possibile sviluppare un algoritmo *polinomiale*, per altri è possibile dimostrare che il problema generale è *NP-completo*; quindi per alcuni si risolve in tempo polinomiale, per altri invece si sviluppano approcci euristici.

Per chiarire questi concetti, riportiamo solo alcune definizioni, che fanno ampio riferimento al testo [32].

4.4.1 Algoritmo polinomiale

Si definisce un *algoritmo polinomiale*, un algoritmo A caratterizzato da una funzione di complessità temporale T_A che risulta essere $O(g(n))$, con $g(n)$ funzione polinomiale di n , (come per esempio n , n^3). Mentre non sono polinomiali quegli algoritmi dove $T_A(n) = O(2^n)$ oppure $O(n!)$. Questa distinzione è importante quando si considerano i tempi di esecuzione impiegati dagli algoritmi in funzione della dimensione dell'istanza n : in particolare un algoritmo non polinomiale risulta essere più veloce di quello polinomiale per valori di n molto piccoli, ma diventa proibitivo l'utilizzo di un algoritmo non polinomiale per valori di n sufficientemente grandi.

4.4.2 La classe P e la classe NP

Definizione (Classe P) : la classe P è l'insieme di tutti i problemi risolubili mediante algoritmi polinomiali.

I problemi appartenenti alla classe P , che possono essere risolti in un tempo di esecuzione limitato da una funzione della loro dimensione, sono spesso definiti anche problemi "facili".

Si consideri un problema decisionale Π (per il quale è richiesto semplicemente una risposta di tipo sì/no) e l'insieme delle istanze che ammettono risposta "sì" (*istanza-sì*).

Definizione (Classe NP) : la classe NP è l'insieme di tutti i problemi di decisione Π con le seguenti caratteristiche:

- ogni istanza-sì I ammette un certificato x_I ;
- esiste un polinomio $p(\cdot)$ tale che $|x_I| \leq p(|I|)$ per ogni istanza-sì I ;
- esiste un algoritmo di certificazione A_Π in grado di verificare ogni istanza-sì I per mezzo di x_I in tempo polinomiale in $|I|$.

E si definisce: un certificato di un'istanza-sì I di un problema decisionale Π una collezione di dati che, applicando un opportuno algoritmo di certificazione A_Π , permette di verificare che I è effettivamente un'istanza-sì.

La definizione di classe NP è stata in origine formalizzata utilizzando un concetto di algoritmo *non deterministico* polinomiale, infatti NP è l'acronimo di Non-Deterministic Polynomial.

4.4.3 Problemi NP-completi e NP-hard

Un modo per riconoscere i problemi *NP-completi* è dato dal seguente:

Teorema : un problema Π è *NP-completo* se:

- $\Pi \in NP$ e
- esiste Π' *NP-completo* tale che $\Pi' \propto \Pi$

dove $\Pi' \propto \Pi$ indica che un problema decisionale Π' è *polinomialmente riducibile* ad un altro problema Π ; e in termini di *complessità* equivale a dire che Π è *almeno difficile* quanto Π' .

Un problema è *NP-hard* quando si considera un problema Π di cui non è possibile verificare la prima condizione ma che soddisfi la seconda condizione del teorema precedente; il problema Π è quindi *almeno difficile* quanto un problema *NP-completo*, ma non si può garantire la sua appartenenza a NP .

Capitolo 5

Implementazione del modello con il solver

Dopo aver costruito il modello matematico, la ricerca della soluzione ottima può essere effettuata da appositi *motori di ottimizzazione*. Come risolutore si è scelto il software *Xpress-IVE* della FICOTM *Xpress Optimization Suite* [45] che implementa gli algoritmi risolutivi (per esempio il metodo *Branch and bound*) di diverse classi di problemi, *programmazione lineare* (PL) e *programmazione di interi misti* (MIP), con buoni risultati in termini di tempo. Inoltre presenta un'interfaccia facile da usare.

Il modello matematico è scritto in *linguaggio Mosel*, potente linguaggio di modellazione e di programmazione, che permette di dichiarare i parametri, di definire le variabili decisionali, i vincoli e i vari dati. Dato che *Mosel* non è un risolutore occorre *chiamare*, tramite *moduli*, determinati *ambienti* per poter trovare le soluzioni. Nel nostro caso utilizziamo i moduli *mmxprs* e *mmive*. Il primo modulo permette l'accesso a *Xpress-Optimizer*, che fornisce un numero di routine relative all'ottimizzazione (da quelle per trovare una soluzione a quelle per le funzioni di gestione del taglio): per esempio se vogliamo *massimizzare* una funzione obiettivo dobbiamo chiamare *Xpress-Optimizer* per massimizzare l'espressione, con la dichiarazione "*uses mmxprs*" e la procedura "*maximize*". Il secondo modulo viene utilizzato dall'ambiente di visualizzazione integrato *Xpress-MPXpress-IVE* per estendere le capacità grafiche, consentendo all'utente di visualizzare grafici di funzioni, diagrammi e altro. Tramite il comando "*run*" il programma legge i dati di ingresso da file in formato testo (per esempio TXT, CSV), assegna i valori ai dati inizializzati, scrive il modello di programmazione PLI 0/1 e chiama la funzione che implementa gli algoritmi risolutivi. Il "*run*" termina quando l'algoritmo ha trovato la soluzione ottimale, o eventualmente quando lo arrestiamo noi, per esempio se vediamo che ha impiegato un tempo ragionevole.

5.1 Metodo Branch and Bound

Per completezza spieghiamo come tecnica di soluzione per i problemi PLI il metodo *Branch and Bound* (BB), introdotto agli inizi degli anni '60.

Il BB fornisce un metodo di ricerca della *soluzione ottima* esplorando *parzialmente* l'insieme delle soluzioni ammissibili; in particolare la "funzione obiettivo viene calcolata per un (piccolo) sottoinsieme delle soluzioni ammissibili con la proprietà di contenere almeno una soluzione ottima".

In dettaglio: *si partiziona (branch) un insieme S_0 delle soluzioni ammissibili del problema iniziale di minimo P_0 in sottoinsiemi S_1, S_2, \dots , via via più ristretti. Per ognuno di questi sottoinsiemi viene calcolato un limite inferiore (lower bound) del valore ottimo della funzione obiettivo. Quei sottoinsiemi di soluzioni ammissibili per il quale il valore del limite non sia migliore del valore di funzione obiettivo di qualche soluzione ammissibile e nota \bar{x}^* di P_0 sono esclusi da ulteriori partizioni. La procedura si arresta quando la migliore soluzione \bar{x}^* , disponibile fino a quel momento, ha un valore di funzione obiettivo non peggiore del bound calcolato per ogni rimanente sottoinsieme.*

Cerchiamo di chiarire la procedura con un esempio informale. Supponiamo di dover trovare lo studente più alto del Politecnico di Torino, allora per risolvere il problema con il metodo BB si può decomporre il problema in più sottoproblemi chiedendo ai Presidi dei Dipartimenti qual è lo studente più alto e poi scegliere il più alto tra questi studenti. Però in alcuni casi la risoluzione di questi sottoproblemi può essere facile, come nel caso di un Dipartimento con pochi studenti iscritti; in altri casi invece il preside di un Dipartimento potrebbe non essere in grado di fornire facilmente il nome dello studente più alto. In questo caso si può nuovamente scomporre il problema per esempio chiedendo ai vari Collegi di fornire il nome dello studente più alto; da questi si potrà risalire al nome più alto del Dipartimento e così via. Per ulteriori approfondimenti si rimanda ai testi [32], [10].

5.2 Dati di input

Consideriamo i corsi della Triennale e Magistrale solo del primo periodo didattico (dell'anno accademico 2017 – 2018) perché è il più critico, e solo i laboratori della sede Centrale/Cittadella perché si verificano maggiormente le sovrapposizioni delle richieste.

Tutti i corsi hanno una numerosità che rispetta la capienza dei laboratori. Ma se ci fosse un corso con un numero di studenti (per esempio 200) superiore alla capienza massima del laboratorio più grande (ovvero il LAIB1 con capienza massima di 156 studenti) non lo consideriamo perché è impossibile assegnarlo; sarà compito del Referente di trovare una soluzione, per esempio richiedendo la suddivisione a squadre.

Eseguiamo la nostra analisi con varie istanze, a seconda dei valori che assegniamo ai parametri in input, e valutiamo i risultati ottenuti (come file di output in formato TXT oppure in XLS) confrontandoli con quelli ottenuti manualmente.

Visto che le attività possono avere durata differente: consideriamo corsi da tre moduli orari consecutivi, corsi da due moduli orari consecutivi e corsi da un solo modulo. Per comodità chiameremo "slot" il modulo orario corrispondente a un'ora e trenta minuti.

Per i dati di ingresso abbiamo tre file in formato testo (nel nostro caso TXT):

- file dei **Corsi** con: nome corso, numero, peso, numero studenti, slot orario richiesto;
- file dei **Laib** con: nome laib, numero, numero PC, capienza massima;

- file degli **Slot** orari richiesti con: nome slot, nome giorno, fascia oraria.

Inizialmente dimostriamo che non c'è soluzione se consideriamo il vincolo "ad ogni corso dobbiamo assegnare un laboratorio nel modulo orario richiesto", di conseguenza per proseguire dobbiamo per forza "rilassare" il vincolo e massimizzare la funzione obiettivo, ovvero "assegnare il massimo numero di corsi al laboratorio nel modulo orario richiesto".

Il programma creato permette di trovare:

- per tutte le istanze:
 - quanti e quali corsi vengono assegnati ai laboratori;
 - quanti e quali corsi rimangono "fuori" dall'assegnazione;
 - quali sono i laboratori e gli slot rimasti liberi.
- solo per l'ultima istanza: definiamo altre fasce orarie di *riserva* per i corsi rimasti "fuori" e troviamo quanti e quali di questi riusciamo ad assegnare nei laboratori e negli slot rimasti liberi.

