

# POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Informatica



Tesi di Laurea Magistrale

## Nuovi metodi di modellazione e ottimizzazione per la pianificazione della didattica

**Supervisore:**

Prof. Cataldo Basile

Prof. Renato Ferrero

**Candidato:**

Manuel Messina

Anno Accademico 2022/2023  
Torino



# Sommario

La Tesi si pone l'obiettivo di sviluppare strumenti per il supporto alla pianificazione dell'orario dei collegi di ICM (Informatica, Cinema e Meccatronica) ed ETF (Elettronica, Telecomunicazioni e Fisica) del Politecnico di Torino, modellando tale attività come un problema di ottimizzazione. La tesi, partendo da un precedente lavoro in cui l'allocatione delle ore di lezione era stata modellata come un problema ILP (Integer Linear Programming) e risolta usando il solver IBM CPLEX, ha avuto come obiettivo il miglioramento e introduzione di nuove funzionalità al fine di produrre un orario che soddisfacesse maggiormente docenti e studenti. Le prime analisi sono state effettuate sul problema ancora aperto delle sovrapposizioni tra insegnamenti. È stato lo sviluppo di un software, esterno al programma di ottimizzazione, che estrae dati precisi su insegnamenti, corsi di laurea e orientamenti. Il punto forte di questo software è l'utilizzo di diverse fonti per la definizione dei vincoli. Le fonti coinvolte in questo processo sono le pagine web dei corsi di laurea con la descrizione degli orientamenti; le conoscenze dei responsabili dei corsi di laurea, su considerazioni di compatibilità tra insegnamenti e opportunità didattica, e le Informazioni sugli studenti iscritti agli insegnamenti nell'anno accademico 2022/2023, utilizzate per definire la compatibilità degli insegnamenti su base statistica. Dall'unione di queste fonti sono stati generati i valori finali di compatibilità tra insegnamenti, utilizzati per la generazione di vincoli più adeguati al problema preso in esame. Successivamente a questa prima fase di ottimizzazione, sono stati analizzati gli altri due aspetti principali che influiscono sulla pianificazione della didattica: i docenti e gli studenti. Le analisi hanno permesso di individuare informazioni di rilievo da aggiungere o modificare rispetto al modello precedente. Per quanto riguarda i docenti, è stata data la possibilità di esprimere, attraverso un questionario, diverse preferenze sull'organizzazione quotidiana e settimanale delle lezioni, sia in positivo che in negativo, utilizzate per la generazione di vincoli. Questi vincoli comprendono la preferenza di assegnazione (giorno e fascia oraria) ed il numero massimo di ore da tenere giornalmente e la pausa tra lezioni. Lato studenti, i vincoli considerati durante il lavoro originale sul numero massimo di ore consecutive e sulle pause tra essi sono rimasti invariati. Tuttavia, è stata sviluppata una soluzione completamente nuova, basata su "grafi", per poterli gestire in modo più preciso. Si è sviluppato un nuovo tool che, rappresentando i diversi orientamenti e le scelte possibili per gli studenti come percorsi all'interno di un grafo che include tutti gli insegnamenti, permette di stimare gli studenti coinvolti in sequenze di slot di lezione. Da questi percorsi e dagli studenti coinvolti è stato possibile generare hard e soft constraint bilanciando le penalità in funzione del peso.

Infine, sono stati raccolti e validati i nuovi dati relativi all'anno accademico 2023/2024 che sono stati usati per simulare la generazione dell'orario dei collegi ETF e ICM usando un nodo di calcolo fornito dall'area IT d'Ateneo. Concludono la Tesi alcuni spunti per una futura espansione del modello e di miglioramento del software oltre che ad una aggiornata versione dei manuali d'uso, includendo i tool indipendenti sviluppati a supporto dell'intero progetto.



# Indice

<b>Lista delle Figure</b>	<b>IX</b>
<b>1 Introduzione</b>	<b>1</b>
<b>2 Background</b>	<b>6</b>
2.1 Ricerca Operativa . . . . .	6
2.2 Programmazione lineare . . . . .	7
2.2.1 Definizione Matematica ILP . . . . .	7
2.3 Modello ILP per l’allocazione orario . . . . .	7
2.3.1 Variabili . . . . .	7
2.3.2 Vincoli . . . . .	8
2.3.3 Funzione obbiettivo . . . . .	8
2.3.4 Ulteriori nozioni . . . . .	8
2.4 Il Solver CPLEX . . . . .	9
2.4.1 Algoritmi Inclusi in CPLEX . . . . .	9
<b>3 Definizione del Problema</b>	<b>11</b>
3.1 Requisiti ereditati . . . . .	11
3.2 Requisiti non considerati . . . . .	13

3.3	Nuovi Requisiti . . . . .	14
3.4	Vincoli Ereditati . . . . .	14
3.4.1	Hard Constraint . . . . .	14
3.4.2	Soft Constraint . . . . .	15
3.5	Vincoli Ereditati e Modificati . . . . .	15
3.5.1	Vincoli su Insegnamenti di un Orientamento . . . . .	15
3.5.2	Vincoli su Studenti iscritti ad un Orientamento . . . . .	17
3.6	Vincoli Introdotti . . . . .	19
3.6.1	Hard Constraint . . . . .	19
3.6.2	Soft Constraint . . . . .	19
<b>4</b>	<b>Sovrapposizione Insegnamenti</b>	<b>20</b>
4.1	Modellazione . . . . .	20
4.1.1	Analisi dei dati . . . . .	21
4.1.2	Schema XSD . . . . .	22
4.2	Raccolta dei dati . . . . .	27
4.2.1	Definizione del Web Scraping . . . . .	27
4.2.2	Scraper . . . . .	27
4.2.3	Utilizzo dati Scraper . . . . .	28
4.3	Manuale d'uso . . . . .	33
4.3.1	lo script scraper.py . . . . .	33
4.3.2	extract_excel_from_xml.py . . . . .	33
4.3.3	generateCompleteExcel.py . . . . .	34
4.3.4	gen_file_intersezioni.py . . . . .	35
4.3.5	compute_final_corr.py . . . . .	35

<b>5</b>	<b>Funzionalità aggiuntive del modello</b>	<b>36</b>
5.1	Gestione Studenti . . . . .	36
5.1.1	Definizione Path . . . . .	36
5.1.2	Implementazione . . . . .	37
5.1.3	Manuale d'uso . . . . .	38
5.2	Implementazione dei nuovi Requisiti . . . . .	39
5.2.1	Modifiche allo Schema XML . . . . .	39
5.2.2	Vincoli Restrizioni Temporali . . . . .	41
5.2.3	Vincoli sedi diverse . . . . .	43
<b>6</b>	<b>Testing e Validazione</b>	<b>46</b>
6.1	Simulazioni su dati a.a 2022/2023 . . . . .	46
6.1.1	Prestazioni . . . . .	47
6.2	Simulazioni su dati a.a 2023/2024 . . . . .	47
6.3	Problema dell'insufficienza di Aule . . . . .	52
6.3.1	Soluzione . . . . .	53
6.4	Problematica Max Slot Docenti . . . . .	56
6.4.1	Soluzione . . . . .	57
<b>7</b>	<b>Conclusioni</b>	<b>58</b>
7.1	Valutazioni conclusive . . . . .	58
7.2	Sviluppi futuri . . . . .	59
7.2.1	Penalità Docenti . . . . .	59
7.2.2	Insegnamenti a Scelta e Fasce Orarie Protette . . . . .	60
7.2.3	Rifinire le funzioni per il calcolo delle penalità . . . . .	60
7.2.4	Aule . . . . .	60



7.2.5 Estensione API web . . . . .	60
<b>Bibliografia</b>	<b>61</b>

# Lista delle Figure

4.1	Screenshot pagina web CDL . . . . .	21
4.2	Esempio Tabelle Orientamento . . . . .	22
4.3	Esempio XML 1: Insegnamento e scelta obbligatoria lingua . . . . .	28
4.4	Esempio XML 2: scelta_tabella . . . . .	28
4.5	Esempio XML 3: scelta_tabella_lingua . . . . .	29
4.6	Esempio di output dello scraper in Excel. . . . .	30
4.7	Esempio di tabella correlazioni dopo la modifica dei Responsabili. . . . .	31
4.8	Risultato post esecuzione scraper.py . . . . .	34
5.1	Mappa sedi Politecnico . . . . .	44
6.1	Statistiche Aule Assegnate anno precedente . . . . .	51
6.2	Aule libere aggiunte al modello . . . . .	51



# Capitolo 1

## Introduzione

La Tesi si pone l'obiettivo di sviluppare strumenti per il supporto alla pianificazione dell'orario dei collegi di ICM (Informatica, Cinema e Meccatronica) ed ETF (Elettronica, Telecomunicazioni e Fisica) del Politecnico di Torino, modellando tale attività come un problema di ottimizzazione. La tesi è partita da un precedente lavoro, in cui l'allocazione delle ore di lezione era stata modellata come un problema ILP (Integer Linear Programming) e risolta usando il solver IBM CPLEX. Con questo lavoro sono state migliorate alcune delle funzionalità esistenti, tra le quali l'applicazione web come software di supporto visuale per l'eventuale modifica delle soluzioni generate e la riformulazione di alcuni dati in ingresso al software. Tra queste funzionalità si riporta ad esempio la gestione delle informazioni sulla disponibilità delle aule, che passano da valori generici per l'intera settimana a valori puntuali per giorno e fascia oraria su base di informazioni forniteci da responsabili delle aule.

E sono state introdotte nuove funzionalità. In particolare, è stata sviluppata una soluzione per modellare e gestire al meglio le sovrapposizioni di insegnamenti, un modello basato sui grafi per l'ottimizzazione delle lezioni consecutive e dei "buchi orario" per la didattica degli studenti, la gestione automatica delle pause per i docenti che devono svolgere didattica in sedi differenti, oltre a fornire numerosi parametri per una configurazione più precisa dell'orario (es. gli orari in cui i docenti preferiscono fare didattica, o le opzioni per restringere lo spazio di allocazione dei blocchi di didattica in funzione dei giorni/fascia oraria, la modellazione delle lezioni degli altri collegi e dei blocchi da allocare al sabato).

Inizialmente, è stato modellato formalmente uno dei problemi ancora aperti: la gestione delle sovrapposizioni tra insegnamenti appartenenti ad uno stesso orientamento. Essendo impossibile evitare sovrapposizioni, è stato introdotto il concetto di "correlazione tra coppie di insegnamenti". Tale correlazione è un valore compreso tra 0 e 100, dove da 0 a 99 si indica la possibilità di sovrapporre due insegnamenti con una penalità crescente (*soft constraint*), e 100 rappresenta due insegnamenti che non possono essere mai sovrapposti (es. insegnamenti obbligatori o scelti da un numero elevato di studenti), pena l'impossibilità

di generare soluzioni valide (*hard constraint*). Dati precisi sugli orientamenti sono stati ottenuti dalle pagine web mediante un software sviluppato appositamente (Scraper). Questo software elabora i dati e genera una matrice di insegnamenti, in cui le righe e le colonne rappresentano gli insegnamenti, mentre l'intersezione di queste rappresenta il valore di correlazione tra le coppie. Grazie al software, è possibile determinare a priori le tre tipologie di coppie di insegnamenti:

- Insegnamenti assolutamente non parallelizzabili (correlazione = 100).
- Insegnamenti parzialmente parallelizzabili (correlazione compresa tra 1 e 99), ovvero insegnamenti che potrebbero essere seguiti nello stesso periodo didattico in un determinato corso di laurea e orientamento.
- Insegnamenti assolutamente parallelizzabili (correlazione = 0), ovvero insegnamenti che in nessun corso di laurea e orientamento potranno far parte dello stesso periodo didattico.

I primi dati così generati sono stati poi sottoposti ai responsabili dei Corsi di Laurea, i quali hanno dato un contributo alla generazione dei valori finali, indicando i valori di correlazione per le coppie di insegnamenti parzialmente parallelizzabili, sulla base di considerazioni di compatibilità e opportunità didattica. Successivamente, questi dati iniziali sono stati aggiornati, tramite tool automatico, considerando quanti studenti in comune gli insegnamenti avevano nell'anno accademico precedente. Attraverso l'utilizzo del software, è stato possibile aggiornare dinamicamente il valore delle correlazioni. In particolare, il valore di correlazione è stato aumentato per le coppie di insegnamenti di un fattore influenzato dal numero di studenti in comune. Questo aggiornamento è stato effettuato utilizzando una formula appositamente sviluppata dopo diverse simulazioni basate sull'allocazione dell'anno accademico 2022/23.

In seguito, sono stati introdotti nuovi vincoli per soddisfare alcune delle esigenze dei docenti. Uno dei vincoli riguarda la possibilità per i docenti di esprimere una preferenza sugli orari e i giorni delle lezioni, applicabile in maniera puntuale ad ogni slot o ad ogni insegnamento. Questo tipo di vincolo può essere espresso sia come hard che soft constraint. Un'altra esigenza riguardava la situazione in cui alcuni docenti potrebbero dover tenere lezioni nello stesso giorno ma in sedi diverse, richiedendo quindi un certo intervallo di tempo, tra una lezione e l'altra, sufficientemente ampio da compensare la distanza fisica. Questo vincolo viene gestito come hard constraint, garantendo al docente almeno uno o due slot liberi, a seconda delle sedi considerate.

Per acquisire le informazioni delle preferenze dei docenti relative a questi ultimi miglioramenti, è stato somministrato un questionario ai docenti, similmente all'anno precedente, i cui dati raccolti sono stati utilizzati sia per definire i dati in input al software, sia per la generazione dei vincoli.

I vincoli generati includono:

- (*Hard constraint*) indisponibilità dei docenti per determinati slot (giorno-fascia oraria), in cui gli slot con quel docente non possono essere assegnati;
- (*Soft constraint*) espressione di gradimento o non gradimento nel tenere lezioni in certi slot (giorno/fascia oraria), in cui il rispetto o la violazione di tali vincoli comporta penalità o bonus nella funzione obiettivo;
- (*Hard constraint*) limite massimo del numero di ore di lezione da poter tenere in un singolo giorno;
- (*Soft constraint*) sulla presenza e/o durata delle pause tra le lezioni.

Spostando poi l'attenzione sui requisiti necessari a rendere migliore l'orario anche per gli studenti, è stata sviluppata una nuova soluzione. Partendo dall'analisi condotta nel lavoro precedente, è stato facile individuare le necessità degli studenti e definire i vincoli da generare:

- (*hard constraint*) massimo numero di ore di lezione al giorno: questo vincolo impone un limite sul numero massimo di ore di lezione che uno studente può avere in un singolo giorno;
- (*hard constraint*) massimo numero di ore di lezione consecutive (*hard constraint*): questo vincolo riguarda il numero massimo di ore di lezione consecutive che uno studente può sostenere senza interruzioni. Allo stesso tempo vengono generati altri vincoli (*soft constraint*) in cui si aggiungono penalità quando si raggiunge o ci si avvicina a tale limite.
- (*soft constraint*) penalità per la presenza di buchi nell'orario: questo vincolo penalizza la presenza di lacune o periodi di tempo dispersi nell'orario di uno studente, incoraggiando la creazione di un orario più coerente e organizzato.

Nella soluzione precedente, determinare correttamente se i vincoli fossero rispettati o meno risultava più complesso del previsto: si sarebbe infatti dovuta individuare una soluzione che non si limitasse a considerare soltanto gli insegnamenti obbligatori, ma che introducesse tutte le opportunità derivanti dagli insegnamenti a scelta. Poiché, infatti, gli studenti possono scegliere diverse combinazioni di insegnamenti per lo stesso periodo didattico, è necessario garantire che i vincoli siano validi per tutte le possibili combinazioni di insegnamenti per tale periodo. Per affrontare questa sfida, sono state considerate le informazioni sugli studenti iscritti ai vari insegnamenti nell'anno accademico 22/23, già estratte in precedenza per la definizione delle correlazioni. Questi dati, insieme alle informazioni ottenute attraverso la modellazione formale di tutti gli orientamenti e le scelte che permettono agli studenti, vengono impiegati per generare dei percorsi di insegnamenti chiamati *'path'*. Ogni path rappresenta una combinazione possibile di insegnamenti da seguire in un determinato periodo didattico (che quindi almeno uno studente potrebbe scegliere) e viene associato a un valore di intersezione, un peso, che indica quanti studenti condividono tutti gli insegnamenti all'interno del path.

In questo modo, gli hard constraint che definiscono i requisiti degli studenti sono stati definiti sui path, assicurando che il numero massimo di ore di lezione sia rispettato per ogni combinazione di insegnamenti. I soft constraint generano invece una penalità in funzione del peso assegnato al path considerato. Questo approccio, tenendo conto delle diverse combinazioni di insegnamenti, garantisce che i vincoli siano generati per tutti gli studenti.

Infine, è stato sviluppato uno strumento in grado di supportare i referenti orario al momento di effettuare spostamenti puntuali delle lezioni. Infatti, nonostante i questionari, alcuni docenti aggiungono dei requisiti o riportano tardivamente le loro necessità. Inoltre, spostamenti sono necessari su richiesta della commissione spazi di ateneo dovuti a carenza di aule. Sulla base delle dipendenze degli orientamenti e di un'allocazione completa, questo strumento visualizza la situazione settimanale, indicando per ogni slot gli altri insegnamenti con cui andrebbe a sovrapporsi. Sulla base delle matrici di correlazione tra coppie di insegnamenti, viene calcolato un indice di compatibilità, che misura quanto lo spostamento potrebbe creare problemi agli studenti. Inoltre, tale strumento esclude in automatico gli slot in cui i docenti del corso da spostare hanno altre lezioni.

Durante il percorso di questa tesi, le nuove funzioni sono state validate attraverso diverse simulazioni e test, utilizzando i dati dell'anno accademico 2022/2023 come esempio, in assenza dei dati del nuovo anno accademico. Così, una volta aver ricevuto i dati ufficiali, è stata effettuata una validazione iniziale di essi e sono stati effettuati i primi tentativi di ottenere una soluzione valida utilizzando un nodo di calcolo fornito dalla Direzione Informatica di ateneo (ISIAD).

La tesi, in conclusione, suggerisce spunti di riflessione per miglioramenti futuri. Tra queste, si suggerisce la possibilità di condurre uno studio matematico sulle funzioni utilizzate per determinare i valori di penalità. Ad esempio, sarebbe interessante riesaminare il livello di penalità associato al valore di correlazione tra insegnamenti o, ancora, ridefinire i livelli di penalità sulla funzione del numero di iscritti, facendo uso dei risultati ottenuti con questa tesi. Insieme all'utilizzo dei risultati di questa tesi, questo studio potrebbe fornire una base teorica solida per creare vincoli di qualità superiore, che conducano a risultati sempre migliori.

Un altro problema irrisolto che merita menzione riguarda l'ottimizzazione dell'assegnazione delle aule con lo scopo di ottimizzare gli spostamenti, quest'ultimo rappresenta una sfida complessa, sia per il modello utilizzato, il quale non si adatta bene al problema, sia per i risultati ottenuti che hanno dimostrato quanto il numero delle aule sia già un fattore critico senza ulteriori vincoli.

Infine, è stato provveduto all'aggiornamento dei manuali di utilizzo precedentemente esistenti, introducendo anche manuali per le nuove funzionalità sia interne che esterne all'ottimizzatore.

Il resto di questa tesi è così organizzato.

- Il Capitolo 2 tratta i concetti fondamentali della programmazione lineare utile per una comprensione efficace dei Capitoli seguenti.
- Il Capitolo 3 descrive formalmente i requisiti trattati e come questi siano stati utilizzati per la generazione dei vincoli. La trattazione comprende oltre che le novità introdotte anche considerazioni effettuate sul modello passato[1].
- Il Capitolo 4 affronta il primo dei requisiti trattati, la Sovrapposizione degli Insegnamenti. Viene prima descritta il problema e poi la soluzione, fornendo anche informazioni utili all'applicazione di tale soluzione.
- Il Capitolo 5 Riporta le informazioni circa le altre funzionalità introdotte. Descrivendo, similmente al precedente capitolo, problemi e soluzioni circa i requisiti non ancora trattati. Quali la gestione degli studenti, i vincoli temporali per gli Slot e gli Insegnamenti e infine per le Sedi Lontane. Si conclude con la nuova funzionalità di visuale grafica, sviluppata per il supporto ai Responsabili di Orario.
- Il Capitolo 6 vengono qui riportate informazioni utili sull'approccio adottato per la validazione di dati e funzioni. Descrivendo accuratamente gli eventi accaduti nel processo e come questi sono stati risolti per arrivare alla soluzione finale.
- Il Capitolo 7 in una prima parte fornisce ulteriori considerazioni riguardo i risultati ottenuti dalle Simulazioni, proponendo poi degli sviluppi futuri per la tesi.



# Capitolo 2

## Background

Si introducono in questo capitolo i concetti fondamentali alla comprensione dei capitoli successivi.

### 2.1 Ricerca Operativa

La Ricerca Operativa è la “disciplina in cui si applicano metodi analitici avanzati per aiutare a prendere decisioni migliori”. Sono varie le tipologie di sistemi e situazioni che possono essere oggetto di studio nella Ricerca Operativa, tra cui sicuramente ricade la gestione e allocazione delle risorse per la generazione di un orario accademico.

I sistemi studiati[2] in Ricerca Operativa hanno in comune la possibilità di essere modellati in modo matematico a partire da una descrizione il più possibile quantitativa. Nei modelli della Ricerca Operativa è presente anche la possibilità di intervenire sul comportamento con una scelta opportuna di alcuni parametri. In Ricerca Operativa si vuole trovare una soluzione, e nel caso ci fossero tante possibili soluzioni, scegliere quella che sia più soddisfacente, mediante l'utilizzo di algoritmi. L'algoritmo scelto, deve però essere in grado di trovare tale soluzione in un tempo ragionevole, per cui è necessario analizzare anticipatamente la complessità del problema o dell'algoritmo stesso. Quando si parla di complessità in questo ambito, non ci si riferisce allo sforzo mentale compiuto nel risolvere il problema, bensì alla quantità di tempo necessaria all'algoritmo per trovare soluzioni, in quanto, per molti algoritmi, il tempo cresce esponenzialmente con la grandezza del problema. La teoria della complessità computazionale fornisce un indispensabile strumento concettuale per affrontare questa analisi.

## 2.2 Programmazione lineare

Si definiscono come problemi di Programmazione lineare (LP) tutti quei problemi di ottimizzazione in cui la funzione obiettivo è lineare e i vincoli sono tutti espressi da disequazioni lineari ed anche, eventualmente, equazioni lineari [2]. Il caso particolare in cui tutte le variabili sono intere è definito come Programmazione Lineare Intera (ILP, *Integer Linear Programming*), ed è proprio quella che verrà affrontata in quanto applicata per la risoluzione del problema.

### 2.2.1 Definizione Matematica ILP

Un problema di ottimizzazione lineare ha come scopo trovare la soluzione che minimizza la funzione obiettivo  $f$  tra tutte le soluzioni  $X$  che soddisfano l'insieme di  $m$  vincoli  $V$  per le  $n$  variabili intere del problema [3]:

$$\begin{aligned} & \min(\max)f(x) \\ v_i(x) & \leq, \geq o = b_i, \forall i = 1..m \\ & x \in \mathbb{N}^n \end{aligned} \tag{2.1}$$

dove:

- $x = (x_1, x_2, \dots, x_n)$  è il vettore di  $n$  variabili intere che rappresenta una possibile soluzione del problema
- $f(x)$  e tutte le  $v_i(x)$  sono definite come  $\mathbb{N}^n \rightarrow \mathbb{N}$
- $0 \leq i < m$ ,  $b_i \in \mathbb{N}$ .

## 2.3 Modello ILP per l'allocazione orario

Vengono qui riportati alcuni concetti fondamentali del modello ILP adottato nel contesto della pianificazione della didattica.

### 2.3.1 Variabili

All'interno del modello sono identificabili due tipi di variabili:

- Variabili indipendenti

Nel caso specifico, le uniche variabili indipendenti del modello riguardano la precisa allocazione di un dato slot. Dato uno slot  $s$  verranno associate ad esso due variabile, una per il giorno e l'altra per l'ora assegnata a tale slot. Da queste due variabili dipendono tante altre variabili dette dipendenti.

- Variabili dipendenti Le variabili dipendenti in questo modello saranno alla base del calcolo della funzione obiettivo. A tali variabili viene assegnato un valore per mezzo di equazioni (vincoli) dipendenti dalle prime variabili menzionate. In particolare il non soddisfacimento di certi vincoli porta alla generazione di penalità attraverso assegnazione di un certo valore a queste variabili.

### 2.3.2 Vincoli

Nel modello di un problema ILP i vincoli sono equazioni o disequazioni che vanno sempre rispettati. Nella realtà possono però distinguersi due tipi di vincoli: *rigidi* e *flessibili*; Tali vincoli devono, poi, trovare una rappresentazione matematica all'interno del modello. Formalmente diremo che se un vincolo rigido non è rispettato per un certo vettore di variabili, tale vettore non è soluzione del nostro problema. Se invece un vincolo flessibile non viene soddisfatto il risultato che avremmo sarebbe una variazione nella funzione obiettivo, non escludendo il vettore dalle possibili soluzioni.

### 2.3.3 Funzione obiettivo

La funzione obiettivo è il parametro per cui vettori di variabili possono essere confrontati per determinare, nel caso entrambi non violassero alcun vincolo, quale risulti in una soluzione migliore del modello. Nel particolare caso della pianificazione, la funzione obiettivo è una funzione di cui si cerca il minimo, il cui argomento, sostanzialmente, è la somma delle penalità derivanti dal non soddisfacimento dei vincoli flessibili.

### 2.3.4 Ulteriori nozioni

Una *soluzione ammissibile* di un problema di programmazione lineare è un vettore  $x_{@}$  che soddisfa tutti i vincoli rigidi. Una *soluzione ottima* di un problema di programmazione lineare è un vettore  $x_{*}$  che minimizza o massimizza la funzione obiettivo  $f$ . L'insieme  $X$  di tutte le soluzioni ammissibili è detto *regione ammissibile*. La risoluzione di un problema di programmazione lineare può ricadere in 3 casi:

- problema *innammissibile*: l'insieme  $X$  delle soluzioni ammissibili è vuoto
- problema *illimitato*: è possibile trovare soluzioni ammissibili che migliorano la funzione obiettivo all'infinito

Problem Set	CPLEX	CLP	GLPK	lp_solve	MINOS
Small CCO	0.0	0.1	1.3	19.0	3.1
Infeasible	0.21	3.6	0.7	43.8	16.3
Netlib	9.1	29.5	52.5	14,975.1	3,198.7
Kennington	12.9	16.1	624.3	19,417.5	10,123.8
Large CCO	13.0	19.0	108.4	3,175.8	41,976.1
FOME	54.5	182.7	6,061.4	33,544.5	59,301.9
Rail	152.5	212.9	N/A2	29,012.2	28,899.9
PDS	179.6	224.5	34,118.3	115,200.0	115,200.0
Grand Total	421.8	688.4	40,966.9	215,387.9	258,719.8

**Table 2.1:** Statistiche del solver (problemi *easy*)

- problema che *ammette soluzione ottima*: esiste almeno una soluzione  $x_*$  che minimizza o massimizza la funzione obiettivo; ne segue che il valore  $k$  definito come  $f(x_*) = k$ , appartiene ai numeri naturali.

## 2.4 Il Solver CPLEX

Per la risoluzione del modello, sin dalla tesi precedente, è stato scelto l'utilizzo del solver commerciale CPLEX, software pubblicato dall'azienda IBM, disponibile liberamente per iniziative accademiche.

CPLEX è uno strumento affermato per la sua completezza e prestazioni all'interno dell'ambito dei risolti LP. Uno studio è stato fatto proprio per comparare la qualità e performance di diversi solver LP open-source, sia gratuiti che a basso costo proprio con IBM CPLEX[4]. I risultati ottenuti da questi esperimenti hanno dimostrato che, nonostante l'attenzione posta nella selezione dei solver da sottoporre all'esperimento, di conseguenza scegliendo quelli che avevano più probabilità di competere con CPLEX, CPLEX si è rivelato essere il più performante. Nell'esperimento i solver sono stati sottoposti alla risoluzioni di 201 problemi LP da risolvere, di cui 180 definiti "easy" e 21 "hard", nella stragrande dei problemi CPLEX si è rivelato quello maggiormente performante. Si riportano in Tabella 2.1 e 2.2 le statistiche estrapolate dal report. Per maggiori dettagli sull'esperimento circa i risultati e i problemi analizzati si faccia riferimento a [4].

### 2.4.1 Algoritmi Inclusi in CPLEX

Il software di CPLEX al suo interno, per la risoluzione di problemi LP, possiede la possibilità di applicare 3 diversi algoritmi, dai quali, se non specificato, viene scelto il più adatto dal software stesso, sulla base dell'analisi del modello che viene inserito [4]. Tali algoritmi sono

Problem	Rows	Columns	Best CLP Solution(Seconds)	CPLEX Solution Time (Seconds)
neos2	132.568	1.56	1.7	7.0
NEOS1	131.581	1.892	2.7	9.2
SGPF	246.077	308.634	5.5	1.4
WAT	201.155	383.927	30.5	8.1
WATSON_2	352.013	671.861	48.7	33.3
bpmpd	479.119	36.786	58.5	12.3
cont4	160.792	40.398	562.2	261.7
Prob_2	528.185	1,259,121	616.0	224.2
nug08-3rd	19.728	20.448	844.4	657.1
cont1	160.792	40.398	1,147.0	265.9
ns168703	50.622	43.749	2,444.8	18,193.7
neos3	512.209	6.624	2,976.0	171.6
cont11	160.792	80.396	2,991.2	6,900.2
Linf_five20c	93.326	69.004	4,146.3	793.6
ns168892	32.768	16.587	28,800.0	28.1
nug20	15.24	72.6	28,800.0	451.7
w100cc00r004p001o004	10.203	321.696	28,800.0	10,795.9
cont1_1	1,918,399	641.598	28,800.0	28,800.0
nug30	52.26	379.35	28,800.0	28,800.0
L1_d10-40x3	80.476	420.366	28,800.0	28,800.0
cont11_1	1,468,599	981.396	28,800.0	28,800.0

**Table 2.2:** Statistiche del solver (problemi *hard*)

indicati nella guida, e consistono in:

- *Simplex Optimizer*
- *Dual-simplex Optimizer*
- *Barrier Optimizer*

Si rimanda alla guida per ulteriori informazioni su queste euristiche i cui dettagli non sono essenziali alla comprensione di questa tesi [5].