5.3 Prima istanza

In questa prima istanza consideriamo solo le richieste relative ai corsi della Laurea Triennale di Ingegneria e di Architettura (vedere anche la tabella 4.4 a pag.49), per un totale di 42 corsi che richiedono 99 slot; in particolare abbiamo:

- 21 corsi da tre slot consecutivi: 63 slot in totale;
- 13 corsi da due slot consecutivi: 26 slot in totale;
- 10 corsi da un solo slot

Consideriamo solo i laboratori di corso Duca: il LAIB1, suddiviso nelle due sale (LAIB1a e LAIB1b), il LAIB3, il LAIB4 e il LAIB5. A seconda della numerosità degli studenti e del numero delle richieste possiamo assegnare o quattro o cinque corsi nello stesso slot. Ogni laboratorio però dispone di 34 slot a settimana quindi in totale abbiamo la possibilità di assegnarne da un minimo di 136 slot ad un massimo di 170 slot; in teoria il problema risulta di facile soluzione perché la disponibilità dei laboratori è ampiamente superiore alle richieste.

Il modello utilizzato per questa istanza ha tutti i vincoli riportati nel quarto capitolo (Assegnamento, Caso particolare del LAIB1, Capienza Laib, e Corsi da tre slot e da due slot consecutivi e nello stesso laboratorio) ma non consideriamo nessun vincolo aggiuntivo.

Tuttavia, al primo "Run", Figura 5.1 a pag. 64, dimostriamo che può esserci soluzione perché abbiamo considerato il primo vincolo, ovvero l'equazione (4.4) che qui riportiamo:

$$\forall_i \quad \sum_{jk} x_{ijk} = 1$$

Stats			
Matrix:		Presolved:	
Rows(constraints):	21111	Rows(constraints):	111
Columns(variables):	20295	Columns(variables):	196
Nonzero elements:	112431	Nonzero elements:	510
Global entities:	20295	Global entities:	196
Sets:	0	Sets:	0
Set members:	0	Set members:	0
Overall status: Finished global search.			
LP relaxation:		Global search:	
Algorithm:	Simplex primal	Current node:	1
Simplex iterations:	0	Depth:	1
Objective:	0	Active nodes:	0
Status:	Unfinished	Best bound:	-1e+040
Time:	0.0s	Best solution:	-1e+040
		Gap:	0%
		Status:	No feasible integer solution exists.
		Time:	0.2s
Time overheads:			
Progress graphs:	0.3s		
Writing output:	0.0s		
Flushing:	0.0s		
Updating status:	0.3s		
Output/Input Stats Matrix Solutions Objective MIP search BB tree User graph			
IIS			

Figura 5.1. Run 1: non c'è soluzione

Quindi dobbiamo per forza rilassare questo vincolo e cercare di assegnare quanti più corsi al laboratorio, sfruttando il secondo vincolo (4.5):

$$\forall_i \quad \sum_{jk} x_{ijk} \leq 1$$

e la funzione obiettivo (5.3):

$$\text{obj} = \sum_{ijk} p_i x_{ijk}$$

che dobbiamo massimizzare.

Il *peso* p_i del corso è legato al fatto che, a seconda delle richieste della Commissione Orario, alcuni corsi possono avere delle "priorità", ovvero devono essere assegnati prima degli altri, come nel caso delle 60 squadre di Informatica del primo anno che hanno un orario fissato e consolidato da diversi anni. Di conseguenza decidiamo di attribuire a questi corsi il massimo valore, per esempio 500; mentre per gli altri corsi scegliamo di dare un peso a seconda del numero degli slot, perché il *Referente orario* ci segnala di avere più difficoltà a spostare un corso da tre slot consecutivi rispetto a uno con un solo slot. Quindi scegliamo di assegnare questi pesi: 500 ai corsi di Informatica da tre slot consecutivi, 100 agli altri corsi da tre slot consecutivi, 50 ai corsi da due slot consecutivi e 10 ai corsi da un solo slot.

Per il secondo "Run" verifichiamo che effettivamente c'è soluzione, Figura 5.2 a pag. 65, e il numero dei corsi che riusciamo ad assegnare ai laboratori sono 98 su 99 impiegando 0.2 secondi.

Esaminando i risultati non è stato assegnato un corso nello slot 9 (martedì dalle 11,30 alle 13,00) perché in quel modulo risultano cinque richieste: quindi per poter assegnare tutti i corsi, due di questi devono necessariamente essere allocati nelle due sale del LAIB1, ma questo non è possibile perché superano la capienza del laboratorio.

Stats			
Matrix:		Presolved:	
Rows(constraints):	21111	Rows(constraints):	108
Columns(variables):	20295	Columns(variables):	228
Nonzero elements:	112431	Nonzero elements:	534
Global entities:	20295	Global entities:	228
Sets:	0	Sets:	0
Set members:	0	Set members:	0
Overall status: Finished global search.			
LP relaxation:		Global search:	
Algorithm:	Simplex primal	Current node:	1
Simplex iterations:	30	Depth:	1
Objective:	31690	Active nodes:	0
Status:	Unfinished	Best bound:	31690
Time:	0.1s	Best solution:	31690
		Gap:	0%
		Status:	Solution is optimal
		Time:	0.2s
Time overheads:			
Progress graphs:	0.5s		
Writing output:	0.0s		
Pausing:	0.0s		
Updating status:	0.5s		
Output/Input Stats Matrix Solutions Objective MIP search BB tree User graph			
IIS			

Figura 5.2. Run 2: soluzione ottimale 31690

In realtà però questo problema è risolvibile perché solitamente si tende ad avere un margine di *tolleranza* sulla capienza, dato che la numerosità degli studenti tende a diminuire entro le prime settimane del semestre. Quindi possiamo diminuire il numero degli studenti del corso rimasto "fuori" oppure diminuire il numero degli studenti di un altro corso, oppure possiamo cambiare ulteriormente i pesi dando la priorità ad un corso rispetto ad un altro; e ripetere l'elaborazione. Ma in ogni caso la scelta deve essere valutata tra il Referente Orario e il Referente dei laboratori.

Se decidiamo di diminuire la numerosità del corso rimasto "fuori" e lanciamo il terzo "Run", Figura 5.3, verifichiamo che in questo caso l'assegnazione è completa e ovviamente la soluzione ottimale è cambiata rispetto a quella precedente, ma notiamo che per sistemare quei cinque corsi del martedì nella fascia oraria 11,30 – 13,00, nove corsi hanno cambiato laboratorio rispetto al secondo "Run".

Stats			
Matrix:		Presolved:	
Rows(constraints):	21111	Rows(constraints):	110
Columns(variables):	20295	Columns(variables):	232
Nonzero elements:	112431	Nonzero elements:	554
Global entities:	20295	Global entities:	232
Sets:	0	Sets:	0
Set members:	0	Set members:	0
Overall status: Finished global search.			
LP relaxation:		Global search:	
Algorithm:	Simplex primal	Current node:	1
Simplex iterations:	30	Depth:	1
Objective:	31700	Active nodes:	0
Status:	Unfinished	Best bound:	31700
Time:	0.3s	Best solution:	31700
		Gap:	0%
		Status:	Solution is optimal
		Time:	0.3s
Time overheads:			
Progress graphs:	0.5s		
Writing output:	0.0s		
Pausing:	0.0s		
Updating status:	0.5s		
Output/Input Stats Matrix Solutions Objective MIP search BB tree User graph			
IIS			

Figura 5.3. Run 3: soluzione ottimale 31700

Il programma propone un'assegnazione "automatica" dei laboratori, quindi per vedere se la situazione cambia proviamo a cambiare l'ordine dei corsi inseriti nel file di testo: per esempio mettiamo in ordine crescente da lunedì a venerdì i corsi di Informatica, e in ordine

alfabetico i corsi da due slot consecutivi e quelli da un solo slot.

Nel quarto "Run", la soluzione ottimale è la stessa del secondo "Run", ma per cinque corsi è cambiato il tipo di laboratorio.

Non facciamo ulteriori prove perché i laboratori non sono utilizzati solo dai corsi della Laurea Triennale; comunque già con un problema semplificato possiamo avere tre opzioni di scelta.

5.4 Seconda istanza

Nella seconda istanza esaminiamo il nostro vero problema, considerando tutti i corsi della Laurea Triennale e Magistrale di Ingegneria e di Architettura, per un totale di 202 moduli orari richiesti, in particolare:

- 22 corsi da tre slot consecutivi: 66 slot in totale;
- 51 corsi da due slot consecutivi: 102 slot in totale;
- 34 corsi da un solo slot.

Diamo ulteriori dettagli sulla distribuzione del numero di slot richiesti per tutti i 34 slot che riportiamo nella Tabella 5.1 a pag. 67.

Ricordando che lo slot 1 corrisponde a lunedì dalle 10,00 alle 11,30 fino allo slot 34 che corrisponde a venerdì dalle 17,30 alla 19,00.

Oltre ai laboratori di corso Duca utilizziamo anche i due laboratori di via Boggio (LAIB2B, LAIB3B); quindi a seconda del numero delle richieste e del numero degli studenti possiamo assegnare nello stesso momento o sei o sette corsi. Ogni laboratorio è suddiviso in 34 moduli orari a settimana, quindi in totale abbiamo la possibilità di assegnarne da un minimo di 204 slot ad un massimo di 238 slot, dovendo allocare 202 richieste esiste la possibilità che emergano sovrapposizioni.

Infatti, già valutando i dati di ingresso le situazioni critiche sono:

- in 8 slot ci sono 6 richieste in parallelo: bisogna fare attenzione alla numerosità degli studenti che rispetti la capienza dei sei laboratori a disposizione (vincolo **Capienza Laib**);
- in 9 slot ci sono 7 richieste in parallelo: quindi non solo bisogna fare attenzione alla numerosità degli studenti (vincolo **Capienza Laib**) ma occorre vedere se due di questi corsi possono essere assegnati alle due sale del LAIB1 (vincolo **Caso particolare del LAIB1**);
- in 5 slot ci sono 8 richieste: quindi di sicuro almeno una deve essere spostata;
- in uno slot ci sono ben 9 richieste in parallelo, quindi o due o tre corsi devono essere sicuramente spostati.

Tabella 5.1. Distribuzione slot richiesti nel primo p.d. 2017 – 2018

Giorno	Nome Slot	Numero slot richiesti
lunedì	1	5
	2	7
	3	7
	4	8
	5	6
	6	3
martedì	7	5
	8	8
	9	8
	10	7
	11	6
	12	5
	13	1
mercoledì	14	4
	15	7
	16	6
	17	5
	18	6
	19	7
	20	4
giovedì	21	6
	22	8
	23	9
	24	7
	25	6
	26	8
	27	3
venerdì	28	7
	29	7
	30	7
	31	5
	32	6
	33	6
	34	2

Anche in questa seconda istanza attribuiamo il peso maggiore ai corsi di Informatica e per gli altri corsi scegliamo di dare un peso a seconda del numero di moduli orari consecutivi; assegniamo nuovamente: peso 500 ai corsi di Informatica da tre slot consecutivi, peso 100 agli altri corsi da tre slot consecutivi, peso 50 ai corsi da due slot consecutivi e peso 10 ai corsi da un solo slot.

Per il primo "Run" utilizziamo il modello con i vincoli generali ma nessun vincolo aggiuntivo e dopo l'elaborazione siamo riusciti ad assegnare 192 richieste su 202 in 0,4 secondi.

Rimangono "fuori":

- tre corsi da due slot consecutivi:
 - un corso negli slot 4 e 5 del lunedì: perché nello slot 4 ci sono 8 richieste;
 - un corso negli slot 8 e 9 del martedì: perché in entrambi gli slot ci sono 8 richieste;
 - un corso negli slot 22 e 23 del giovedì: perché nello slot 22 abbiamo 8 richieste e nello slot 23 ne abbiamo ben 9;
- quattro corsi da uno slot:
 - un corso nello slot 3 del lunedì: perché pur avendo 7 richieste quel corso ha 130 studenti e quindi in quello slot posso allocare al massimo 6 richieste;
 - un corso nello slot 4 del lunedì: perché ho 8 richieste e quindi una deve per forza essere spostata;
 - un corso nello slot 23 del giovedì: perché ne abbiamo ben 9 di richieste;
 - un corso nello slot 26 del giovedì: perché ho 8 richieste e quindi una deve per forza essere spostata;

Per un totale di 10 slot non assegnati; come previsto le sovrapposizioni si sono presentate.

Per un secondo "Run" proviamo ad aggiungere il vincolo **Software/attrezzatura disponibile solo in un laboratorio**: quindi decidiamo che due corsi da due slot consecutivi devono per forza andare in un laboratorio specifico, mettiamo il LAIB5, perché un corso utilizza un software che al momento è stato installato solo lì, e un altro corso ha lasciato in custodia delle attrezzature che devono essere utilizzate durante l'esercitazione.

Dopo l'elaborazione, ovviamente la soluzione ottimale è sempre la stessa e quindi assegniamo sempre 192 richieste su 202. Ma confrontando i risultati con il primo Run, notiamo che i corsi rimasti "fuori" sono sempre gli stessi, mentre negli altri slot i corsi hanno cambiato il tipo di laboratorio assegnato e in alcuni casi anche la sede, in particolare:

- 10 corsi di Informatica (da tre slot consecutivi) hanno cambiato laboratorio di cui: 6 hanno cambiato anche la sede, 4 hanno cambiato il laboratorio ma non la sede;
- un corso (da tre slot consecutivi) è rimasto sempre nella stessa sede ma ha cambiato laboratorio;

- 30 corsi da due slot consecutivi hanno cambiato laboratorio di cui: 11 hanno cambiato anche la sede, mentre 19 sono rimasti nella stessa sede ma laboratorio diverso;
- 9 corsi da un solo slot hanno cambiato laboratorio di cui: 3 anche la sede e 6 solo il laboratorio;

Per un terzo "Run" proviamo a modificare i dati di ingresso mettendoli in ordine di giorno e di fascia oraria richiesta, e utilizziamo il modello con solo i vincoli generali ma nessun vincolo aggiuntivo.

Dopo l'elaborazione, la soluzione ottimale è sempre la stessa e riusciamo ad assegnare 192 richieste su 202.

Ma confrontando il risultato rispetto ai due primi Run notiamo che questa volta dei corsi rimasti "fuori": tre corsi da due slot consecutivi sono diversi ma sono sempre negli stessi slot degli altri Run precedenti, mentre i quattro corsi da un solo slot sono gli stessi dei precedenti.

Per quanto riguarda i corsi che hanno cambiato il tipo di laboratorio confrontiamo il risultato tra i tre Run:

- dei 20 corsi di Informatica (da tre slot consecutivi): 6 corsi hanno cambiato laboratorio rispetto agli altri due, 14 invece hanno dato lo stesso risultato dei precedenti (5 uguali al primo e secondo Run, 5 uguali al primo Run e 4 uguali al secondo Run);
- dei due corsi (da tre slot consecutivi): un corso ha cambiato laboratorio rispetto agli altri due Run, e un corso è uguale al primo Run;
- dei 46 corsi da due slot consecutivi: 18 corsi hanno cambiato laboratorio rispetto agli altri due, 28 corsi hanno dato lo stesso laboratorio dei precedenti (11 uguali al primo e secondo Run, 9 uguali al primo Run e 8 uguali al secondo Run);
- dei 30 corsi da un solo slot: 6 corsi hanno cambiato laboratorio rispetto agli altri due, 24 invece hanno dato lo stesso risultato dei precedenti (17 uguali al primo e secondo Run, 4 uguali al primo Run e 3 uguali al secondo Run);

Per il confronto tra la soluzione realizzata manualmente e i risultati ottenuti dal programma (con questi tre Run) consideriamo solo i corsi rimasti "fuori", perché la scelta del tipo di laboratorio che effettivamente è stato assegnato ai vari corsi è legata anche ad altri fattori che in questa tesi non potevamo considerare perché parte del problema più complesso: ovvero tenere conto anche dell'orario delle lezioni in aula, e in particolare controllare non solo l'aula ma anche la sede assegnata per i corsi in aula nelle fasce orarie adiacenti ai corsi svolti in laboratorio.

Però dei corsi rimasti "fuori" riscontriamo quanto segue:

- dei tre corsi da due slot consecutivi:
 - il corso negli slot 4 e 5 del lunedì (perché nello slot 4 ci sono 8 richieste): è diverso ma in ogni caso ha dovuto cambiare orario;

- il corso negli slot 8 e 9 del martedì (perché in entrambi gli slot ci sono 8 richieste) è diverso ma in realtà sono stati assegnati tutti perché uno di questi è stato assegnato in un'aula attrezzata con i computer;
- il corso negli slot 22 e 23 del giovedì (perché nello slot 22 abbiamo 8 richieste e nello slot 23 ne abbiamo ben 9): è diverso ma in ogni caso ha cambiato orario;
- dei quattro corsi da uno slot:
 - il corso nello slot 3 del lunedì anche nella realtà è stato spostato, perché pur avendo 7 richieste quel corso ha 130 studenti e quindi in quello slot sono stati allocati solo 6 richieste e non sono state utilizzate le due sale del LAIB1;
 - il corso nello slot 4 del lunedì in realtà non ha cambiato orario perché è stato spostato in un'aula attrezzata con i computer;
 - il corso nello slot 23 del giovedì (perché abbiamo ben 9 di richieste) invece è stato allocato perché in quel caso il Referente orario non poteva modificarlo perché con molti studenti e quindi nella realtà è stato spostato un corso da due slot consecutivi (23 e 24) con un numero di studenti molto basso;
 - un corso nello slot 26 del giovedì (perché abbiamo 8 richieste) è diverso ma in ogni caso ha cambiato orario.

5.5 Terza istanza

Per questa ultima istanza siamo partiti dai risultati ottenuti nella seconda istanza, dal primo e dal secondo Run, dove abbiamo assegnato il laboratorio a 192 richieste su 202.

In questo caso però il modello cambia perché per i corsi rimasti "fuori" assegniamo un altro orario scegliendo tra gli slot rimasti liberi e quindi rilanciamo l'elaborazione. In entrambi i casi l'assegnazione risulta completata totalmente e in pochissimo tempo, inferiore al secondo.

In conclusione tutte queste prove dimostrano ampiamente che automatizzare, anche solo l'assegnazione dei laboratori, propone ottime soluzioni che possono essere valutate dai Referenti Orari e dal Referente dei laboratori per scegliere quella migliore e in breve tempo; a differenza del mese e mezzo (se non due mesi) di lavoro impiegato, dalla raccolta dei dati fino al caricamento dell'Orario sull'applicativo.

Capitolo 6

Possibili miglioramenti sull'applicativo web di Ateneo

I problemi descritti nel secondo capitolo dimostrano che il software **Gestione Orari** [3] è utile per pubblicare e consultare l'orario generato manualmente, su un'interfaccia grafica, ma non permette una sorta di **simulazione** per verificare le reali sovrapposizioni tra le varie materie dei vari Corsi di Laurea o per visualizzare gli eventuali conflitti legati alle indisponibilità del docente.