## Capitolo 3

# Definizione del Problema

Il problema della generazione dell'orario accademico può essere risolto modellando l'attività come ottimizzazione di un problema di programmazione lineare intera. Le equazioni del modello, cioè i vincoli, vengono stabiliti a seguito di un'analisi dei requisiti del problema. In questo capitolo vengono riportati i requisiti considerati durante il lavoro di tesi. Buona parte dei quali sono stati ereditati dalla tesi precedente[1], e sono riportati qui per avere l'intero modello disponibile in un'unico documento e come base per estendere ulteriormente il modello con l'obiettivo di rafforzarlo.

### 3.1 Requisiti ereditati

1. “Vi sono Corsi di Laurea, Orientamenti, Insegnamenti, Slot<sup>1</sup>, Fasce Orarie, Giorni, Locali, Docenti”.
2. “I Corsi di Laurea sono di tipo Magistrale o Triennale”.
3. “Un Corso di Laurea prevede diversi Orientamenti<sup>2</sup>”.
4. “Un Orientamento propone diversi Insegnamenti<sup>3</sup>”.
5. “Un Orientamento è costituito da più anni: tre per gli Orientamenti di Corsi di Laurea Triennali e due per quelli Magistrali”.

---

<sup>1</sup>Si intende la singola lezione indivisibile erogabile e può occupare 1.5 ore o un suo multiplo.

<sup>2</sup>Per ogni Corso di Laurea vi sono più Orientamenti.

<sup>3</sup>Un Insegnamento può essere presente in Orientamenti differenti.

6. “Un Orientamento ha Insegnamenti Obbligatori<sup>4</sup>, Insegnamenti parzialmente Obbligatori<sup>5</sup>, Insegnamenti a Scelta<sup>6</sup>”.
7. “In un Orientamento posso essere generati diversi percorsi.”<sup>7</sup>”
8. “Un Insegnamento si tiene in uno o più semestri ed è associato ad un anno per ogni Orientamento”.
9. “Un Insegnamento è composto da uno o più Slot”.
10. “Per ogni Insegnamento viene definito un Template Orario<sup>8</sup>”.
11. “Un Insegnamento è associato a uno o più Docenti e uno di essi è titolare di esso<sup>9</sup>”.
12. “Un Insegnamento è frequentato da un certo numero di studenti”.
13. “Un Insegnamento ha un piano di allocazione settimanale fisso per l’intera durata del semestre”.
14. “Un Insegnamento fa riferimento ad un Collegio”.
15. “Uno Slot può essere una *Lezione*, un’*Esercitazione in Aula* o un’*Esercitazione in Laboratorio*”.
16. “I Locali possono essere *Aule* e *Laboratori*. Le Aule si suddividono a loro volta in *Aule attrezzate CA2*, *Aule attrezzate CA*, *Aule TableBox*, *Aule WallBox* e *Aula 5T*; mentre i laboratori sono: *LABINF*, *LAIB*, *LADISPE*, *LED*, *LED1*, *LED2*, *ACSLAB*”.
17. “Gli Slot vanno allocati in un tipo di Locale specifico”.
18. “Un Insegnamento può prevedere Slot in cui gli studenti sono divisi in squadre; quindi potranno essere allocati anche in parallelo”.
19. “Ad ogni Slot sono associati uno o più Docenti”.

---

<sup>4</sup>Tutti gli studenti iscritti all’orientamento saranno anche iscritti all’insegnamento

<sup>5</sup>Sono tutti quegli insegnamenti che si trovano in decisione binaria. Molto spesso si tratta dalla scelta della lingua per un insegnamento equivalente, o di insegnamenti equamente molto affini all’orientamento. È quindi elevato il numero di studenti iscritti all’insegnamento per un determinato orientamento

<sup>6</sup>Sono un’insieme di insegnamenti, tipicamente appartenenti ad altri orientamenti, che servono al raggiungimento dei crediti necessari. Questi insegnamenti possono essere più o meno rilevanti per l’Orientamento in questione. Alcuni di questi vengono segnalati come "suggeriti" dal cdl.

<sup>7</sup>Un percorso consiste nell’insieme di Insegnamenti scelti da uno studente durante la compilazione del piano carriera. Rilevanti le informazioni sui percorsi per stabilire vincoli a favore degli studenti.

<sup>8</sup>Rappresenta la suddivisione desiderata dei vari Slot all’interno della settimana, specificando per ognuno di essi le risorse necessarie per la sua allocazione.

<sup>9</sup>Il Docente titolare non deve necessariamente essere presente in tutti gli Slot dell’Insegnamento; la sua figura è importante in quanto si occupa di definire il Template Orario per l’Insegnamento.

20. “Ogni Slot si svolge in un giorno della settimana: dal Lunedì al Venerdì ed eventualmente il Sabato”.
21. “Ogni Slot dura 1,5 ore e si svolge in una fascia oraria: dalle 8:30 alle 19:00”.
22. “Le Aule hanno una *capienza* massima”. Fatto strettamente in relazione con il numero di studenti frequentanti un Insegnamento.
23. “Uno Slot è assegnato ad un *Locale*”.
24. “I Docenti assegnano *preferenze* circa gli slot orari in cui desiderano a fare lezione.<sup>10</sup>”.
25. “I Docenti possono indicare un numero limitato di slot orari in cui sono impossibilitati a tenere una lezione<sup>11</sup>”.
26. “I Docenti potrebbero avere ulteriori incompatibilità dovute a Insegnamenti erogati all’interno di altri Collegi”.
27. “Per ogni Insegnamento viene definito un *Piano di Allocazione settimanale*”.
28. “Gli studenti scelgono un Orientamento e un piano di studi<sup>12</sup>”.
29. Due Insegnamenti possono essere:
  - Totalmente Sovrapponibili. “Conflitto vincolante”
  - Parzialmente “Conflitto non vincolante”
  - Totalmente NON Sovrapponibili “Nessun conflitto”
30. “Un Piano di Allocazione corrisponde all’orario settimanale di tutti gli Insegnamenti contenente tutti gli Slot che li costituiscono con le relative informazioni circa il Giorno, la Fascia oraria, i Docenti e il Locale assegnati per ognuno di essi”.
31. “Esistono dei vincoli tra Slot appartenenti allo stesso Insegnamento”.

## 3.2 Requisiti non considerati

1. I Docenti possono fornire informazioni in merito al fatto che le loro lezioni all’interno di un Insegnamento sono limitate in un periodo specifico del semestre.
2. Esiste il piano di allocazione dell’anno precedente.

---

<sup>10</sup>Informazioni non vincolanti al fine di trovare una soluzione valida.

<sup>11</sup>Informazioni vincolanti.

<sup>12</sup>Selezionando determinati Insegnamenti tra quelli non obbligatori viene generato il piano di studi, ovvero un’istanza di percorso



3. I Docenti possono esprimere una valutazione di gradimento dell'allocazione dell'anno precedente.
4. I Docenti titolari di un Insegnamento definiscono l'allocazione ideale degli Slot durante la settimana.

### 3.3 Nuovi Requisiti

1. "Insegnamenti o Slot potrebbero voler essere allocati in un sottoinsieme di Giorni e Fasce Orarie<sup>13</sup>."
2. "I Docenti potrebbero dover tenere lezioni per Insegnamenti diversi nello stesso giorno. Eventualmente anche in Sedi differenti<sup>14</sup>."

### 3.4 Vincoli Ereditati

#### 3.4.1 Hard Constraint

Vengono elencati di seguito i vincoli rigidi ereditati, senza modifiche, dalla Tesi precedente.

- Slot allocabili solo in prefissati slot orari: dal Lunedì al Venerdì dalle 8:30 alle 19:00 e opzionalmente il Sabato mattina dalle 8:30 alle 13:00.
- *Locali*:
  - Uno Slot va allocato in un Locale del tipo richiesto.
  - Uno Slot va allocato in un Locale di capienza congrua alla cardinalità degli studenti frequentanti l'Insegnamento.
- *Slot*:
  - Alcuni Slot che devono essere pre-allocati in un predeterminato slot orario.
  - Possono essere definite delle relazioni binarie Slot: *almenoUnGiornoDopo*, *subitoDopo*, *stessoGiorno*, *giornoDiverso* e *parallelo*.
- *Docente*:

---

<sup>13</sup>Senza limitare la decisione ad un Giorno e Fascia Oraria specifica.

<sup>14</sup>Necessità si un certo lasso di tempo tra uno Slot e l'altro.

- Lo stesso Docente non può essere assegnato contemporaneamente a due Slot differenti.
- Non è possibile allocare uno Slot tenuto da un Docente in uno slot settimanale in cui ha indicato incompatibilità<sup>15</sup>.

### 3.4.2 Soft Constraint

Vengono elencati di seguito i vincoli flessibili ereditati, senza modifiche, dalla Tesi precedente.

- *Slot*:
  - Possono essere definite delle relazioni binarie Slot: *almenoUnGiornoDopo*, *subitoDopo*, *stessoGiorno*, *giornoDiverso* e *parallelo*.
- *Docente*:
  - Un Docente può esprimere parere favorevole o contrario circa la possibilità di fare lezione in uno slot settimanale.
  - Un Docente può esprimere una distanza minima/massima in termini di slot orari tra Slot a lui collegati ma appartenenti a Insegnamenti diversi nello stesso giorno.

## 3.5 Vincoli Ereditati e Modificati

In questa sezione vengono descritti i precedenti vincoli analizzati e come questi sono stati riadattati al nuovo modello.

### 3.5.1 Vincoli su Insegnamenti di un Orientamento

In precedenza sono stati stabiliti dei vincoli riguardo alla sovrapposizione degli Slot di vari Insegnamenti. Gli Insegnamenti venivano osservati all'interno di un Orientamento e inclusi in categorie: *Obbligatorio*, *Obbligatorio a Scelta*, *Suggerito*, *Tabella a Scelta*, *Credito Libero*. Dopo aver determinato la categoria di appartenenza per ciascun Insegnamento, venivano introdotti dei vincoli sulle coppie di Insegnamento, assegnando un peso in base alle categorie degli Insegnamenti considerati. Viene riportata in tabella 3.1 la modellazione

---

<sup>15</sup>L'Incompatibilità può essere derivata da motivi personali o da lezione da tenere per Insegnamenti di altri Corsi di Laurea

svolta nella precedenti tesi.

	Obbligatorio	Obbligatorio a scelta	Suggerito	Tabella a scelta	Credito libero
Obbligatorio	hard const.	hard const.	hard const.	LV_4*k	LV_3*k
Obbligatorio a scelta	hard const.	hard const.	hard const.	LV_4*k	LV_3*k
Suggerito	hard const.	hard const.	LV_4*k	LV_3*k	LV_2*k
Tabella a scelta	LV_4*k	LV_4*k	LV_3*k	LV_2*k	LV_1*k
Credito libero	LV_3*k	LV_3*k	LV_2*k	LV_1*k	LV_0

**Table 3.1:** tipologia Insegnamenti in Orientamento.

Dalla tabella riportata si evidenzia un utilizzo di penalità fisse per determinate coppie di Categorie. Tale modellazione pone un limite nella generazione dei vincoli. Per un certo Orientamento possono essere individuati Insegnamenti più pertinenti al percorso di studio o Insegnamenti la cui affluenza di studenti è maggiore rispetto ad altri. Si dovrebbe, quindi, preferire un modello in cui penalità maggiori vengono assegnate ad insegnamenti con più pertinenza o con più studenti in comune. Viene pertanto introdotto un nuovo modello basato sul concetto di “Correlazione”, la correlazione viene definita come un valore tra due Insegnamenti per cui maggiore è il valore, minore sarà il desiderio di avere gli orario sovrapposti.

$$Correlazione \in [0; 100] \tag{3.1}$$

$$\text{if } Correlazione = 100 \longrightarrow \text{Hard Constraint} \tag{3.2}$$

$$\text{if } Correlazione \in [1; 99] \longrightarrow \text{Soft Constraint} \tag{3.3}$$

$$\text{if } Correlazione = 0 \longrightarrow \text{No Constraint} \tag{3.4}$$

Definito questo concetto viene quindi ampliato il modello precedente (Tabella 3.1). Viene di seguito data una definizione formale del nuovo modello, stabilendo le regole per la generazione di tali valori, seguita poi da una tabella riassuntiva, la Tabella 3.2.

### Definizione Categorie

Qui le definizioni sono date prendendo come riferimento le pagine web che descrivono il piano di studi dei diversi Corsi di Laurea.<sup>16</sup>

<sup>16</sup>Non tutte le pagine sono strutturate allo stesso modo, la descrizione viene generalizzata per poter essere compatibile con le varie casistiche.

- *Obbligatorio*: Un insegnamento si definisce “Obbligatorio” quando appare nella tabella in riferimento ad uno specifico anno, senza che questi sia in opposizione con un altro insegnamento. Ogni studente che abbia scelto questo orientamento dovrà mettere nel suo piano di studi un insegnamento obbligatorio.
- *Obbligatorio a scelta*: Un insegnamento si definisce “Obbligatorio a scelta” quando appare nella tabella in riferimento ad uno specifico anno, ma è in opposizione ad un altro insegnamento. Questi includono sia insegnamenti per cui è possibile scegliere la lingua che insegnamenti nella stessa lingua<sup>17</sup>.
- *Insegnamento a scelta*: Un insegnamento si definisce a “Insegnamento a Scelta” quando non appare direttamente nella tabella in riferimento ad un anno, bensì si trova all’interno di tabelle esterne ed è un insegnamento che per lo più delle volte sarà in opposizione a diversi altri Insegnamenti<sup>18</sup>.

### Definizione Correlazione

I valori presenti in tabella 3.2 rappresentano la soglia minima che due Insegnamenti delle rispettive categorie debbano assumere. Tali valori per poter rispecchiare la realtà non devono essere fissi, l’obiettivo infatti è quello di creare un modello flessibile, in cui il rapporto tra gli insegnamenti tenga conto di fattori esterni.