In questo capitolo cerchiamo di dare alcune indicazioni di possibili miglioramenti sull'applicativo web che sono "oggettivi" perché legati alle problematiche descritte nel secondo capitolo, e "soggettivi" perché frutto dell'esperienza personale di tutte le parti coinvolte, dai Referenti orari al Referente dell'assegnazione delle risorse, in particolare quello dei laboratori.

Le interfacce grafiche e i manuali di istruzione dei software presentati nel terzo capitolo ci hanno aiutato a trovare quelle migliorie per la nostra interfaccia.

6.1 Miglioramenti oggettivi

Per miglioramenti oggettivi intendiamo i miglioramenti legati:

- alla scheda online (**Modulo di predisposizione orari**) che deve essere compilata dai docenti e al file excel dei dati estrapolati che viene utilizzato dai Referenti Orari;
- al software **Gestione Orari** utilizzato dai Referenti Orari e dai Referenti delle risorse per caricare l'orario nell'area *Gestione template orario*; con particolare attenzione all'area *Gestione richieste e conflitti* e all'area *Visualizzazione Log e operazioni*, accessibili solo dal personale abilitato;

6.1.1 Modulo di predisposizione orari e file dei dati estrapolati

Innanzitutto sarebbe utile scrivere un manuale di istruzione per la compilazione del Modulo di predisposizione orari, che ad oggi non esiste; inoltre occorre che nelle varie sezioni

della scheda ci siano tasti per un "help in linea", per guidare l'utente passo dopo passo; e siano presenti più dettagli.

Nella sezione **dati incarico** della scheda del Passo 1 (Figura 2.2 a pag.8) bisognerebbe inserire:

- il *numero degli studenti iscritti* dell'anno precedente con l'eventuale aggiunta della percentuale, stabilita dalla Commissione orari, per l'anno accademico in corso in modo tale da avere una stima degli studenti "previsti";
- i crediti relativi al corso (CFU);
- il numero di crediti che il docente deve coprire e quelli assegnati agli assistenti;
- il *monte ore settimanale* e il numero dei moduli settimanali suddivisi per la lezione, per l'esercitazione in aula e in laboratorio.

Nella sezione **informazioni aggiuntive** i campi non pre-strutturati, "*eventuali e motivate incompatibilità di orario dei docenti coinvolti*" e "*ulteriori comunicazioni sull'organizzazione del corso che possono essere utili in fase di predisposizione degli orari*", o non vengono compilati oppure vengono compilati con troppi dettagli descrittivi e spesso sono ricopiati in entrambi i campi. Sarebbe quindi utile predisporre dei campi a "scelta":

- per l'*incompatibilità di orario* il docente dovrebbe poter selezionare da una lista o un menu a tendina i corsi di cui è titolare e indicare anche gli assistenti;
- per l'*indisponibilità* o le *preferenze orarie* il docente dovrebbe poter scegliere il giorno o i giorni e il modulo orario in cui non è disponibile oppure indicare i giorni e i moduli orari in cui preferirebbe fare lezione.

In questo modo si potrebbero visualizzare eventuali conflitti legati alla materia o alle indisponibilità del docente.

Nella sezione **organizzazione dei laboratori**, della scheda Passo 2 (Figura 2.3 a pag.10) viene riportata la stessa sezione dei **dati incarico** con gli stessi dati mancanti descritti prima.

Nella sezione **selezione laboratorio** il docente non segnala le caratteristiche del laboratorio di cui ha bisogno, ma sceglie il laboratorio dell'Ateneo che vorrebbe gli fosse assegnato; anche se è presente la nota che potrebbe venirgli assegnato un altro laboratorio simile in caso di indisponibilità. Bisognerebbe aggiungere questi dati mancanti:

- il *numero massimo delle postazioni PC* in dotazione nel laboratorio selezionato, informazione che potrebbe essere utile al docente per sapere quanti computer può effettivamente utilizzare per la sua esercitazione; per ottenere queste informazioni il docente deve consultare un altro sito web;
- il *numero degli studenti previsti/stimati*.

Una volta che il docente compila e invia la scheda online, il Referente dei laboratori riceve in anteprima una mail con i dati che corrispondono a quest'ultima sezione, ma mancano i dati sulla *numerosità del corso*. Se un docente chiede per esempio il LAIB4, che ha una capienza di 115 studenti, e nelle note scrive che il corso sarà suddiviso in tre squadre, non si capisce se il corso al massimo può avere 115 studenti diviso poi in tre squadre o se ogni squadra è composta al massimo da 115 studenti.

Nel **Tipo di utilizzo** vengono date due scelte: "*prevalente settimanale*" e "*numero limitato di settimane*" e tutti i dettagli (per esempio: se il laboratorio viene utilizzato a settimane alterne oppure se si il laboratorio viene utilizzato dalla quarta settimana a fine semestre) il docente deve indicarli nel campo *note*, ma sarebbe più conveniente dare un'opzione di scelta, per esempio una griglia formata da quattordici celle, che indicano le settimane del periodo didattico, così il docente deve solo selezionare le settimane in cui utilizzerà il laboratorio, come indicato in Figura 6.1.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
x		x		x		x		x		x		x	

1	2	3	4	5	6	7	8	9	10	11	12	13	14
			x	x	x	x	x	x	x	x	x	x	x

Figura 6.1. Scelta settimane di utilizzo

Per indicare il **software/attrezzature** di utilizzo, è presente un altro campo non pre-strutturato e questo non è funzionale perché nella maggior parte dei casi o non viene compilato oppure le indicazioni sono sempre generiche. La differenza tra un'aula e un laboratorio è proprio dovuta al fatto che il docente deve poter utilizzare un determinato software, quindi deve sapere esattamente se il software di cui ha bisogno è presente e correttamente funzionante, la versione e il numero delle licenze disponibili.

Per l'anno accademico 2018 – 2019 è stata introdotta questa modifica: se il docente sceglie uno dei laboratori dell'*Area IT* viene inviata in automatico una mail al docente con tutte le indicazioni del link dove selezionare il software, da un elenco di tutti quelli disponibili, se non è presente può compilare un campo e inviare direttamente la richiesta (tramite un tasto) al servizio preposto. Miglioria funzionale ma non ottimale, perché il docente deve leggere un'altra mail, andare in un'altra area per controllare e selezionare il software.

Sarebbe più utile avere già nella scheda online, dopo aver scelto il laboratorio dell'*Area IT*, il link descritto prima; oppure avere un menu a tendina e l'elenco dei software disponibili da selezionare e nel caso non fosse presente avere il campo note dove scrivere il software necessario e il tasto per l'invio della richiesta.

La scheda online rimane "aperta" alla compilazione per un certo periodo; dopo la scadenza il Referente Orario estrapola tutti i dati in un file in formato excel, composto da ben 33 colonne relative al corso e 6 colonne relative agli esami. Alcune di queste colonne contengono i dati forniti dall'Area Gestione Didattica, non modificabili, quali:

matricola titolare, cognome, nome, collaboratori, codice insegnamento, numcor, tipologia - tipo laurea - anno - alfabetica - nome sede, titolo materia, periodo didattico, docente titolare, email; e ancora: numero blocchi lezione, numero squadre esercitazione, numero blocchi esercitazione per squadra, numero squadre laboratorio, numero blocchi laboratorio per squadra.

Poi abbiamo le colonne che indicano i dati logistici (sede e indirizzo) del laboratorio selezionato e infine le colonne corrispondenti ai campi compilati e selezionati dal docente.

Quindi abbiamo un insieme di dati dall'Area Gestione Didattica non modificabili, ma che vengono "cambiati" dalle preferenze segnalate dai docenti e altri dati che risultano ininfluenti per la pianificazione di un orario, come l'indirizzo mail del docente o l'indirizzo logistico dei laboratori.

Per esempio se dalla Didattica risulta che un corso ha: 5 blocchi per la lezione, 1 blocco per le esercitazioni per 2 squadre, 1 blocco per il laboratorio per 3 squadre, in totale quel corso ha bisogno di 10 blocchi (un blocco corrisponde a un modulo orario di un'ora e trenta minuti). Ma se il docente nel campo *modalità utilizzo* scrive: "*Lezioni in 1 blocco da 3 ore ed 1 da 1,5 ore; Esercitazioni in aula in blocchi da 3 ore; Esercitazioni in laboratorio (LAIB) in blocchi da 3 ore*"; e invece dall'orario pubblicato si vedono allocati: 2 blocchi come lezione in aula, 10 blocchi come esercitazione in aula, 6 blocchi come esercitazione ai laboratori per un totale di 18 blocchi, i conti non tornano!

Quindi dal file estrapolato dai Referenti orari devono essere ben chiari e presenti questi dati:

- il **numero dei crediti** (CFU) che potrebbe essere utile per chiarire effettivamente quanti moduli orari deve avere quel corso, e il numero di crediti che il docente e gli assegnisti devono coprire;
- il **monte ore settimanale** delle lezioni, delle esercitazioni in aula e in laboratorio (dato fornito dalla Didattica);
- il **numero di blocchi settimanale** della lezione, dell'esercitazione in aula, dell'esercitazione in laboratorio forniti dalla Didattica e anche quelli indicati dal docente;
- **numero degli studenti iscritti** dell'anno precedente con la percentuale stabilita dalla Commissione per l'anno accademico da pianificare, per avere una stima del numero degli studenti per assegnare le aule e i laboratori più adeguati.

In questo modo i Referenti Orari ottengono queste informazioni senza doverle chiedere direttamente ai docenti o cercarle nelle varie aree della *Gestione didattica*, consultando tutti i corsi erogati dal proprio Collegio.