### 3.5.2 Vincoli su Studenti iscritti ad un Orientamento

Nella generazione dell’orario è stato sicuramente importante tenere in considerazione le necessità degli studenti. Per generare un orario che non appesantisse la didattica sono stati presi in considerazione i seguenti vincoli [1]:

- *Hard Constraint*
  - Uno Studente non deve avere più di 5 slot orari consecutivi allocati in un determinato giorno tra gli Insegnamenti definiti obbligatori, suggeriti o obbligatori a scelta per il suo Orientamento.
  - Non deve avere più di 6 slot orari allocati in un determinato giorno tra gli Insegnamenti definiti obbligatori, suggeriti o obbligatori a scelta.
- *Soft Constraint*

---

<sup>17</sup>Dato un Corso di Laurea è possibile identificare una lingua principale in cui esso viene erogato, se la lingua corrisponde allora sarà considerato *Obbligatorio a Scelta (lingua principale)* altrimenti *Obbligatorio a Scelta (lingua secondaria)*

<sup>18</sup>Un Insegnamento a Scelta è Suggerito se appare così descritto all’interno della pagina

	Obbligatorio	Obbligatorio a scelta (lingua principale)	Obbligatorio a scelta (lingua secondaria)	Insegnamento a Scelta Suggestito	Insegnamento a Scelta
Obbligatorio	100				
Obbligatorio a scelta (lingua principale)	100	100			
Obbligatorio a scelta (lingua secondaria)	100	min 90	100		
Insegnamento a Scelta Suggestito	min 95	min 95	min 95	min 95	
Insegnamento a Scelta	min 1	min 1	min 1	min 1	min 1

**Table 3.2:** tipologia Insegnamenti in Orientamento.

- Se uno studente di un orientamento ha 4 slot orari di lezione consecutivi tra gli Insegnamenti definiti *core* per l’Orientamento allora viene assegnata una penalità LV\_1 al Piano Allocazione.
- Se uno studente di un orientamento ha 5 slot orari di lezione consecutivi tra gli Insegnamenti definiti *core* per l’Orientamento allora viene assegnata una penalità LV\_2 al Piano Allocazione.
- Se nell’orario di uno studente di un orientamento nella stessa giornata è presente un intervallo di 2 slot orari tra Slot appartenenti ad Insegnamenti definiti *core* per l’Orientamento allora viene assegnata una penalità LV\_1 al Piano Allocazione.
- Se nella stessa giornata è presente un intervallo di almeno 3 slot orari tra Slot appartenenti ad Insegnamenti definiti *core* per l’Orientamento.
- Se nella stessa giornata sono presenti due intervalli tra Slot appartenenti ad Insegnamenti definiti *core* per l’Orientamento allora viene assegnata una penalità LV\_6 al Piano Allocazione.

I vincoli descritti sono del tutto ragionevoli se si vuole garantire agli studenti un buon equilibrio nel carico di lavoro. Si noti però che, nella sopra citata analisi non sono stati presi in considerazione gli *Insegnamenti a Scelta* e sono stati attribuiti dei pesi non variabili alle soft constraint, pesi basati soltanto sulla descrizione del requisito. Questi vincoli portano ad una approssimazione del modello rispetto alla realtà. Si è quindi intervenuti per migliorare questo aspetto.

## Nuovo Modello

Il nuovo modello consiste in una struttura a *Percorsi*, non si visualizza più il problema all'interno di un Orientamento ma ad un livello globale<sup>19</sup>. Per “Percorso” si intende un insieme di Insegnamenti, erogati nello stesso periodo didattico, che potrebbero<sup>20</sup> far parte del carico didattico scelto da uno studente. Si mantengono intatti i requisiti formali precedenti, con l'introduzione di questo nuovo concetto, ovvero il percorso, si rende possibile l'inclusione nel problema, sia degli Insegnamenti a Scelta che della possibilità di bilanciare i pesi, attribuiti ai soft constraint<sup>21</sup>.

## 3.6 Vincoli Introdotti

### 3.6.1 Hard Constraint

- *Docenti*
  - A un Docente non possono essere allocati Slot contigui se appartenenti a Sedi lontane tra loro.
- *Slot*
  - Alcuni Slot potrebbero dover essere vincolati a specifici Giorni o Fasce Orarie<sup>22</sup>.

### 3.6.2 Soft Constraint

- *Docenti*
  - Un Docente potrebbe esprimere una preferenza riguardo a specifici Slot o interi insegnamenti circa gli Slot orari<sup>23</sup>.

---

<sup>19</sup>Un certo percorso potrebbe non essere esclusivo di un singolo Orientamento ma presentarsi in più Orientamenti

<sup>20</sup>Viene utilizzato il condizionale perché dato un Orientamento che prevede Insegnamenti a scelta, non tutti gli studenti iscritti a quell'Orientamento affronteranno lo stesso Percorso bensì una delle tante combinazioni generabili

<sup>21</sup>Per definire il peso si utilizza il numero di studenti in comune, privilegiando percorsi a più alta affluenza di studenti

<sup>22</sup>Ad esempio gli slot dei Crediti Liberi del terzo anno triennale si vogliono allocare il Lunedì o il Giovedì dopo le 14.30

<sup>23</sup>Ad esempio un Docente potrebbe specificare una preferenza nel tenere lezioni da 2 Slot consecutivi solo il Lunedì o il Martedì

## Capitolo 4

# Sovrapposizione Insegnamenti

Come già anticipato nel precedente capitolo, la gestione delle sovrapposizioni tra Insegnamenti è stato uno dei requisiti a cui si è posta particolare attenzione, analizzando il modello precedente si è intravista una lacuna riguardo i vincoli generati, in particolare per i valori di penalità dei soft constraint.

Con la seguente analisi si vuole introdurre il nuovo modello, dandone descrizione e la guida al software sviluppato.

### 4.1 Modellazione

Nella nuova modellazione è presente il concetto di *correlazione*, valore di supporto per la generazione dei vincoli. Dati due Insegnamenti che potrebbero sovrapporsi, sarà proprio il valore di correlazione a determinare il peso di tale vincolo. Pertanto il modello oltre a porre le fondamenta della soluzione, crea un ambiente per la decisione di tali valori. Maggiore sarà la pertinenza dei valori con la realtà migliore sarà il risultato ottenuto.

Le fondamenta di tale modello si basano sui dati reperibili dalle pagine web che riportano la descrizione degli Orientamenti<sup>1</sup>. Da queste pagine è infatti possibile stabilire una descrizione formale degli Insegnamenti per Orientamento. Con descrizione formale si intende sia collocare esattamente l'Insegnamento all'interno dell'Orientamento che stabilirne i rapporti con gli altri Insegnamenti (cfr. Sezione 3.5.1)

Per riportare i dati della descrizione formale è stato utilizzato il formato XML, dando prima una definizione dello schema xsd e poi estraendo e riportando i dati tramite software.

---

<sup>1</sup>Per ogni Corso di Laurea vi è una pagina web con una descrizione formale di tutti i suoi Orientamento

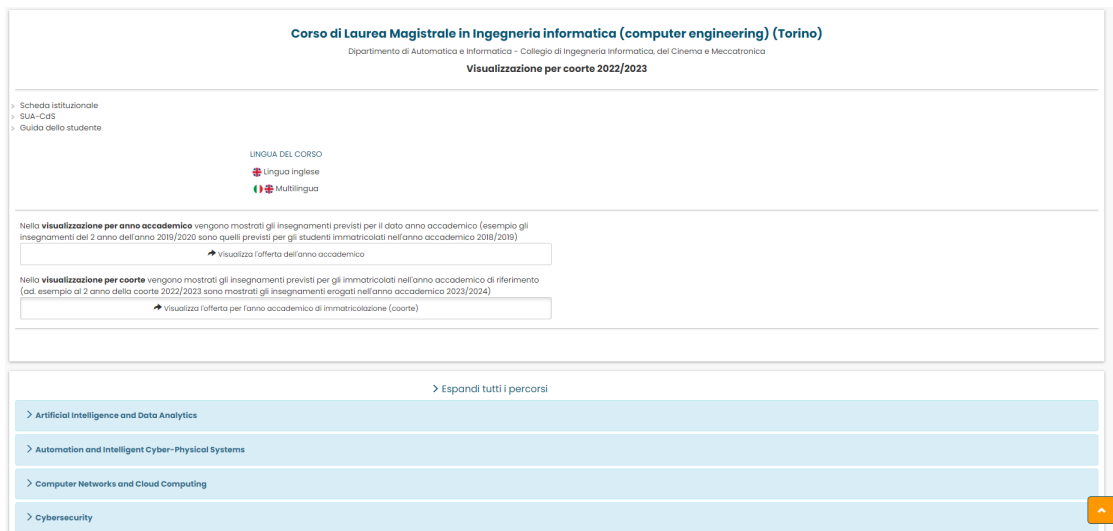


Figure 4.1: Screenshot pagina web CDL

Vengono di seguito spiegati i passaggi.

#### 4.1.1 Analisi dei dati

Le pagine web si presentano come in Figura 4.1 ove si può notare la presenza delle seguenti informazioni: il nome del Corso di Laurea, la lingua del Corso e infine l'elenco degli Orientamenti, che vengono rappresentati da tabelle come in Figura 4.2.

A loro volta gli Orientamenti sono composti da tabelle, essenzialmente di due tipi:

- Tabelle annuali: Descrivono uno specifico periodo didattico, elencando gli Insegnamenti ed eventualmente le Tabelle che ne fanno parte.
- Tabelle degli insegnamenti a scelta: Contengono un elenco di Insegnamenti mutualmente esclusivi, il raggiungimento dei crediti della tabella pone il limite al numero massimo di Insegnamenti selezionabili.

Un Insegnamento è descritto nella stessa tabella da diversi attributi, di cui solo un sottoinsieme sono necessari allo scopo, gli attributi sono:

- Semestre
- Codice Insegnamento



## Sovrapposizione Insegnamenti

▼ 1° anno 2022/2023									
Periodo	Codice	SSD	Insegnamento	Lingua	Crediti	Docente	Note	Orario	
1	02LSEOV	ING-INF/05 (0)	Computer architectures	🇮🇹	10	P. Montuschi	🟢		🕒
1	01NWMOV	ING-INF/01 (0)	Electronics for embedded systems	🇮🇹	10	C. Passerone	🟢		🕒
1	02NPSOV	ING-INF/05 (8)	Operating systems for embedded systems	🇮🇹	8	S. Di Carlo	🟢		🕒
1	02LQDOV	ING-INF/05 (6)	Specification and simulation of digital systems	🇮🇹	6	P. Camurati	🟢		🕒
2	01UDNOV	ING-INF/05 (6)	Cybersecurity for Embedded Systems	🇮🇹	6	P. Pinnetto	🟢		🕒
2	01NOYOV	ING-INF/01 (8)	Microelectronic systems	🇮🇹	6	M. Graziano	🟢		🕒
2	04GSPOV	ING-INF/05 (8)	Software engineering	🇮🇹	8	M. Morisio	🟢		🕒
2	02LVNOV	ING-INF/05 (6)	Synthesis and optimization of digital systems	🇮🇹	6	A. Collimera	🟢		🕒

▼ 2° anno 2023/2024									
Periodo	Codice	SSD	Insegnamento	Lingua	Crediti	Docente	Note	Orario	
1	01UDGOV	ING-INF/05 (6)	Energy management for IoT	🇮🇹	6	M. Panfili	🟢		🕒
1			Insegnamento a scelta 1		6		🟢		🕒
1	01QYHOV	ING-INF/05 (6)	System-on-chip architecture	🇮🇹	6	A. Macii	🟢		🕒
1	01RZSOV	ING-INF/05 (6)	Testing and fault tolerance	🇮🇹	6	M. Sorza Rearda	🟢		🕒
1,2	05KSPOV		Final project work		30		🟢		🕒
1,2	03UEWOV		Challenge	🇮🇹	8		🟢		🕒
2			oppure				🟢		
2			Crediti liberi		6		🟢		

▼ Insegnamento a scelta 1									
Periodo	Codice	SSD	Insegnamento	Lingua	Crediti	Docente	Note	Orario	
1	01DTDOV	ING-INF/01 (6)	Edge Computing Systems for AI and ML	🇮🇹	6	L. Lavagno	🟢		🕒
1	01NOXOV	ING-INF/01 (6)	Electronic systems engineering	🇮🇹	6	E. Passero	🟢		🕒
1	02JRCOV	ING-INF/05 (6)	Human-Computer Interaction	🇮🇹	6	L. De Russis	🟢		🕒
1	01URPOV	ING-INF/05 (6)	Machine learning for vision and multimedia	🇮🇹	6	F. Lamberti	🟢		🕒
1	01NWMOV	ING-INF/01 (8)	Modeling and optimization of embedded systems	🇮🇹	6	L. Lavagno	🟢		🕒
1	03QWROV	ING-INF/05 (6)	Programming for IoT applications	🇮🇹	6	E. Patti	🟢		🕒

**Figure 4.2:** Esempio Tabelle Orientamento

- Nome Insegnamento
- Lingua
- Crediti

Una volta compresa la Struttura degli Orientamento è possibile acquisire i dati al fine di individuare le coppie di Insegnamento per cui si ha bisogno di calcolare la correlazione<sup>2</sup>.

Si dica che due Insegnamenti hanno rischio di sovrapporsi, se e solo se, esiste almeno un Orientamento in cui possono essere scelti nello stesso Periodo Didattico<sup>3</sup>. Infatti è possibile che coppie di Insegnamenti che non hanno problema di sovrapposizione in un dato Orientamento lo abbiano in altri<sup>4</sup>. Per questo motivo, in un prima fase, i dati vengono raggruppati per Corso di Laurea e successivamente vengono uniti in un'unica struttura globale.

### 4.1.2 Schema XSD

A fronte di un'analisi fatta su tutti i Corsi di Laurea da considerare, sono sorte alcune non conformità tra le diverse pagine, cioè non è stato possibile individuare uno schema preciso

<sup>2</sup>Esistono insegnamenti che faranno mai parte dello stesso Orientamento nello stesso anno e semestre

<sup>3</sup>La coppia anno-semestre

<sup>4</sup>Anche di corsi di laurea differenti

per la disposizione delle informazioni. Lo schema quindi attraverso diverse correzioni, man mano che venivano trovate nelle pagine, è stato creato in modo da poter generalizzare l'attuale stato di esse, non è quindi possibile garantire la sua funzionalità a seguito di cambiamenti futuri.

Lo schema vale per file XML che rappresentino un singolo Corso di Laurea, vengono di seguito riportati estratti del codice:

- Corso di Laurea

```
1 <xs:element name="CDL">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element ref="orientamento" maxOccurs="unbounded" minOccurs="0"
5         "/>
6     </xs:sequence>
7     <xs:attribute type="xs:string" name="nome"/>
8   </xs:complexType>
9 </xs:element>
```

- Orientamento, definito come un sequenza di periodi

```
1 <xs:element name="orientamento">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element ref="periodo" maxOccurs="unbounded" minOccurs="0"/>
5     </xs:sequence>
6     <xs:attribute type="xs:string" name="nome" use="optional"/>
7   </xs:complexType>
8 </xs:element>
```

- Periodo, formato dall'anno e dal semestre a cui fa riferimento e da tutti gli Insegnamenti che vengono erogati in quel periodo

```
1 <xs:element name="periodo">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element ref="anno" minOccurs="1" maxOccurs="1"/>
5       <xs:element ref="semestre" minOccurs="1" maxOccurs="1"/>
6       <xs:element ref="insegnamenti_in_orientamento"/>
7     </xs:sequence>
```

```

8 </xs:complexType>
9 </xs:element>
10
11 <xs:element name="anno" type="xs:int"></xs:element>
12
13 <xs:element name="semestre" type="xs:int"></xs:element>

```

- Insegnamenti in orientamento, oggetto racchiude in se diverse casistiche, creando uno standard di rappresentazione per i dati presenti nelle pagine web.

```

1 <xs:element name="insegnamenti_in_orientamento">
2 <xs:complexType>
3 <xs:choice maxOccurs="unbounded" minOccurs="0">
4 <xs:element ref="scelta_tabella"/>
5 <xs:element ref="insegnamento"/>
6 <xs:element ref="scelta_tabella_lingua"/>
7 <xs:element ref="scelta_obbligatoria_lingua"/>
8 </xs:choice>
9 </xs:complexType>
10 </xs:element>
11

```

Le diverse casistiche considerate in questo modello sono le seguenti:

- Insegnamento, insegnamento obbligatorio che si trova all'interno di una tabella di uno specifico anno.

```

1 <xs:element name="insegnamento">
2 <xs:complexType>
3 <xs:simpleContent>
4 <xs:extension base="xs:string">
5 <xs:attribute type="xs:string" name="nome" use="optional"/>
6 </xs:extension>
7 </xs:simpleContent>
8 </xs:complexType>
9 </xs:element>
10

```

- Scelta obbligatoria lingua, sono gli insegnamenti che all'interno della tabella annuale si trovano in "oppure"<sup>5</sup>.

---

<sup>5</sup>In molti orientamenti questa formula è utilizzata per gli Insegnamenti in cui è possibile scegliere la lingua, tuttavia anche insegnamenti nella stessa lingua

```

1 <xs:element name="scelta_obbligatoria_lingua">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element ref="opzione" maxOccurs="unbounded" minOccurs="0"/>
5     </xs:sequence>
6   </xs:complexType>
7 </xs:element>
8

```

- Scelta tabella lingua, caso particolare che avviene nel secondo anno di Ingegneria Fisica L3, in cui non sono i singoli Insegnamenti ad essere in opzione, ma lo sono due tabelle. Vengono definite in `tabella_lingua` e comprendono un insieme di Insegnamenti equivalente ma in lingua opposta all'altra tabella.

```

1 <xs:element name="scelta_tabella_lingua">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element ref="tabella_lingua" maxOccurs="unbounded" minOccurs="0"/>
5     </xs:sequence>
6   </xs:complexType>
7 </xs:element>
8

```

- Tabella lingua, la tabella lingua differisce dalle comuni *scelta tabella*, in quanto questo tipo di tabella contiene soltanto Insegnamenti obbligatori non mutualmente esclusivi, la scelta di una tabella comporta infatti la selezione di tutti gli Insegnamenti che la compongono.

```

1 <xs:element name="tabella_lingua">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element ref="insegnamento" maxOccurs="unbounded" minOccurs="0"/>
5     </xs:sequence>
6     <xs:attribute type="xs:string" name="nome_tabella" use="optional"/>
7   </xs:complexType>
8 </xs:element>
9

```

- Scelta tabella, consiste in un insieme di insegnamenti mutualmente esclusivi. Gli

Insegnamenti vengono riportati come oggetto *opzione* e non come *Insegnamento*<sup>6</sup>.

```

1 <xs:element name="scelta_tabella">
2   <xs:complexType mixed="true">
3     <xs:sequence>
4       <xs:element ref="opzione" minOccurs="unbounded" minOccurs="0"/>
5     </xs:sequence>
6     <xs:attribute type="xs:string" name="nome_tabella" use="optional"/>
7     <xs:attribute type="xs:float" name="crediti" use="optional"/>
8   </xs:complexType>
9 </xs:element>
10
```

- Opzione, un'opzione è un insegnamento in mutua esclusione ad altri Insegnamenti e si trova all'interno di *scelta\_tabella* o di una *scelta\_obbligatoria*. Un'opzione contiene più attributi di un *Insegnamento*, si inserisce il numero di crediti utilizzato per riuscire a capire quanti insegnamenti possono essere scelti da una tabella<sup>7</sup>. Contiene poi un attributo di nome "correlazione" i cui valori possibili sono 20 o 95, non c'è nessuna regola fissa che vieti di utilizzare altri valori, tuttavia per semplicità in questa fase di sviluppo era stato deciso di utilizzare il valore 20 per tutti gli insegnamenti a scelta e 95 per gli insegnamenti suggeriti. Questi valori non sono definitivi, bensì verranno modificati in successive fasi. I valori per quanto non legati a nessuna regola o vincolo sono utili negli step seguenti almeno per visualizzare in maniera semplice i rapporti tra insegnamento<sup>8</sup>. Infine l'attributo "preferito\_lingua" si usa per le opzioni all'interno di una scelta obbligatoria lingua, dove il valore è impostato a "yes" se la lingua dell'insegnamento corrisponde alla lingua principale del corso di laurea.

```

1 <xs:element name="opzione">
2   <xs:complexType>
3     <xs:simpleContent>
4       <xs:extension base="xs:string">
5         <xs:attribute type="xs:string" name="nome" use="optional"/>
6         <xs:attribute type="xs:float" name="crediti" use="optional"/>
7         <xs:attribute type="xs:byte" name="correlazione" use="optional"/>
8       </xs:extension>
9     </xs:simpleContent>
10  </xs:complexType>
11 </xs:element>
```

<sup>6</sup>La selezione degli insegnamenti viene limitata attraverso l'attributo dei crediti

<sup>7</sup>Anche le tabelle hanno l'attributo crediti, se quest'ultimo è maggiore anche di un solo Insegnamento che ne fa parte esiste la possibilità di scegliere più di un insegnamento per tabella

<sup>8</sup>Ad esempio un valore 20 di correlazione tra due insegnamenti mi aiuta a capire che quei due insegnamenti hanno rischio di intersecarsi, se fosse 0 il problema non ci sarebbe perché consapevole che sono totalmente parallelizzabili.

```
8      <xs:attribute type="xs:string" name="preferito_lingua" use="
      optional"/>
9      </xs:extension>
10     </xs:simpleContent>
11     </xs:complexType>
12 </xs:element>
13
```

## 4.2 Raccolta dei dati

Lo studio dei dati e la successiva modellazione hanno portato ad un successivo sviluppo di un software in grado di navigare nelle pagine web, estrarre i dati necessari e riassumerli nello schema creato. Questa tecnica in termini Informatici è detta “Web Scraping”, pertanto il software sviluppato durante questa fase è stato denominato *Scraper*.

### 4.2.1 Definizione del Web Scraping

Il web scraping (detto anche web harvesting o web data extraction) è una tecnica informatica di estrazione di dati da un sito web per mezzo di programmi software. Di solito, tali programmi simulano la navigazione umana nel World Wide Web utilizzando l’Hypertext Transfer Protocol (HTTP) o attraverso browser, come Internet Explorer o Mozilla Firefox.

Strettamente correlato all’indicizzazione dei siti Internet, tale tecnica è attuata mediante l’uso di bot dalla maggior parte dei motori di ricerca. D’altro canto, il web scraping si concentra di più sulla trasformazione di dati non strutturati presenti in Rete, di solito in formato HTML, in metadati che possono essere memorizzati e analizzati in locale in un database [6].

### 4.2.2 Scraper

Lo scraper è stato sviluppato in Python utilizzando principalmente la libreria BeautifulSoup4 pensata apposta per questo tipo di utilizzi. Come primo passo si inseriscono in un file i riferimenti a tutti gli URL che si ha intenzione di considerare, ovvero tutti i corsi di laurea dei collegi ICM ed ETF. Da questi dati di input attraverso lo scraper vengono fatte richieste a queste pagine web utilizzando i dati contenuti nell’HTML per sistamarli in un file XML.

Esempi di output del tool sono riportati nelle seguenti figure:

- in Figura 4.3 è riportato un esempio di “Insegnamento” e “scelta obbligatoria lingua”;

```

CDL nome="Corso di Laurea Magistrale in Ingegneria informatica (computer engineering) (Torino)"
<orientamento nome="Artificial Intelligence and Data Analytics">
  <periodo>
    <anno>1</anno>
    <semestre>1</semestre>
    <insegnamenti_in_orientamento>
      <insegnamento nome="Big_data_processing_and_analytics">01DSHOV</insegnamento>
      <scelta_obbligatoria_lingua>
        <opzione nome="Architetture_dei_sistemi_di_elaborazione">02G0LOV</opzione>
        <opzione nome="Computer_architectures" preferito_lingua="yes">02LSEOV</opzione>
      </scelta_obbligatoria_lingua>
      <scelta_obbligatoria_lingua>
        <opzione nome="Data_Science_and_Database_Technology" preferito_lingua="yes">01SQJOV</opzione>
        <opzione nome="Data_Science_e_Tecnologie_per_le_Basi_di_Dati">01SQMOV</opzione>
      </scelta_obbligatoria_lingua>
      <scelta_obbligatoria_lingua>
        <opzione nome="Computer_network_technologies_and_services" preferito_lingua="yes">01OTWOV</opzione>
        <opzione nome="Tecnologie_e_servizi_di_rete">02KPNOV</opzione>
      </scelta_obbligatoria_lingua>
    </insegnamenti_in_orientamento>
  </periodo>

```

Figure 4.3: Esempio XML 1: Insegnamento e scelta obbligatoria lingua

```

<anno>2</anno>
<semestre>1</semestre>
<insegnamenti_in_orientamento>
  <insegnamento nome="Advanced_Machine_Learning">01URWOV</insegnamento>
  <scelta_tabella nome_tabella="Insegnamento_a_scelta_1" crediti="6.0">
    <opzione nome="Cloud_Computing_Technologies" crediti="6.0" correlazione="20">01HFPOV</opzione>
    <opzione nome="Computational_intelligence" crediti="6.0" correlazione="95">01URROV</opzione>
    <opzione nome="Distributed_systems_programming" crediti="6.0" correlazione="20">01TXZOV</opzione>
    <opzione nome="Human_Computer_Interaction" crediti="6.0" correlazione="20">02JSKOV</opzione>
    <opzione nome="Optimization_methods_and_algorithms" crediti="6.0" correlazione="20">01OUVOV</opzione>
    <opzione nome="Robot_Learning" crediti="6.0" correlazione="20">01HFNOV</opzione>
    <opzione nome="Software_Engineering_II" crediti="6.0" correlazione="20">01SQNOV</opzione>
  </scelta_tabella>

```

Figure 4.4: Esempio XML 2: scelta\_tabella

- in Figura 4.4 è riportato un esempio di “scelta\_tabella”;
- in Figura 4.5 è riportato un esempio di “scelta\_tabella\_lingua”.

### 4.2.3 Utilizzo dati Scraper

I dati così prodotti dallo scraper vengono ulteriormente maneggiati da altri script scritti in Python. L’obiettivo era quello di riorganizzare i dati in un altro formato per rendere più semplice l’intervento umano su di essi. A fine della fase precedente infatti si era ottenuta una descrizione formale degli Orientamenti ma non era ancora chiaro e facile da stabilire il valore di correlazione per una coppia di insegnamenti. Dagli XML generati era però facile intuire quali regole potessero attuarsi per stabilire quanto sarebbe dovuto essere il valore di correlazione tra due insegnamenti.

```

<periodo>
  <anno>2</anno>
  <semestre>1</semestre>
  <insegnamenti_in_orientamento>
    <scelta_tabella nome_tabella="Crediti_liberi_dal_catalogo_di_Ateneo_Grandi_Sfide_Globali" crediti="6.0"/>
    <scelta_tabella lingua>
      <tabella_lingua nome_tabella="2nd_year_taught_in_English">
        <insegnamento nome="Circuit_theory_and_applications">01QVQ0D</insegnamento>
        <insegnamento nome="Electronic_devices">02LTH0D</insegnamento>
        <insegnamento nome="Mathematical_analysis_II">03KXU0D</insegnamento>
        <insegnamento nome="Physics_II">06KXW0D</insegnamento>
      </tabella_lingua>
      <tabella_lingua nome_tabella="Secondo_anno_erogato_in_italiano">
        <insegnamento nome="Analisi_matematica_II">23ACI0D</insegnamento>
        <insegnamento nome="Dispositivi_elettronici">05APM0D</insegnamento>
        <insegnamento nome="Elettrotecnica">01AUL0D</insegnamento>
        <insegnamento nome="Fisica_II">19AXP0D</insegnamento>
      </tabella_lingua>
    </scelta_tabella_lingua>
  </insegnamenti_in_orientamento>
</periodo>

```

Figure 4.5: Esempio XML 3: scelta\_tabella\_lingua

## Regole

- Qualunque degli Insegnamenti che non faccia parte dello stesso periodo deve essere considerata come correlazione 0, cioè nell'allocazione i due insegnamenti sono totalmente parallelizzabili.
- Per gli Insegnamenti all'interno dello stesso periodo si fa invece riferimento alla tabella 3.2

Fatte queste considerazioni si voleva ottenere una visualizzazione semplice dei dati che permettesse anche la modifica dei valori di correlazione. Il modello visuale che si è decisi di adottare è quello di una matrice trinangolare, rappresentata come tabella triangolare in un file excel, avente come header nome dell'insegnamento e codice insegnamento, sia in riga che in colonna. In questo modo è possibile stabilire tramite incrocio il valore della coppia ed eventualmente modificarlo. La generazione dei dati è la seguente, lo script per ogni file XML che legge, quindi per ogni corso di laurea genera una cartella per il corso di laurea, all'interno viene generata una cartella per orientamento e infine due file Excel per orientamento più altri due per corso di laurea. I due file generati sono la rappresentazione dei dati contenuti nell'XML divisi per semestre<sup>9</sup>.