6.1.2 Software Gestione Orari

L'utilizzo di questo software è limitato agli sviluppatori dell'applicativo, ad alcuni Referenti Orari e ai Referenti dell'assegnazione delle risorse, aule e laboratori, che sono incaricati di inserire tutto l'orario.

Durante la fase del caricamento dei blocchi orari nell'area **Gestione template orari** vengono segnalati dei warning che però non impediscono di proseguire e di inserire l'orario.

Uno di questi messaggi di errore segnala che un "*blocco è andato in conflitto con il template orario di altri Corsi di Laurea*", Figura 2.9 di pag. 19; ma questo conflitto non è visibile in nessun'altra schermata.

Potrebbe essere utile avere un'area, durante la fase di caricamento dell'orario e prima della pubblicazione, dove visualizzare tutti i conflitti per confermarli o modificarli, simile all'area **Gestione richieste e conflitti** (solo dopo la pubblicazione dell'orario), dove sono visibili ma non modificabili i conflitti tra un blocco caricato da template e un blocco caricato da calendario che corrisponde ad un evento (ovvero workshop, concorso, Altri eventi Didattici, manutenzione, pulizie o altro), quindi

L'altro messaggio di errore segnala sempre che "*si è superato il 5% dell'offerta formativa*", Figura 2.10 a pag. 20, qualunque sia il numero di blocchi aggiunti o tolti legati alla materia nella sezione *Gestione blocchi*. E man mano che si caricano i blocchetti nella "settimana tipo", il numero corrispondente alle *ore rimanenti* diminuisce fino ad azzerarsi, altrimenti l'orario non può essere pubblicato. Ma in questo modo si perde l'informazione del numero totale dei blocchi che deve avere la lezione e l'esercitazione. Quindi occorre che sia presente e ben fissato:

- il massimo numero di blocchi per la lezione, per l'esercitazione in aula e in laboratorio,
- il numero di blocchi che si è aggiunto o tolto,
- un campo dove spiegare perché si sono modificati i blocchi orari.

Una volta pubblicato l'orario le ulteriori modifiche da *template* o da *calendario* possono essere fatte dal personale abilitato.

Ma in realtà anche il docente, dalla propria pagina personale, può accedere ad un'altra area dell'applicativo, che corrisponde ad un'integrazione del software, **Orari di lezioni e prenotazioni aule** [4], che gli permette la prenotazione (ma al momento è stata disabilitata) e il rilascio delle risorse che gli sono state assegnate. In questi anni solo alcuni docenti hanno utilizzato questa procedura per cancellare alcuni blocchi durante il periodo didattico, perché la maggior parte dei docenti non ne è a conoscenza.

Però la cancellazione o modifica avviene immediatamente e non viene inviata nessuna comunicazione (al momento è stato disabilitato l'invio della mail) al servizio Logistica per le aule e all'Area IT per i laboratori. Queste modifiche/cancellazioni (di tutti gli anni accademici) sono elencate nell'Area **Visualizzazione log operazioni**.

In quest'area però non vengono visualizzate le cancellazioni o modifiche fatte da tutti gli utenti abilitati all'utilizzo del software, quindi se un utente per sbaglio cancella un blocco da Calendario questo non è segnalato da nessuna parte.

Allora sarebbe utile che in quest'area fossero visualizzate tutte le modifiche fatte dagli utenti abilitati, con ulteriori filtri di ricerca; al momento ci sono solo tre filtri: *titolo materia, aula, tipo di operazione*. Potrebbero essere utili filtri per:

- *anno accademico*: perché al momento risultano elencate tutte le modifiche di tutti gli anni accademici dal 2011 ad oggi;

- *utente*: perché potrebbe essere necessario visualizzare l'elenco delle modifiche che un utente ha dovuto fare per un corso e avere così un'idea per l'anno successivo dell'effettivo utilizzo della risorsa.

6.2 Miglioramenti soggettivi

I miglioramenti "soggettivi" che andiamo ad indicare in questa sezione sono conseguenza dell'esperienza personale di tutte le parti coinvolte, dai Referenti orari al Referente dell'assegnazione delle risorse (in particolare quello dei laboratori), e sono frutto della consultazione dei manuali di istruzione e della visualizzazione delle schermate grafiche dei software descritti nel terzo capitolo.

6.2.1 Proposte suggerite da altri software

Il primo miglioramento potrebbe essere quello di rendere **obbligatoria** la compilazione della scheda online da parte di tutti i docenti, altrimenti se già la raccolta dati continua ad essere manuale come possiamo pensare di automatizzare l'intera pianificazione?

Per rendere più *stimolante* la compilazione, oltre alle migliorie indicate nel paragrafo precedente, occorre semplificarla con soli campi a scelta e con "help in linea" per guidare l'utente passo dopo passo.

Per esempio per quanto riguarda l'**incompatibilità tra materie**, l'**indisponibilità** e le **preferenze dei docenti**, alcuni dei software citati nel terzo capitolo propongono delle soluzioni interessanti e molto dettagliate.

Il software **EDT** [43], utilizzato per l'orario scolastico, presenta una schermata dove viene indicata la settimana come una griglia dove le righe sono le fasce orarie (di un'ora) e le colonne i giorni della settimana; il docente può scegliere tre opzioni a seconda dei colori: il rosso per l'*indisponibilità*, l'arancione per le *indisponibilità opzionali* e il verde per la *preferenza*, Figura 6.2.

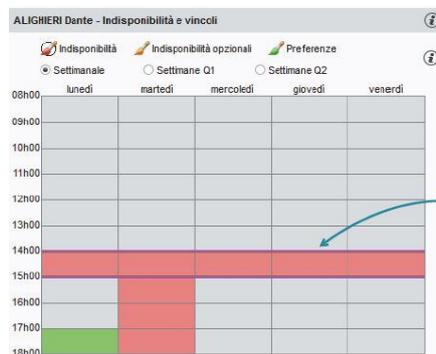


Figura 6.2. Vincoli e preferenze in EDT

A questa si aggiunge un'altra schermata, Figura 6.3, dove il docente indica tutti i dettagli dei vincoli e delle preferenze utili per l'ottimizzazione semplicemente selezionando le varie voci proposte:

- *massime ore nella giornata, al mattino, al pomeriggio;*
- *numero giorni e ora di inizio della lezione;*
- *numero giorni e ora di fine della lezione;*
- *numero massimo di mezze giornate di lavoro;*
- *numero ore buche tollerate*

The screenshot shows a configuration window for 'ALIGHIERI Dante - Vincoli e preferenze di ottimizzazione'. It includes several sections:

- Max ore:** Three dropdown menus for 'Giornata', 'Mattino', and 'Pomeriggio', all set to 'Niente'.
- Gestione orari:** Two dropdown menus for 'giorni alla settimana, iniziare non prima delle' (set to 2) and 'giorni alla settimana, finire non oltre le' (set to 2), with corresponding time pickers for '09h00' and '18h00'.
- Giorni e 1/2 giornate libere:** A dropdown for 'Assegna...' (set to 1), a dropdown for 'giornate libere' (set to 0), and a dropdown for 'mezze giornate libere' (set to 0).
- Massimo di mezze giornate di lavoro:** Two dropdown menus for 'Mattino' (set to 'Niente') and 'Pomeriggio' (set to 2).
- Preferenze di ottimizzazione:** A checkbox for 'Lavorare solo mezza giornata al giorno' (unchecked) and a dropdown for 'Numero di ore di buca tollerate' (set to 2).

Figura 6.3. Vincoli e preferenze di ottimizzazione in EDT

Nelle varie sezioni è presente il tasto "i" che guida l'utente durante la compilazione.

Anche il software **Hyperplanning** [47], per l'orario universitario, presenta una schermata simile, Figura 6.4, per l'indisponibilità e le preferenze:

The screenshot shows a grid interface for 'Prof. A SIMOV Isaac - Indisponibilità e preferenze'. On the left is a list of professors. The main area is a grid with columns for days (lunedì, martedì, mercoledì, giovedì, venerdì) and rows for time slots (09h00, 11h00, 12h00, 13h00, 14h00, 15h00, 16h00, 17h00, 18h00). The 'giovedì' column is highlighted in red, and a large black arrow points downwards through it. At the bottom, there is a 'Periodo attivo' bar and a 'Personalizza' button.

Figura 6.4. Indisponibilità e preferenze dei docenti in Hyperplanning

Per quanto riguarda *l'incompatibilità tra materie* il software propone una schermata dove viene selezionata la materia principale e un'altra finestra dove si possono selezionare e aggiungere le altre materie per fissare l'incompatibilità tra loro, Figura 6.5.

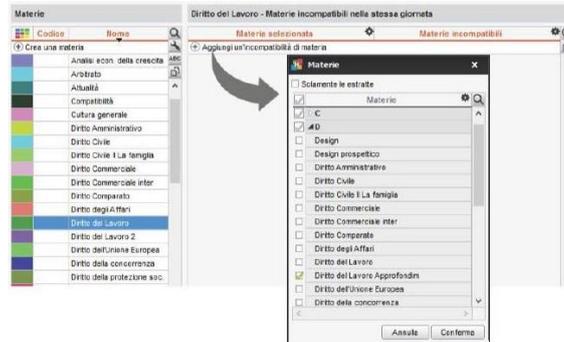


Figura 6.5. Incompatibilità tra materie in Hyperplanning

Il software **Untis** permette al docente di indicare la propria preferenza o indisponibilità non solo tramite un colore ma anche con una "*peso*" per indicare la priorità delle preferenze (valore positivo) e delle indisponibilità (valore negativo), come mostrato in Figura 6.6.