Questo output viene poi fatto visionare dai responsabili di corso in modo tale da poter intervenire grazie alle loro conoscenze interne, su quale possa essere un valore più adatto

<sup>9</sup>La divisione per semestre è non per anno e semestre è stata scelta perché alcuni insegnamenti che non all'interno di un orientamento non si trovano allo stesso anno, potrebbero farlo per altri orientamenti. Potrebbe quindi essere utili tenere tale informazione per il momento in cui verrà fatto un merge di tutti i dati



		Chemistry	Computer_sciences	English_Language_1st_level	Mathematical_analysis_I	Mathematical_analysis_II	Algorithms_and_data_struct	Electromagnetism_and_Circu	Mathematical_analysis_II	Computer_networks	Electronic_systems_technol	Operating_systems	Signal_analysis_and_process	Calcolo_numerico_metodi	Energia_progresso_e_soste	Etica_della_Tecnologia	Gestione_industriale_della	Imprenditorialita'_e_innova	Introduzione_alle_Applicazi	Modellazione_digitale_di_ir	Processi_di_re-impiego_e_v	Storia_delle_Scienze_e_dell	Strumenti_dell'Ingegneria_F
		06KWRMLM	07JCJLML	02MCCLML	04KWQLML	01URNLML	01URLML	07KXULML	08KSILML	06QXWMLM	04JEZMLM	02OGGLML	01SUNLML	01OYZMLM	01HMLWMLM	02BDYMLM	01RMLMLM	01DXULML	01DYBLML	01POSMLM	01FSWMLM	01RZWMLM	
Chemistry	06KWRMLM	0																					
Computer_sciences	07JCJLML	100	0																				
English_Language_1st_level	02MCCLML	100	100	0																			
Mathematical_analysis_I	04KWQLML	100	100	100	0																		
Algorithms_and_data_structures	01URNLML	0	0	0	0	0																	
Electromagnetism_and_Circuit_theory	01URLML	0	0	0	0	100	0																
Mathematical_analysis_II	07KXULML	0	0	0	0	100	100	0															
Computer_networks	08KSILML	0	0	0	0	0	0	0	0														
Electronic_systems_technologies_and_measurements	06QXWMLM	0	0	0	0	0	0	0	100	0													
Operating_systems	04JEZMLM	0	0	0	0	0	0	0	100	100	0												
Signal_analysis_and_processing	02OGGLML	0	0	0	0	0	0	0	100	100	100	0											
Calcolo_numerico_metodi_e_software	01SUNLML	0	0	0	0	0	0	0	20	20	20	20	0										
Energia_progresso_e_sostenibilita'	01OYZMLM	0	0	0	0	0	0	0	20	20	20	20	0	0									
Etica_della_Tecnologia	01HMLWMLM	0	0	0	0	0	0	0	20	20	20	20	0	0	0								
Gestione_industriale_della_qualita'	02BDYMLM	0	0	0	0	0	0	0	20	20	20	20	0	0	0	0							
Imprenditorialita'_e_innovazione	01RMLMLM	0	0	0	0	0	0	0	20	20	20	20	0	0	0	0	0						
Introduzione_alle_Applicazioni_Web	01DXULML	0	0	0	0	0	0	0	20	20	20	20	0	0	0	0	0	0					
Modellazione_digitale_di_impianti	01DYBLML	0	0	0	0	0	0	0	20	20	20	20	0	0	0	0	0	0	0	0			
Processi_di_re-impiego_e_valorizzazione_energetica	01POSMLM	0	0	0	0	0	0	0	20	20	20	20	0	0	0	0	0	0	0	0	0		
Storia_delle_Scienze_e_delle_Tecniche	01FSWMLM	0	0	0	0	0	0	0	20	20	20	20	0	0	0	0	0	0	0	0	0	0	
Strumenti_dell'Ingegneria_per_l'industria_4.0	01RZWMLM	0	0	0	0	0	0	0	20	20	20	20	0	0	0	0	0	0	0	0	0	0	0

Figure 4.6: Esempio di output dello scraper in Excel.

per certe coppie di insegnamento. La modifica va fatta solo su quei valori diversi da 0 o da 100, ovvero per tutti gli insegnamenti che hanno rischio di sovrapposizione ma che in base alla scelta dell'Orientamento potrebbe assumere più o meno importanza. Una volta ottenuti tutti i con le modifiche apportate si procede con la generazione di un unico file Excel per CDL, che racchiude tutte le informazioni prese dagli Orientamenti<sup>10</sup>. Manca ancora qualcosa prima di utilizzare questi dati definitivamente, bisogna considerare che certi insegnamenti sono mutuati o aggregati ad altri insegnamenti, questo comporta che alcuni insegnamenti pur essendo lo stesso tra i vari Orientamenti risulteranno avere un Codice Insegnamento differente. Qui entra in gioco il file fornito dall'ateneo contenente delle informazioni che ci permettono di ricondurre diversi Codici insegnamento a degli id univoci, questi id univoci sono essenziali per avere una visione globale del problema e permetterci di fare le giuste considerazioni nel calcolo finale delle correlazioni. Tramite altro script vengono utilizzati i dati a nostra disposizione per generare le coppie di insegnamenti finali. Queste coppie saranno identificate con il loro id incrementale e avranno un valore di correlazione che prenderà informazioni globali, perché saranno incluse le informazioni di tutti gli Orientamenti. Tutti i dati generati vengono salvati all'interno del database nella

<sup>10</sup>Ad esempio se in un orientamento una coppia a correlazione bassa e in un altro è 100 è ovvio che debba essere preso in considerazione il valore maggiore.

		Chemistry	Computer_sciences	English_Language_1st_level	Mathematical_analysis_I	Mathematical_analysis_II	Algorithms_and_data_struct	Electromagnetism_and_Circ	Mathematical_analysis_II	Computer_networks	Electronic_systems_technol	Operating_systems	Signal_analysis_and_proces	Calcolo_numerico_metodi_	Energia_progresso_e_soste	Etica_della_Tecnologia	Gestione_industriale_della	Imprenditorialita_e_innova	Introduzione_alle_Applicazi	Modellazione_digitale_di_ir	Processi_di_re-impiego_e_u	Storia_delle_Scienze_e_dell	Strumenti_dell'Ingegneria_f
		06KWRLM	07JCJLM	02MCCLM	04KWQLM	01URNLM	01URLLM	07KXULM	08KSILM	06QXWLM	04IEZLM	02OGGLM	01SUNLM	01OYZLM	01HMMWLM	02BDYLM	01RMMLM	01DXULM	01DYBLM	01POSUM	01FSWLM	01RZWLM	
Chemistry	06KWRLM	0																					
Computer_sciences	07JCJLM	100	0																				
English_Language_1st_level	02MCCLM	100	100	0																			
Mathematical_analysis_I	04KWQLM	100	100	100	0																		
Algorithms_and_data_structures	01URNLM	0	0	0	0	0																	
Electromagnetism_and_Circuit_theory	01URLLM	0	0	0	0	100	0																
Mathematical_analysis_II	07KXULM	0	0	0	0	100	100	0															
Computer_networks	08KSILM	0	0	0	0	0	0	0	0														
Electronic_systems_technologies_and_measurements	06QXWLM	0	0	0	0	0	0	0	100	0													
Operating_systems	04IEZLM	0	0	0	0	0	0	0	100	100	0												
Signal_analysis_and_processing	02OGGLM	0	0	0	0	0	0	0	100	100	100	0											
Calcolo_numerico_metodi_e_software	01SUNLM	0	0	0	0	0	0	0	30	30	30	30	0										
Energia_progresso_e_sostenibilita'	01OYZLM	0	0	0	0	0	0	0	20	20	20	20	0	0									
Etica_della_Tecnologia	01HMMWLM	0	0	0	0	0	0	0	30	30	30	30	0	0	0								
Gestione_industriale_della_qualita'	02BDYLM	0	0	0	0	0	0	0	30	30	30	30	0	0	0	0							
Imprenditorialita_e_innovazione	01RMMLM	0	0	0	0	0	0	0	50	50	50	50	0	0	0	0	0						
Introduzione_alle_Applicazioni_Web	01DXULM	0	0	0	0	0	0	0	100	100	100	100	0	0	0	0	0	0					
Modellazione_digitale_di_impianti	01DYBLM	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0		
Processi_di_re-impiego_e_valorizzazione_energetica	01POSUM	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	
Storia_delle_Scienze_e_delle_Tecniche	01FSWLM	0	0	0	0	0	0	0	30	30	30	30	0	0	0	0	0	0	0	0	0	0	0
Strumenti_dell'Ingegneria_per_l'industria_4.0	01RZWLM	0	0	0	0	0	0	0	30	30	30	30	0	0	0	0	0	0	0	0	0	0	0

Figure 4.7: Esempio di tabella correlazioni dopo la modifica dei Responsabili.

Tabella "Info\_correlazione" avente 4 colonne:

- ID\_INC1
- ID\_INC2
- correlazione
- correlazione\_finale

I nomi sono autoesplicativi, ma manca ancora un'informazione. Le colonne occupate dai dati prima generati occuperanno solo le prime 3 colonne l'ultima colonna è riservata ad un'altra fase, volta alla ulteriore correzione dei valori. In quest'ultima fase vengono presi in considerazione gli studenti iscritti ai vari Insegnamenti si genera un'intersezione di matricole e si calcola il numero di studenti iscritti alle coppie di insegnamenti. I valori di correlazione vengono corretti attraverso delle funzioni dipendenti da

- numero studenti in comune
- percentuale studenti di un insegnamento sull'altro
- correlazione di partenza

## Descrizione funzione matematica scelta

I dati in ingresso alla funzione sono i seguenti:

- $nStud$ <sup>11</sup>

- $p1$ <sup>12</sup>

$$p1 = \frac{nStud}{totStudIns1} \quad (4.1)$$

- $p2$ <sup>13</sup>

$$p2 = \frac{nStud}{totStudIns2} \quad (4.2)$$

- $corr$ <sup>14</sup>

Non a tutte le coppie di Insegnamento è permesso arrivare ad avere correlazione a 100, ovvero imponendo un hard constraint, si stabilisce che due Insegnamenti che superino il valore 100 attraverso la formula avranno attribuito come valore finale 100 se gli studenti in comune superano una certa soglia 99 viceversa.

```

1 def gen_correlazione_finale(nStud, p1, p2, corr):
2   with open('info_corr_finale.txt', 'a+') as log_info:
3     max_corr = 100 if nStud > 50 else 99
4     if corr == 100:
5       return 100
6     elif corr in range(90, 100):
7       new_val = corr + nStud // 5
8     else:
9       new_val = corr + min(p1,p2) + 3*(nStud//10)
10    if(new_val > 100):
11      log_info.write(f'E stato superato il valore 100 con max_corr = {
12      max_corr}\n')
13    return min(new_val, max_corr)

```

Un esempio delle informazioni salvate nel database a fine procedura:

<sup>11</sup>Il numero di studenti in comune

<sup>12</sup>La percentuale di che  $nStud$  occupa sul numero totale di studenti del primo insegnamento

<sup>13</sup>La percentuale di che  $nStud$  occupa sul numero totale di studenti del secondo insegnamento

<sup>14</sup>Correlazione generata dalle fase di correzione dei responsabili di corso

ID_INC1	ID_INC2	correlazione	correlazione_finale
273709	274315	60	88
273002	273264	80	87
273263	273704	80	86
273263	273466	60	86
273572	274855	80	84
273703	274855	80	84
272697	273477	80	84
272697	273577	80	84
273204	274331	80	84
272697	273263	80	83
274331	275208	80	83
273457	273463	60	82
273438	273477	80	81
273439	273477	80	80
273466	273577	60	77
273263	273467	60	77
273466	273709	60	74
272778	273703	60	74
273263	273709	60	72

Table 4.1: Parte tabelle estratta da DB Finale

## 4.3 Manuale d'uso

### 4.3.1 lo script scraper.py

Questo script lanciato da solo sarà in grado, se definiti gli URL delle pagine web, di generare i vari file XML descritti nella Sezione 4.3.2. Prima di eseguirlo sarà quindi necessario andare ad impostare eventuali valori all'interno del file constantsScraper.py, a meno che non vogliano essere apportate modifiche al programma, gli unici parametri da modificare sono gli [URI\\_CDL](#) ed eventualmente il percorso in cui si intendono salvare i file generati [BASE\\_PATH\\_FILE\\_XML](#).

### 4.3.2 extract\_excel\_from\_xml.py

Questo script contiene al suo interno due funzioni, una funzione per la creazione dei file excel pronti alla modifica dei responsabili, la struttura di come vengono salvati nel filesystem è la seguente:

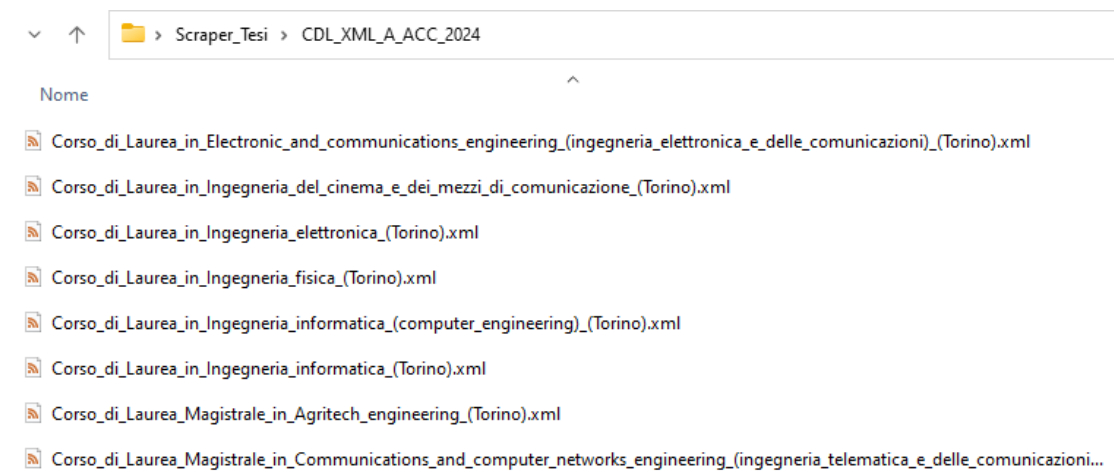


Figure 4.8: Risultato post esecuzione scraper.py

```

1 [CDL_EXCELS](Nome cartella arbitrario, root): # Tra [] le directory, *
   elementi che possono ripetersi
2 *[NOME_CD_L]:
3   *[NOME_ORIENTAMENTO]:
4     excel_orientamento_semestre1.xlsx
5     excel_orientamento_semestre2.xlsx
6 excel_cdl_semestre1.xlsx
7 excel_cdl_semestre2.xlsx

```

La struttura dei file Excel segue la figura 4.3, i responsabili potranno intervenire sia sui file “orientamento” che “cdl”, essendo i file cdl completi di tutti gli orientamenti sarà sufficiente intervenire o solo sul file cdl o su tutti i file orientamento. Nel caso venga seguita la seconda procedura, a modifiche ultimate dovrà essere eseguita la seconda funzione dello script la quale si occuperà di generare nuovamente i file cdl dai file orientamento modificati, bisogna però modificare i percorsi utilizzati e assicurarsi che siano corretti prima dell’esecuzione.

### 4.3.3 generateCompleteExcel.py

Script che partendo dai file excel\_cdl\_semestreX.xlsx di tutti i cdl e con l’appoggio dei dati forniti dal GOF, trasforma le coppie di codici insegnamento in coppie di ID\_INC in modo tale da poter trovare un valore globale valido indipendentemente dal CDL in esame.

#### 4.3.4 gen\_file\_intersezioni.py

Partendo da una directory contenente file tipo “id\_inc”.xlsx con l’elenco delle matricole iscritte all’insegnamento viene generato un file unico, il cui nome può essere deciso modificando lo script, che riassume tutte le intersezioni diverse da 0 tra due Insegnamenti. Il file è strutturato in 5 colonne

Insegnamento1	Insegnamento2	NStudenti	Intersezione1	Intersezione2
---------------	---------------	-----------	---------------	---------------

- Insegnamento1<sup>15</sup>
- Insegnamento2<sup>16</sup>
- NStudenti<sup>17</sup>
- Intersezione1<sup>18</sup> come da equazione 4.1
- Intersezione2<sup>19</sup> come da equazione 4.2

#### 4.3.5 compute\_final\_corr.py

Infine, eseguendo quest’ultimo script mandando in input il file generato al passo precedente, i valori di correlazione salvati nel database verranno corretti attraverso la funzione 4.2.3

---

<sup>15</sup>ID INC insegnamento 1

<sup>16</sup>ID INC insegnamento 2

<sup>17</sup>Studenti in Intersezione

<sup>18</sup>Percentuale degli studenti di Insegnamento1 che seguono anche Insegnamento2

<sup>19</sup>Percentuale degli studenti di Insegnamento2 che seguono anche Insegnamento1

## Capitolo 5

# Funzionalità aggiuntive del modello

In questo capitolo verranno elencati e spiegati altri aspetti delle funzionalità che sono state modificate e aggiunte al modello precedente.