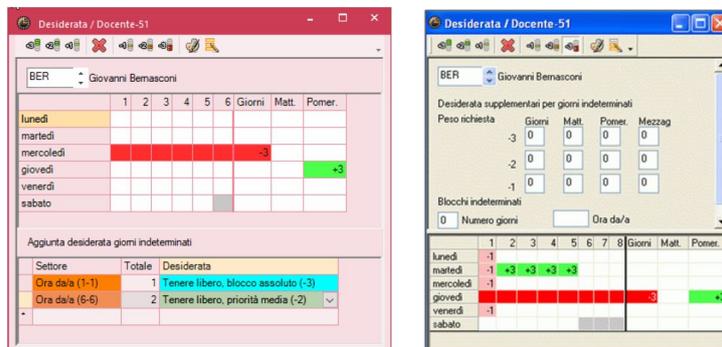


Figura 6.6. Desiderata del docente e Desiderata supplementare per i giorni non disponibili in Untis

Si può così ottenere un'ulteriore schermata che visualizza, in una griglia, i vari pesi di ogni docente (nelle righe) per ogni giorno (in colonna), come mostrato in Figura 6.7 a pag. 79.

Anche il software open source, **UniTime**, nella settimana tipo a griglia, Figura 6.8 a pag.79, il docente ha la possibilità di indicare facilmente le fasce orarie preferite o le indisponibilità colorando le celle, che corrispondono al giorno e alla fascia oraria; a seconda dei colori sappiamo se si tratta di una preferenza o di una indisponibilità .

Nome	lunedì		martedì		mercoledì		giovedì		venerdì		sabato	
	Matt	Pomer	Matt	Pomer	Matt	Pomer	Matt	Pomer	Matt	Pomer	Matt	Pomer
ALB												
BER					-3	-3			+3			
BIA			-3	-3					+3			
BRA	-2	-2										
CAT												
NER									-3	-3		
PER					+2	+2		-3	+2	+2	-2	-2
REZ												
ROS					-3	-3						
SOL											-3	-3

Figura 6.7. Desiderata giorni in Untis

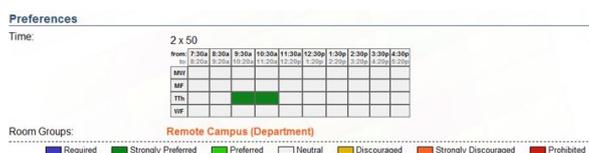


Figura 6.8. Richieste e indisponibilità in UniTime

6.2.2 Eventuali modifiche dell'applicativo web

Il software **Gestione Orari** [3] permette di comporre e pubblicare l'orario delle lezioni di tutti i corsi erogati presso il *Politecnico di Torino*, ed è uno strumento di consultazione per tutti gli utenti (docenti, studenti e personale tecnico amministrativo).

Il software si basa però su una quantità di dati forniti dall'*area Gestione Didattica*, che è l'unico servizio che può modificarli; con un ulteriore strumento esterno al software e con cadenza giornaliera un altro servizio preposto si occupa di allineare i dati registrati all'interno del software con i dati didattici.

Quindi una importante modifica consiste nel **centralizzare la base dati** e uniformare i vari servizi; come del resto avviene in tutti i software disponibili in commercio.

Nell'*area Gestione template orari* l'utente abilitato, tramite una serie di *Filtri* (descritti nel secondo capitolo), inserisce l'orario di tutti i corsi, caricando i blocchi legati alle *Materie* su un riquadro che corrisponde alla settimana tipo, Figura 2.5 a pag.16.

Una volta completato l'inserimento di tutti i blocchetti e corretti gli eventuali warning che si sono presentati, l'utente si trova di fronte all'orario di un determinato Corso di Laurea nella settimana tipo: dal lunedì al venerdì, dalle 8,30 alle 19,00.

Quindi questa visualizzazione è più utile al *Referente Orario* che al *Referente delle risorse*; in particolare quello dei laboratori, quando inserisce i vari blocchetti, non ha la percezione dell'occupazione dei laboratori; dovrebbe avere la possibilità di visualizzare l'occupazione dei laboratori nella settimana tipo; questa è possibile nell'apposita *Area Situazione Generale Aule*, Figura 6.9 di pagina 80, costituita da un calendario giornaliero che mostra tutti i blocchi di tutti i template orari per le diverse aule e laboratori dell'intero Ateneo (disposte a sinistra della schermata in ordine di *taglia*) nelle diverse fasce orarie della giornata (sono indicate tutte le mezze ore o a scelta le ore dalle 8,30 alle 19,00), con la possibilità di scegliere quale giorno visualizzare.

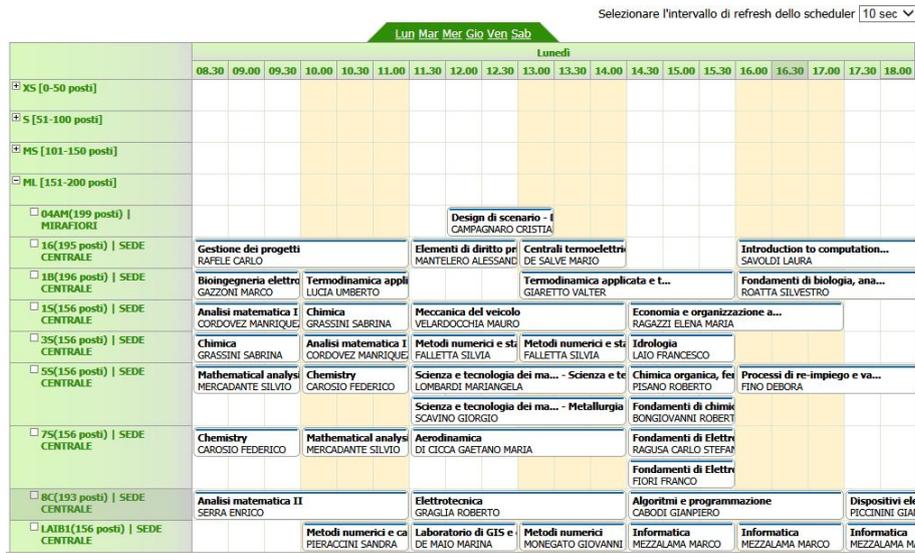


Figura 6.9. Occupazione aule e laboratori

In pratica questa è la visualizzazione al computer della *lavagna con le aule* utilizzata dal Referente dell'assegnazione delle aule. Ma purtroppo in questa schermata non è possibile fare alcuna modifica dei blocchi: né inserirli né spostarli da un'aula all'altra. Non si possono neanche selezionare una serie di aule o laboratori da visualizzare, si può solo scegliere di espandere o comprimere la visualizzazione a seconda della taglia delle aule/laboratori; quindi risulta essere poco funzionale anche per la visualizzazione. Solo nell'Area *Gestione calendario*, dopo aver pubblicato l'orario, si possono fare le modifiche selezionando l'aula o il laboratorio tramite dei Filtri; a sinistra della schermata compare l'elenco delle aule/laboratori in ordine di *taglia* (XS, S, M, ML, L) come mostrato in Figura 6.10.

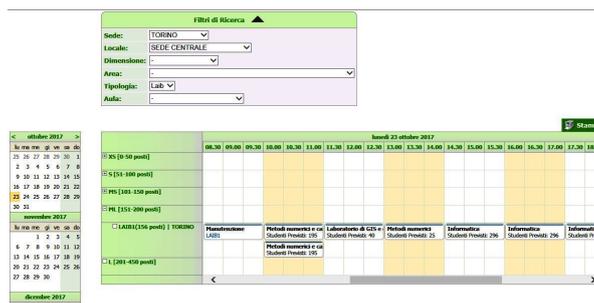


Figura 6.10. Gestione calendario

Ma anche in questo caso le modifiche possono essere fatte non direttamente in questa schermata; occorre selezionare la singola aula/laboratorio, per accedere ad un'ulteriore schermata che mostra l'occupazione dei corsi dell'aula selezionata per la settimana

selezionata e solo a questo punto si può procedere alla modifica, Figura 6.11.

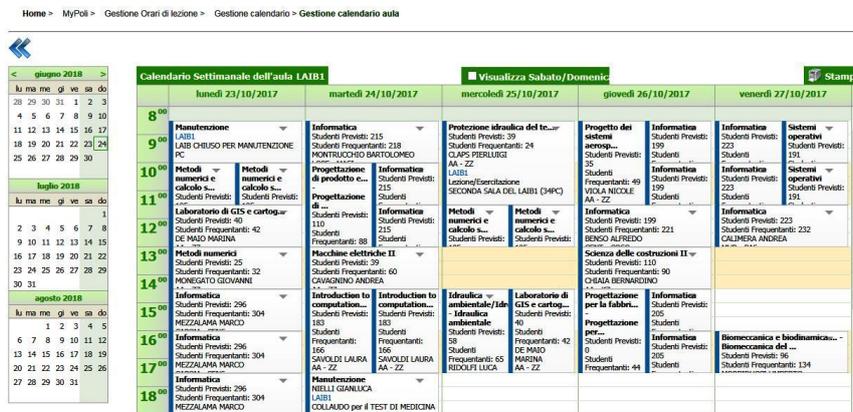


Figura 6.11. Occupazione singola risorsa

Sarebbe invece utilissimo avere all'interno di questa schermata i *Filtri* per ottenere i *blocchi Materia* da inserire nel calendario giornaliero in corrispondenza dell'aula scelta, ed avere la possibilità di spostarli da un'aula all'altra.

Con una composizione manuale dei vari Filtri potremmo avere una schermata come mostrato in Figura 6.12.