### 5.1 Gestione Studenti

Come già anticipato nel Capitolo 3, i requisiti degli studenti necessitavano di una modifica in fase di creazione, in quanto i vincoli non era applicati ad un intero percorso che uno studente potesse scegliere, bensì venivano considerati solo gli Insegnamenti Obbligatorii, non considerando gli Insegnamenti a scelta si aveva un risultato approssimato della realtà, alcuni studenti potevano quindi non vedere rispettati tali vincoli. Metodo risolutivo per rimuovere L'approssimazione dal modello è stata l'idea di voler definire i vincoli studenti non per Orientamento ma globalmente tramite l'introduzione del concetto di Path o percorso.

#### 5.1.1 Definizione Path

Dato un orientamento con insegnamenti a scelta il percorso è definito come gli insegnamenti che uno studente potrebbe scegliere di frequentare. Per generare tutti i path possibili di un Orientamento di utilizza il calcolo combinatorio, dopotutto un percorso andrebbe visto come una combinazione senza ripetizione di insegnamenti e non è importante l'ordine in cui vengono considerati. La dimensione di un percorso non è fissa, si danno le seguenti definizioni matematiche.

## Legenda

- nI = Numero di Insegnamenti Obbligatori
- nSO = Numero di Scelte obbligatorie lingua
- nTS = Numero di Tabelle scelta
- nI\_TS = Numero di Insegnamenti in una Tabella Scelta

## Definizioni

La lunghezza di un path viene calcolata come:

$$nI + nSO + nTS \quad (5.1)$$

Il numero di combinazioni ottenibili in un dato Orientamento è definito come

$$(1^{nI}) * (2^{nSO}) * \left( \prod_{n=1}^{nTS} nI\_TS_n \right) \quad (5.2)$$

### 5.1.2 Implementazione

Per la generazione dei percorsi sono stati utilizzati i file generati durante la fase di scraping, in particolare i file XML che descrivono i diversi CDL. Da questi dati è stato generato un grafo orientato. Si parte da due nodi “Start” e “End” e selezionando in ordine Insegnamenti Obbligatori, Obbligatori a scelta e infine Insegnamenti a Scelta si iniziavano ad inserire nuovi nodi per ogni Insegnamento incontrato, tutti gli Insegnamenti mutualmente esclusivi sono dei nodi con gli stessi collegamenti (i.e. i collegamenti a nodi precedenti e a nodi successivi è lo stesso), non vi è nessun peso per gli archi del grafo, quindi nel momento in cui si calcolano tutti i percorsi minimi che portano da Start ad End ci porta alla creazione di tutte le combinazioni di insegnamento possibili, ovvero riusciamo così a definire i percorsi di un Orientamento in un certo periodo. Una volta generati i percorsi se ne prova a calcolare il peso. Il peso in questo caso consiste nel numero di studenti in comune a tutti gli insegnamenti del percorso. A volte i dati potrebbero non essere sufficienti<sup>1</sup>, in quei casi l’insegnamento viene scartato e considerato il sotto-percorso. Nel caso non sia abbia nessun dato disponibile per nessuno degli insegnamenti di un percorso, come ad esempio per i nuovi Corsi di Laurea Magistrale, si decide di attribuire un valore simbolico di peso, equivalente a -1, poichè la problematica è stata riscontrata

---

<sup>1</sup>Potrebbe non essere disponibile il file contenenti gli studenti iscritti ad esempio per gli insegnamenti di nuova istituzione



negli insegnamenti di corsi di laurea nuovi di cui non si hanno a disposizione tali dati. È importante tenere in considerazione questi corsi anche senza la presenza di vero è proprio peso.

### 5.1.3 Manuale d'uso

Per definire i vincoli ed integrarli al calcolo del programma è prima necessario definire un file json contenente tali informazioni, in modo tale da poterlo inserire come input al programma. Innanzitutto bisogna essere in possesso dei file XML descrittivi dei diversi corsi di laurea e dei file Excel contenenti le matricole iscritti agli Insegnamenti. Soddisfatti i requisiti si utilizza uno script in Python che genererà un file json in output in cui saranno presenti i vari percorsi con il peso<sup>2</sup> loro associato. Viene riportato un esempio del file di output generato:

```

1 {
2     "273696_274702_276441": 102,
3     "273693_274702_276441": 102,
4     "273697_274703_276442": 100,
5     "273694_274703_276442": 100,
6     "272693_273260_274336": 97,
7     "273467_273574": 93,
8     "273698_274704_276443": 91,
9     "273695_274704_276443": 91,
10    "273417_274855": 82,
11    "272595_272625_272626_274336": 80,
12    "272754_273204_274460": 80,
13    "274372_275354_275449_276165": 79,
14    "272754_274460_275208": 78,
15    "273577_273709": 75,
16    "272755_273204_274460": 74,
17    "274402_275555_275710": 72,
18    "272755_274460_275208": 72,
19    "272610_272778": 70,
20    "274373_275355_275450_276166": 61,
21    "272610_272778_273417": 61,
22    "273263_273577": 57,
23    "273419_274855": 56,
24    "272698_273003": 54,
25    "273001_273391": 53,
26    "275166_275698_275828": 45,
27    "274382_274902_275110": 40
28 }
```

**Listing 5.1:** Estratto file json Percorsi studenti

<sup>2</sup>Intersezione numero studenti

Due parametri da tenere in considerazione prima di eseguire il calcolo della soluzione è impostare all'interno del main.py due parametri:

- set\_pathPercorsiStudenti
- set\_sogliaStudenti

Il primo serve per definire il percorso del file contenente le informazioni, il secondo per decidere una eventuale soglia per i percorsi che debbano essere trasformati in hard constraint e i soft constraint con alta penalità. Questa possibilità viene data sia per permettere un testing più veloce che per poter rimediare a casi estremi in cui sono presenti molti vincoli stringenti che non permette al programma di trovare soluzione. Per cui si fornisce un'alternativa per allentare qualche constraint senza però trascurare del tutto alcuni percorsi.

## 5.2 Implementazione dei nuovi Requisiti

Sono stati implementati nuovi requisiti in particolare per la gestione di Slot o Insegnamenti per il quale certi Slot Temporal<sup>3</sup> erano preferibili, e per la gestione di tempo di pausa tra Slot associati ad uno stesso docente ma in Sedi diverse. Per poter implementare correttamente queste funzioni sono state fatte delle modifiche all'interno dello schema utilizzato per la descrizione degli insegnamenti. Si fa riferimento al Capitolo 4 e alla Sezione 4.2.1 della tesi di partenza [1], in cui vengono fornite informazioni riguardo al file schemaUseCase.xsd al quale sono state apportate le seguenti modifiche.

### 5.2.1 Modifiche allo Schema XML

#### Premesse

Sono necessarie le seguenti premesse.

- Gli Insegnamenti sono descritti tramite file in linguaggio XML seguendo il testo sopra citato, l'oggetto minimo che rappresenta la struttura degli slot di un determinato Insegnamento è chiamato *TemplateOrario* definito come:
- Template Orario:

---

<sup>3</sup>Giorni e Fasce orarie

```

1  <xs:element name="TemplateOrario">
2  <xs:complexType>
3  <xs:sequence>
4  <xs:element name="id" minOccurs="1" maxOccurs="1"></
xs:element>
5  <xs:element name="ID_INC" type="xs:int" minOccurs="1
" maxOccurs="1"></xs:element>
6  <xs:element ref="NumIscritti" minOccurs="1"
maxOccurs="1"></xs:element> <!-- se noto, altrimenti 0, verrà
estratto dal GOF -->
7  <xs:element ref="NumIscrittiPrimaFrequenza"
minOccurs="0" maxOccurs="1"></xs:element>
8  <!-- se noto, altrimenti 0, verrà estratto dal GOF
-->
9  <xs:element ref="CapienzaAula" minOccurs="0"
maxOccurs="1"></xs:element> <!-- se il Docente vuole specificare
la capienza dell'Aula -->
10 <xs:element ref="SlotScelto" minOccurs="0" maxOccurs
="unbounded"></xs:element>
11 <xs:element ref="SlotGenerico" minOccurs="0"
maxOccurs="unbounded"></xs:element>
12 <xs:element ref="Vincolo" minOccurs="0" maxOccurs="
unbounded"></xs:element>
13 <xs:element ref="Operatore" minOccurs="0" maxOccurs="
unbounded"></xs:element>
14 <xs:element name="Note" type="xs:string" minOccurs="
0" maxOccurs="1"></xs:element>
15 </xs:sequence>
16 </xs:complexType>
17 </xs:element>

```

Listing 5.2: Estratto tesi precedente [1]

- Alcuni Docenti tengono lezioni in Insegnamenti di altri collegi, questi Insegnamenti non andrebbero considerati, tuttavia rappresentando questi indisponibilità per il docente sono informazioni necessarie al programma.
- Alcuni Insegnamenti potrebbero avere l'orario già assegnato con già le aule predisposte.

In precedenza si creava un TemplateOrario con solo SlotScelti per le casistiche indicate sopra, si è quindi introdotto un nuovo oggetto molto simile al TemplateOrario chiamato *OrarioFisso* e *OrarioFissoNoAula*. Per avere una rappresentazione più ordinata era stato creato OrarioFisso con le stesse funzioni di TemplateOrario ma pensato per avere solo SlotScelti. Da lì è stato esteso ad OrarioFissoNoAula in quanto il numero di aule disponibili per i collegi è dato al programma, per evitare quindi sovraccarichi sul modello è stato utilizzato questo oggetto che internamente al programma è processato in maniera tale da non diminuire il numero di aule disponibili.

## OrarioFisso

```

1 <xs:element name="OrarioFisso">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element name="id" minOccurs="1" maxOccurs="1"></xs:element>
5       <xs:element name="ID_INC" type="xs:int" minOccurs="1" maxOccurs=
6 "1"></xs:element>
7       <xs:element ref="NumIscritti" minOccurs="1" maxOccurs="1"></
8 xs:element>
9       <xs:element ref="NumIscrittiPrimaFrequenza" minOccurs="0"
10 maxOccurs="1"></xs:element>
11       <xs:element ref="CapienzaAula" minOccurs="0" maxOccurs="1"></
12 xs:element>
13       <xs:element ref="SlotScelto" minOccurs="0" maxOccurs="unbounded"
14 ></xs:element>
15       <xs:element ref="Vincolo" minOccurs="0" maxOccurs="unbounded"></
16 xs:element>
17       <xs:element ref="Operatore" minOccurs="0" maxOccurs="unbounded">
18 </xs:element>
19       <xs:element name="Note" type="xs:string" minOccurs="0" maxOccurs
20 ="1"></xs:element>
21       <xs:element ref="Sede" minOccurs="0" maxOccurs="1"></xs:element>
22     </xs:sequence>
23   </xs:complexType>
24 </xs:element>

```

OrarioFissoNoAula ha gli stessi attributi ma il tag diverso che viene processato per creare Slot di tipo TipoSlot.SlotSceltoNoAula.

### 5.2.2 Vincoli Restrizioni Temporal

Nel precedente lavoro era già stata la possibilità ai docenti, tramite sondaggio jotform, di poter esprimere preferenze sullo slot orario a cui essere assegnati ma questa preferenza veniva però applicata a tutti gli slot di cui fa parte il docente. Le Restrizioni Temporal sono un meccanismo invece che permettere di specificare la preferenza di uno o diversi Slot orari per precisi Slot o anche per un intero Insegnamento.

#### Modifiche e aggiunte allo Schema XML

L'elemento RestrizioniTemporal è definito da un'insieme di Giorni e FasceOrarie e da una penalità che può essere sia di tipo hard che soft. L'oggetto può essere inserito all'interno di SlotGenerico o di TemplateOrario con cardinalità minima uguale a 0 e massima non

definita<sup>4</sup>. Ad esempio, potrebbe essere necessario modellare il seguente scenario:

- si vuole per un certo Insegnamento che lo slot da 3h venga allocato il lunedì in qualsiasi Fascia Oraria o eventualmente il martedì dopo le 14.30;
- si vuole invece per lo slot da 1.5h dello stesso Insegnamento che questi venga allocato alle 17.30 di qualsiasi giorno;
- e questi devono essere requisiti hard.

Il requisito verrebbe descritto nel seguente modo (Generalizzando):

```

1 <SlotGenerico slotId="slot1">
2   <NumSlotConsecutivi>2</NumSlotConsecutivi>
3   <RestrizioniTemporali>
4     <Giorno>Lun</Giorno>
5     <Penalita>LVH</Penalita>
6   </RestrizioniTemporali>
7   <RestrizioniTemporali>
8     <Giorno>Mar</Giorno>
9     <FasciaOraria>14.30-16.00</FasciaOraria>
10    <FasciaOraria>16.00-17.30</FasciaOraria>
11    <Penalita>LVH</Penalita>
12  </RestrizioniTemporali>
13 </SlotGenerico>
14
15 <SlotGenerico slotId="slot2">
16   <NumSlotConsecutivi>1</NumSlotConsecutivi>
17   <RestrizioniTemporali>
18     <FasciaOraria>16.00-17.30</FasciaOraria>
19     <Penalita>LVH</Penalita>
20   </RestrizioniTemporali>
21 </SlotGenerico>
22

```

Similmente potrebbero venire aggiunte ad un TemplateOrario, è anche possibile definire contemporaneamente RestrizioniTemporali sia per TemplateOrario che per SlotGenerico, nel caso questo avvenga le Restrizioni verranno applicato con priorità. Ogni Slot vedrà applicati solo i vincoli delle proprie RestrizioniTemporali, se non presenti verranno applicate quelle presenti nel TemplateOrario.

Viene anche riportato lo schema delle restrizioni temporali.

<sup>4</sup>Possono esserci più restrizioni per permettere diverse combinazioni

```

1 <xs:element name="RestrizioniTemporali">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element ref="FasciaOraria" minOccurs="0" maxOccurs="
unbounded"/></xs:element>
5       <xs:element ref="Giorno" minOccurs="0" maxOccurs="unbounded"/></
xs:element>
6       <xs:element ref="Penalita" minOccurs="1" maxOccurs="1"/></
xs:element>
7     </xs:sequence>
8   </xs:complexType>
9 </xs:element>

```

### 5.2.3 Vincoli sedi diverse

Il problema delle sedi distaccate è già stato descritto, si vuole in questa sezione descrivere le scelte fatte a fronte di considerazioni euristiche e di simulazioni effettuate.