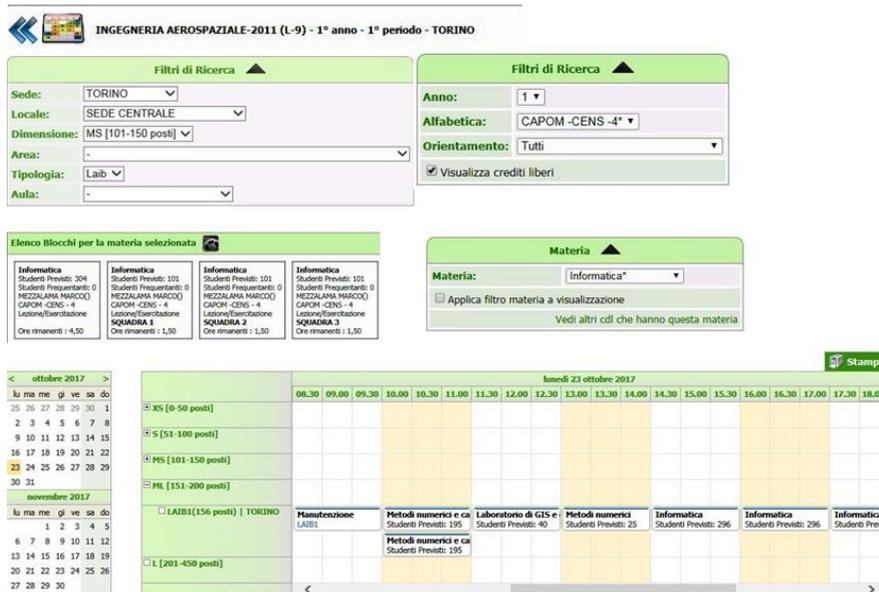


Figura 6.12. Schermata desiderata

Un esempio di questo tipo di schermata viene proposto dal software *Hyperplanning*, Figura 6.13 a pag.82, dove si possono selezionare da una lista le aule che visualizzeremo

come righe e nelle colonne vedremo le fasce orarie (visualizzazione giornaliera); e dove i blocchi delle materie sono colorati per poter essere facilmente modificati e spostati.

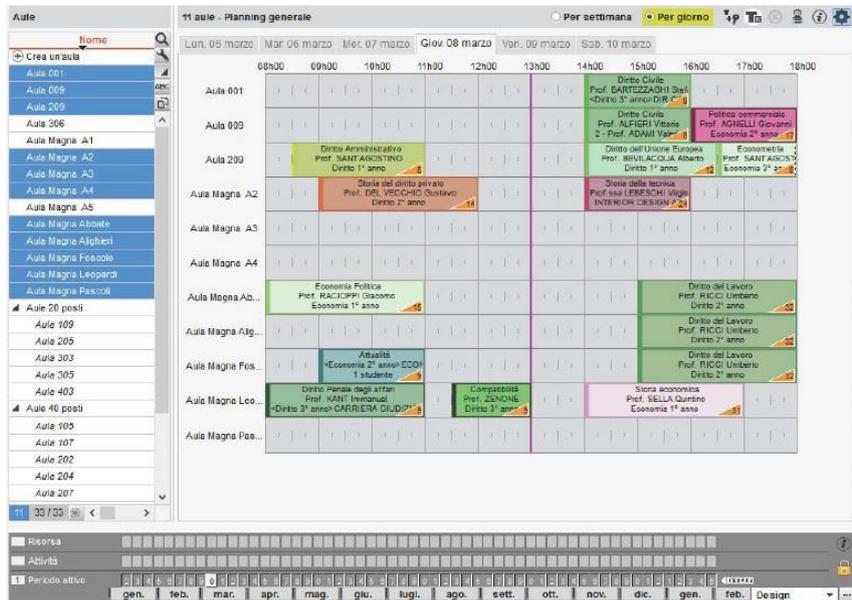


Figura 6.13. Spostamento materia da un'aula all'altra in Hyperplanning

Potrebbe essere utile che anche nel nostro template orario i blocchi fossero colorati, non per tutte le materie ma solo per Collegio al quale appartengono, e un colore particolare quando la materia corrisponde a più corsi di Laurea.

Un altro applicativo che propone queste differenze di visualizzazione: "template" per aula e i "colori" per i blocchi materie è l'**University Planner (UP2.0 [53])**, realizzato da *Cineca*, Figura 6.14.

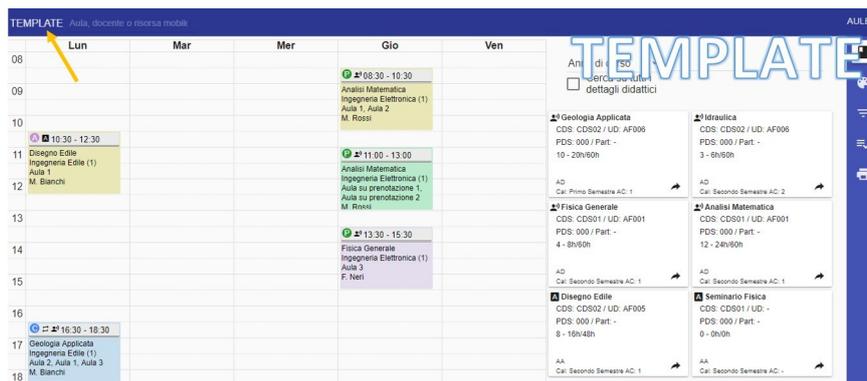


Figura 6.14. Inserimento nel template delle materie per aula in UP2.0

Capitolo 7

Conclusioni

Generare in modo automatico l'orario accademico è un problema di non facile ed immediata soluzione. Ma con questa tesi si è voluto dimostrare che esistono svariate soluzioni sia di tipo commerciale, tra i software già disponibili, sia di tipo open source, generate dalla comunità scientifica che da circa 60 anni lavora e propone valide soluzioni per il cosiddetto *timetabling problem*. Ma il problema principale è la definizione di specifiche regole e vincoli per la pianificazione dell'orario che sia valida per tutte le università, perché un programma scritto per creare l'orario di un'università difficilmente può essere utilizzato in un'altra università. Sarebbe utile avere un *problema standard* che racchiuda una serie di set di vincoli di un insieme di università in modo da scrivere dei programmi portatili e adattabili a tutti i casi.

La pianificazione dell'Orario accademico presso il *Politecnico di Torino* è un'operazione che, ancora oggi, viene svolta in modo manuale da parte del personale docente e del personale tecnico amministrativo, che dedicano molto tempo e molte energie per poter accontentare tutti gli studenti, i docenti e anche il personale di supporto; purtroppo però i risultati ottenuti spesso non rispondono pienamente ai desiderata richiesti.

Dal 2011 solo la pubblicazione dell'orario viene fatta in modo semiautomatico con l'inserimento dell'orario su un applicativo web che presenta pregi e difetti; mentre la generazione vera e propria dell'orario e l'assegnazione delle risorse viene fatta manualmente e in modo decentralizzato. Al problema della decentralizzazione della pianificazione dell'orario si aggiunge il grande problema della mancanza delle risorse adeguate per soddisfare tutti i corsi erogati presso il Politecnico di Torino.

Anche se una prima implementazione di orario automatizzato è stata fatta da un tesista [2] nel 2011 per la ex *IV* Facoltà, tuttavia in questi anni non si è deciso di estendere il risultato ottenuto per l'intero ateneo, poiché il *Collegio di Ingegneria Gestionale* presenta un caso semplificato rispetto agli altri Collegi. Quindi creare un orario per tutto l'Ateneo richiede uno studio per gradi e molto dettagliato.

Si è quindi pensato di cominciare con l'automatizzare l'assegnazione dei corsi (della laurea Triennale e Magistrale del primo periodo didattico) ai laboratori di informatica dell'*Area Information Technology* condivisi fra tutti i Collegi ed assegnati centralmente. Si è dimostrato che non è possibile assegnare i laboratori in base alle richieste definite dalla generazione degli orari, ma che si può solo massimizzare l'assegnazione dei corsi

ad ogni laboratorio. Successivamente utilizzando orari di "riserva" per i corsi che sono rimasti "fuori", si è riusciti ad assegnare tutti i corsi ai laboratori, con risultati ottenuti in pochissimo tempo, inferiore al secondo.

L'obiettivo futuro è estendere questa procedura automatizzata a tutte le risorse di Ateneo per "centralizzare" l'assegnazione di tutti i corsi erogati al Politecnico, consentendo alle persone coinvolte nella pianificazione dell'orario di recuperare del tempo da dedicare ad altre attività lavorative.

In merito al software *Gestione Orari* per la pubblicazione dell'orario si sono evidenziate le possibili migliorie per rendere più facile il suo utilizzo e per dare un ulteriore supporto all'assegnazione "manuale" delle risorse.

Anche se con questa tesi si è reso più visibile all'intero corpo docente la complessità del problema e il lavoro che c'è "dietro le quinte", l'obiettivo finale è quello di sopperire al fatto che ancora oggi al *Politecnico di Torino* non esista un processo per centralizzare la pianificazione dell'orario fornendo almeno un punto di partenza; visto che esistono diverse soluzioni bisogna solo decidere quale strada intraprendere: esterna, testando uno dei software open source o chiedendo ad una ditta di creare un programma ad hoc per la nostra realtà accademica; oppure interna, costruendo uno strumento potente per la generazione dell'orario di tutti i corsi offerti al Politecnico di Torino, anche sulla base dei risultati ottenuti con questa tesi.

Poiché la nuova squadra di governo, appena formata, ha intenzione di apportare nuovi cambiamenti nella pianificazione dell'orario, proprio sfruttando gli algoritmi risolutivi nell'ambito della Ricerca operativa, fino ad arrivare ad un vero e proprio automatismo dell'intero processo, auspichiamo che questa tesi possa essere presa come spunto per valutazioni e suggerimenti per questo percorso di cambiamento.

Capitolo 8

Appendici

8.1 Appendice A: listato del modello in linguaggio *Mosel*

Riportiamo il listato del modello completo della terza istanza considerando tutti i vincoli (*Assegnamento, Caso particolare del LAIB1, Capienza Laib, Corsi di Informatica del primo anno e corsi da tre moduli orari consecutivi e Corsi da due moduli orari consecutivi*) e il vincolo aggiuntivo *Software/attrezzatura disponibile solo in un laboratorio*.

8.2 Appendice B: risultato ottenuto

Riportiamo il risultato ottenuto dal modello riportato in Appendice A, che riporta:

- quanti e quali corsi vengono assegnati ai laboratori;
- quanti e quali corsi rimangono "fuori" dall'assegnazione;
- quali sono i laboratori e gli slot rimasti liberi;
- quanti e quali corsi rimasti fuori vengono assegnati nei laboratori e nelle fasce orarie libere.

Bibliografia

- [1] J. Aubin, J.A. Ferland, *A large scale timetabling problem*, Computers and Operational Research, 16(1):67-77, 1989.
- [2] R. Bargetto, F. Della Croce, *Ottimizzazione dell'orario universitario*, Tesi, Politecnico di Torino, 2011.
- [3] S. Cabras, *Gestione Orari - Manuale Utente*, Ufficio Sviluppo e gestione applicazioni, Area Information Technology, Politecnico di Torino, 2011.
- [4] S. Cabras, *Orari di lezioni e prenotazioni aule - Manuale Utente*, Ufficio Sviluppo e gestione applicazioni, Area Information Technology, Politecnico di Torino, 2011.
- [5] M.W. Carter, C.A. Tovey, *When is the classroom assignment problem hard?* Operations Research, 40(1S):28-39, 1989.
- [6] F. Della Croce, F. Salassa, *Modelli di programmazione lineare - Ricerca operativa*, slide delle lezioni.
- [7] S. Daskalaki, T. Birbas, E. Housos, *An integer programming formulation for a case study in university timetabling*, European Journal of Operational Research, Volume 153, Issue 1, Pages 117-135, 2004.
- [8] L. De Giovanni, L. Brentegani, *Ricerca Operativa, Modelli di Programmazione Lineare*, Appunti di lezione.
- [9] J.J. Dinkel, J. Mote, M.A. Venkataramanan, *An efficient decision support system for academic course scheduling*, Operations Research, 37(6):853-864, 1989.
- [10] F. Facchinei, C. Mannino, S. Lucidi, M. Roma, *Appunti delle lezioni di Ricerca Operativa*, Università di Roma, Sede di Latina (università Pontina), Corso di Laurea in Ingegneria Informatica, a.a. 2003-2004.
- [11] V. Gambale, M. Marzolla, tesi *Un software per la gestione dell'orario delle lezioni*, Alma Mater Studiorum - Università di Bologna, Scuola di Scienze, a.a. 2014-2015.
- [12] C.C. Gotlieb, *The construction of class-teacher timetables*. In C. M. Popplewell, editor IFIP congress 62, pages 73-77. North-Holland, 1963
- [13] S. Govetto, Articolo internet del 12 agosto 2016 sull'11a Internal Conference on the Practice and Theory of Automated Timetabling, <http://qui.uniud.it/notizieEventi/ateneo/informatica-conferenza-internazionale-alluniversita-di-udine>.
- [14] A. Hertz, *Tabu search for large scale timetabling problems*, European Journal of Operational Research, 54:39-47, 1991.
- [15] A. Hertz, *Finding a feasible course schedule using tabu search*, Discrete Applied Mathematics, 35(3):255-270, 1992.

-
- [16] G. Lach, M. Lach, J. Schick, E. Zorn, *University course timetabling with Moses: System demonstration. An IT system for academic resource planning at universities* Proceedings of the 11th International Conference on Practice and Theory of Automated Timetabling (PATAT-2016), Udine, Italy, August 23-26, 2016.
- [17] P. Morse, G. Kimball *Methods of operations research* Dower Publications, USA, 2003.
- [18] T. Müller, *Constraint-based Timetabling*, Ph.D. Thesis, Charles University in Prague Faculty of Mathematics and Physics, Prague 2005.
- [19] T. Müller, K. Murray, H. Rudová *Modeling and Solution of a Complex University Course Timetabling Problem*, PATAT 2007.
- [20] T. Müller, *University Course Timetabling - Solver Evolution*, Proceedings of the 11th International Conference on Practice and Theory of Automated Timetabling (PATAT-2016), Udine, Italy, August 23-26, 2016.
- [21] T. Müller, *UniTime: State of the Project*, Open Apereo 2018, Montreal, Quebec, 3rd-7th June
- [22] T. Müller, Z. Mullerova *Internationalization of UniTime*, Open Apereo 2018, Montreal, Quebec, 3rd-7th June.
- [23] J. Pearl, *Heuristics: Intelligent search strategies for computer problem solving*, Addison-Wesley, Reading MA, 1984.
- [24] Regolamento didattico di Ateneo per i corsi istituiti in applicazione del D.M. 270/04 (Regolamento sull'autonomia), Emanato con Decreto del Rettore n.491 del 4 ottobre 2016.
- [25] H. Rudová, T. Müller, K. Murray, *Complex university course timetabling*, Journal of Scheduling 14(2):187-207, 2011.
- [26] P. Serafini, *Ricerca Operativa*, Springer, 2009.
- [27] A. Schaerf, *A Survey of Automated Timetabling*, Artificial Intelligence Review 13: 87-127, 1999. Kluwer Academic Publishers. Printed in the Netherlands.
- [28] A. Schaerf, *Tabu search techniques for large high-school timetabling problems*, In Proc. of the 13th Nat Conf. on Artificial Intelligence (AAAI-96), pages 363-368, Portland, USA, 1996. AAAI Press/MIT Press.
- [29] A. Schaerf, L. Di Gaspero *Il problema della generazione automatica dell'orario delle lezioni: teoria e pratica* 2004.
- [30] G. Schmidt and T. Strohlein, *Timetable construction an annotated bibliography*, The Computer Journal, 23(4):307-316, 1979.
- [31] G. Solotorevsky, E. Gudes, A. Meisels, *RAPS: A rule-based language specifying resource allocation and timetabling problems*, IEEE Transactions on Knowledge and Data Engineering, 6(5):681-697, 1994.
- [32] R. Tadei, F. Della Croce, A. Grosso, *Fondamenti di Ottimizzazione*, Progetto Leonardo, Esculapio, Bologna, 2005.
- [33] A. Tripathy, *School timetabling - A case in large binary integer linear programming*, Management Science, 30(12):1473-1489, 1984.
- [34] A. Tripathy, *Computerised decision aid for timetabling - A case analysis*, Discrete Applied Mathematics, 35(3):313-323, 1992.
- [35] P. Toth, D. Vigo, *The Vehicle Routing Problem*, SIAM, 2001.

- [36] D. de Werra. *An introduction to timetabling*, European Journal of Operational Research, 19; pages 151-162, 1985.
- [37] BEST - Bullet Education Scheduling and Timetabling, , Copyright 2017 Bullet Solutions, Portogallo, USA <https://www.bulletsolutions.com/>
- [38] F. Speciale, <http://www.edscuola.it/archivio/software/dborario.htm>
- [39] O. Bettelli - Applicazioni Informatiche, Bologna, <http://www.istanze.unibo.it/oscar/ai08.htm>
- [40] G. Rizzo, FILEorario <http://www.grizzo.it/index.html>
- [41] software Orario Facile 10 - Mathema Software, Torino, <https://www.orariofacile.com/>
- [42] ORARIO WEB, <https://www.orarioweb.it/>
- [43] EDT l'orario scolastico, INDEX-EDUCATION, contatto@index-education.it, Chieri (TO), <https://www.index-education.com/it/>
- [44] FET - Free Timetabling Software L. Lalescu <https://lalescu.ro/liviu/fet/>
- [45] FICOTM Xpress Optimization Suite Fair Isaac Corporation Trademark <http://www.fico.com/en/products/fico-xpress-optimization>
- [46] home page of International Timetabling Competition <http://www.cs.qub.ac.uk/itc2007/> e <http://www.unitime.org/itc2007/>
- [47] HYPERPLANNING 2018, INDEX EDUCATION, contatto@index-education.it, Chieri (TO), <https://www.index-education.com/it/>
- [48] Mimosa Scheduling Software - Version 7.1.10 *Mimosa Scheduling Software Tutorial for beginners*, © Copyright 2018 Mimosa Software Ltd, FINLAND <http://www.mimosasoftware.com/>
- [49] Lantiv Scheduling Studio 8, © 2018 LANTIV, contact@lantiv.com, <http://scheduling-studio.lantiv.com/>
- [50] <http://www.timetableplus.com/>, TIMETABLE Plus, © 2014 Timetable Plus, Malaysia e Singapore
- [51] UniTime - University Timetabling, © 2018 UniTime, <http://www.unitime.org/>
- [52] UniUntis, Untis GmbH - Gruber e Petters, Stockerau, Austria www.untis.at/ [contattoitalia](mailto:contattoitalia@planscuola.it): PLANSCUOLA s.r.l. Renato Leonardi, <http://www.planscuola.it/>
- [53] University Planner 2.0 CINECA, Bologna 2018, <https://wiki.u-gov.it/confluence/display/public/UGOVHELP/University+Planner+2.0>, Technical Portal HOME
- [54] WinOR, *Software x la gestione e creazione di un Orario Scolastico in Excel*, 2007 <http://www.dinomolli.it/orario.asp>
- [55] ZonabitOrario, Zonabit Sistemi Srl, Bolzano-Roma, <https://www.zonabit.it>