#### Descrizione Formale

- Elenco delle sedi
  - Sede Centrale - Cittadella Politecnica
  - Sede Centrale - Corso Duca degli Abruzzi
  - Castello del Valentino
  - Via Morgari
  - Lingotto
  - Mirafiori
- Considerazioni
  - Sede Centrale - Cittadella Politecnica e Sede Centrale - Corso Duca degli Abruzzi si trovano vicine tra loro
  - Castello del Valentino e Via Morgari si trovano vicine tra loro
  - Mirafiori è la sede con maggiore distanza media
- Scelte
  - Se gli Insegnamenti si trovano in sedi vicine non viene aggiunto nessun vincolo
  - Se uno degli Slot ha sede a Mirafiori la distanza temporale tra slot dovrebbe essere almeno di due slot temporali (3h)

- Tutti gli altri casi di sedi diverse dovrebbero avere almeno uno slot temporale di distanza pari a 1(1.5h)

- Funzione

$$\begin{aligned}
 & d \in D, \forall s1, s2 \in slotDocente(d), s1 \neq s2, \\
 & sed1 = sede(s1), sed2 = sede(s2), dist = evalDistanzaSedi(s1, s2) \\
 & se : X_{d_{s1}} = X_{d_{s2}} \ \& \ dist > 0 \Rightarrow \\
 & (X_{h_{s1}} \geq X_{h_{s2}} + numSlot(s2) + dist) \vee (X_{h_{s2}} \geq X_{h_{s1}} + numSlot(s1) + dist)
 \end{aligned}
 \tag{5.3}$$

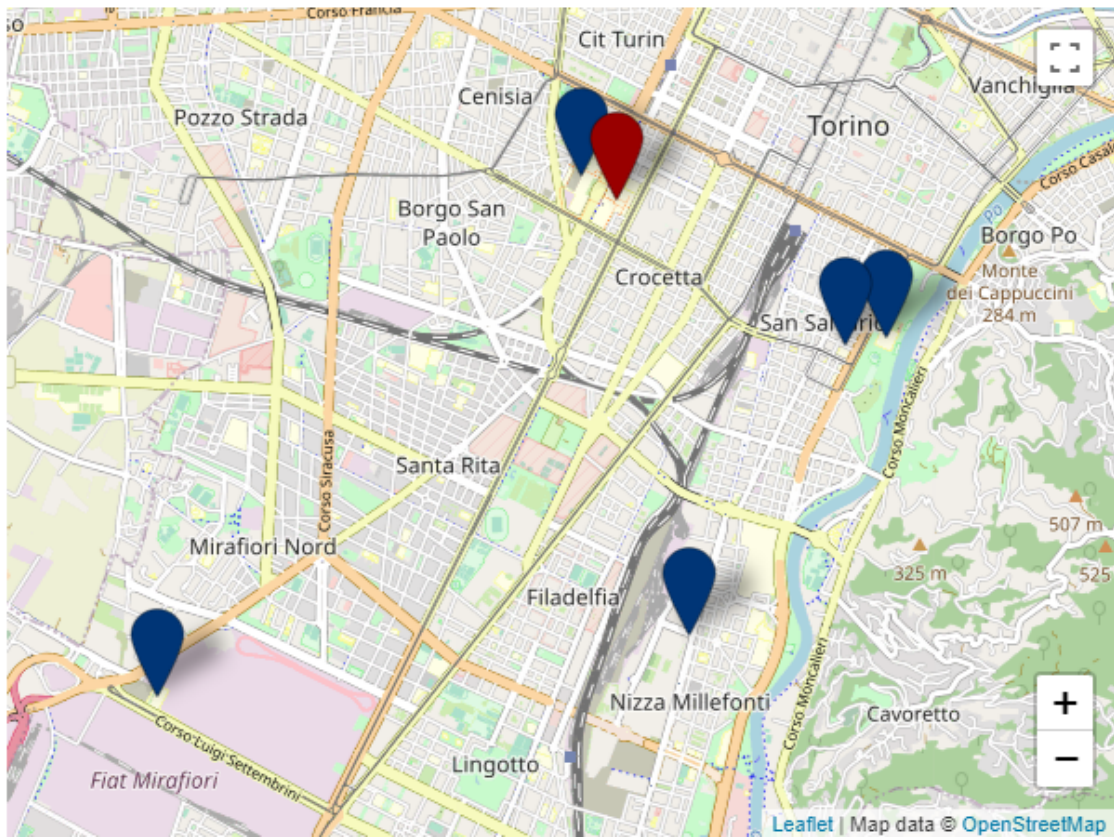


Figure 5.1: Mappa sedi Politecnico

## XSD

Si riporta la struttura dati enumerativa usata per lo Schema XML.

```
1 <xs:element name="Sede">
2   <xs:simpleType>
3     <xs:restriction base="xs:string">
4       <xs:enumeration value="Castello del Valentino"></xs:enumeration>
5       <xs:enumeration value="Lingotto"></xs:enumeration>
6       <xs:enumeration value="Mirafiori"></xs:enumeration>
7       <xs:enumeration value="Sede Centrale - Cittadella Politecnica">
8         </xs:enumeration>
9       <xs:enumeration value="Sede Centrale - Corso Duca degli Abruzzi">
10        </xs:enumeration>
11      <xs:enumeration value="Via Morgari"></xs:enumeration>
12    </xs:restriction>
13  </xs:simpleType>
14 </xs:element>
```



## Capitolo 6

# Testing e Validazione

In questo capitolo vengono presentati i risultati dell'applicazione del tool su dati dell'anno accademico corrente e del seguente.

### 6.1 Simulazioni su dati a.a 2022/2023

In una prima fase di tesi, in cui l'obbiettivo era lo sviluppo delle nuove funzioni, si sono rivelati decisamente utili i dati consolidati del precedente lavoro. I dati inizialmente a disposizione comprendevano:

- la descrizione degli Insegnamenti in formato xml.<sup>1</sup>.
- il database con le informazioni più aggiornate fornite dal GOF.
- i risultati del Jotform.

Il dataset presente, infatti, essendo molto simile a quello che ci si aspettava per l'anno seguente ha permesso di effettuare simulazioni utili al fine del lavoro. Tramite simulazione è stato possibile validare le nuove funzionalità introdotte, studiarne il comportamento in associazione ai vincoli preesistenti e infine determinare il valore delle penalità più adatto per il modello.

---

<sup>1</sup>Dati fondamentali per poter eseguire una vera e propria simulazione

### 6.1.1 Prestazioni

Le simulazioni sono state effettuate sia in ambiente locale<sup>2</sup> che in una VM fornita dall'Area IT del Politecnico di Torino, con 16 cores e 64 GB di RAM. Una volta assodate le modifiche, le performance in entrambe le macchine sono state molto soddisfacenti. Si riusciva in poche ore, a volte anche minuti a determinare l'esistenza di una soluzione. In queste simulazioni era però stato utilizzato il file precedente contenente il numero di aule disponibili, il quale non considerava l'esatto numero di aule disponibili, ma una sua approssimazione. Una volta generato il file corretto delle aule le performance sono scese drasticamente. Si notava quindi già in questa fase una scarsità di aule.

## 6.2 Simulazioni su dati a.a 2023/2024

Le simulazioni del nuovo anno accademico sono state realizzate dopo un lungo periodo di raccolta dati. Il processo non è semplice in quanto si richiede la partecipazione di tutti i docenti. Bisogna infatti sia ottenere i dati del Jotform per definire le nuove indisponibilità, che fare un'intervista per ottenere le informazioni sulla struttura dei corsi, ad esempio quali docenti vi partecipano e come eventualmente sono divisi i laboratori. Per le nuove simulazioni è stato inoltre reso disponibile un nuovo nodo di calcolo, molto più grande del precedente, con 64 core e 256 GB di RAM. La disponibilità di questa potenza di calcolo, rispetto alle precedenti ha permesso l'introduzione di ulteriori vincoli non considerati nella fase finale dello scorso anno. Di seguito viene elencato tutto ciò che è stato preso in considerazione in più rispetto alle simulazioni precedenti.

Lo schema XSD per la rappresentazione XML delle aule è stato modificato per permettere di specificare l'esatto numero di aule per giorno e fascia oraria, diversamente dalla precedente soluzione che teneva in considerazione un numero fisso di aule per l'intera settimana.

```

1 <GruppoLocali>
2   <Nome>Aule piccole</Nome>
3   <Locale>Aula</Locale>
4   <Capienza>Piccola</Capienza>
5   <PreseElettriche>Si</PreseElettriche>
6   <Quantita>14</Quantita>
7   <Da>
8     <Giorno>Lun</Giorno>
9     <FasciaOraria>8.30-10.00</FasciaOraria>
10  </Da>
11  <A>
12    <Giorno>Sab</Giorno>
13    <FasciaOraria>17.30-19.00</FasciaOraria>

```

<sup>2</sup>Un computer portatile con un processore Intel Core I7 10th gen a 2.59GHz (6 core e 12 processori logici) e 16 GB di RAM

```
14 </A>
15
```

Listing 6.1: Esempio xml Aule (Old Style)

```
1 <GruppoLocaliNew>
2   <Nome>Aule piccole</Nome>
3   <Locale>Aula</Locale>
4   <Capienza>Piccola</Capienza>
5   <Disponibilita>
6     <Giorno>Lun</Giorno>
7     <Quantita_Fascia>12,8,6,12,0,6,17</Quantita_Fascia>
8   </Disponibilita>
9   <Disponibilita>
10    <Giorno>Mar</Giorno>
11    <Quantita_Fascia>11,3,1,6,0,9,12</Quantita_Fascia>
12  </Disponibilita>
13  <Disponibilita>
14    <Giorno>Mer</Giorno>
15    <Quantita_Fascia>9,6,6,4,9,3,5</Quantita_Fascia>
16  </Disponibilita>
17  <Disponibilita>
18    <Giorno>Gio</Giorno>
19    <Quantita_Fascia>6,0,1,7,0,10,8</Quantita_Fascia>
20  </Disponibilita>
21  <Disponibilita>
22    <Giorno>Ven</Giorno>
23    <Quantita_Fascia>9,6,6,4,10,3,3</Quantita_Fascia>
24  </Disponibilita>
25 </GruppoLocaliNew>
26
```

Listing 6.2: Esempio xml Aule (New Style) Anno precedente + Libere

Maggiori informazioni presenti nel jotform sono state utilizzate, sia per docenti titolari che non. Rispetto alla precedente simulazione molti più constraint sono stati presi in considerazione. In particolare:

Dal jotform sono state estratte preferenze per la gestione del carico di lavoro. Chiedendo ai docenti il numero massimo di slot che fossero disposti a svolgere al giorno e l'eventuale durata della pausa desiderata tra slot di lezioni. Attraverso questa richiesta si sono estratti dati necessari per generare i *maxSlotPerDay* e per *distanzaMinima*. Attraverso invece una tabella settimanale sono state chieste le varie indisponibilità e preferenze di orario da cui sono state estratte le *preferenze*.

```
1 "Lamberti Fabrizio": 4,
2   "Lavagno Luciano": 4,
3   "Lioy Antonio": 4,
4   "Maffiodo Daniela": 2,
5   "Magli Enrico": 3,
```

```
6  "Maio Ivano Adolfo": 2,  
7  "Malan Stefano Alberto": 2,  
8  "Malnati Giovanni": 4,  
9  "Marchetto Guido": 4,  
10 "Martina Maurizio": 2,  
11 "Mastrantonio Gianluca": 3,  
12 "Mazali Tatiana": 3,  
13 "Milanesio Daniele": 4,  
14 "Monaci Sara": 3,  
15 "Monge Roffarello Alberto": 2,  
16 "Montuschi Paolo": 2,  
17
```

**Listing 6.3:** Estratto docenti.json maxSlotperDay

```
1  "distanzaMinima": {  
2    "Crovetti Paolo Stefano": "1_LV6",  
3    "Dall'Asta Luca": "1_LV6",  
4    "De Martino Andrea": "1_LV6",  
5    "De Russis Luigi": "1_LV6",  
6    "Demarchi Danilo": "1_LV6",  
7    "Di Fabrizio Enzo Mario": "1_LV6",  
8    "Donati Guerrieri Simona": "1_LV6",  
9    "Dovis Fabio": "1_LV6",  
10   "Falletta Silvia": "1_LV6",  
11   "Farinetti Laura": "1_LV6",  
12   "Fraterrigo Garofalo Silvia": "1_LV6",  
13   "Galleani Lorenzo": "1_LV6"  
14 }  
15
```

**Listing 6.4:** Estratto docenti.json distanzaMinima

```
1  "favorevole": {  
2    "Lun_13.00-14.30_LV4": "ugualeA",  
3    "Mar_10.00-11.30_LV4": "ugualeA",  
4    "Mar_11.30-13.00_LV4": "ugualeA",  
5    "Mar_13.00-14.30_LV4": "ugualeA",  
6    "Mer_10.00-11.30_LV4": "ugualeA",  
7    "Mer_11.30-13.00_LV4": "ugualeA",  
8    "Mer_13.00-14.30_LV4": "ugualeA",  
9    "Gio_11.30-13.00_LV4": "ugualeA",  
10   "Gio_13.00-14.30_LV4": "ugualeA",  
11   "Ven_10.00-11.30_LV4": "ugualeA",  
12   "Ven_11.30-13.00_LV4": "ugualeA",  
13   "Ven_13.00-14.30_LV4": "ugualeA"  
14 },  
15 "contrario": {  
16   "Lun_8.30-10.00_LVH": "ugualeA",  
17   "Lun_16.00-17.30_LV4": "ugualeA",  
18   "Mar_8.30-10.00_LVH": "ugualeA",  
19   "Mar_16.00-17.30_LV4": "ugualeA",
```

```
20     "Mer_8.30-10.00_LVH": "ugualeA",
21     "Mer_16.00-17.30_LV4": "ugualeA",
22     "Gio_8.30-10.00_LVH": "ugualeA",
23     "Gio_16.00-17.30_LV4": "ugualeA",
24     "Ven_8.30-10.00_LVH": "ugualeA",
25     "Ven_16.00-17.30_LV4": "ugualeA"
26 }
```

**Listing 6.5:** Estratto docenti.json preferenze

A seguito dell'introduzione dei nuovi vincoli sono sorte due principali problematiche, il software infatti non era in grado di trovare alcuna soluzioni in tempi ragionevoli. La simulazione più lunga effettuata senza riuscire a trovare soluzione è stata di ben 72 ore, in quel lasso di tempo il software non era riuscito a determinare l'esistenza di una effettiva soluzione. Dopo attente analisi si è scoperto che le problematiche che portavano a questo risultato erano: la quantità di aule limitata e la quantità di slot massimi giornalieri richiesti dai docenti.

### Problematica Aule

Il problema delle aule è un problema ristretto alle sole aule ordinarie, cioè non riguarda le disponibilità dei laboratori. Il problema viene facilmente esposto una volta analizzate le statistiche circa gli slot aula necessari e gli slot aula disponibili. Si riportano di seguito alcuni dati interessanti.

- Sono stati considerati 295 file Insegnamento
- Si hanno 1.084 Slot da 1.5h da allocare di cui 821 da allocare il Aule Ordinarie
- Le Aule a disposizione sono state calcolate in riferimento a quelle assegnate l'anno precedente Figura 6.1 e aggiungendo la disponibilità delle aule libere Figura 6.2. Da questi dati è emerso una disponibilità di 24 aule per Fascia Oraria<sup>3</sup>. Da un semplice calcolo, sapendo che si considerano 7 fasce orarie per 5 giorni la settimana, si trova che ci sono 840 slot aula a settimana.
- Dalle statistiche emergono anche dati preoccupanti sulla capienza 6.2. In riferimento ai dati delle prime simulazioni effettuate, si nota che il numero di slot per le aule di Piccole e Medio-Grandi dimensioni supera di 3 volte la disponibilità reale, le aule Medie superano di poco la disponibilità e infine più della metà delle aule Grandi sono disponibili. La situazione migliora aggiungendo le aule Libere, ma il problema persiste in quanto le aule di capienza Piccola e MedioGrande non sono sufficienti per gli slot da contenere come nel Listing 6.2.

<sup>3</sup>Non è tenuta in considerazione la capienza

- Dai dati forniti si nota, inoltre, uno squilibrio nella disponibilità stessa tra le diverse fasce orarie. Ad esempio si hanno, in particolare, molte aule nelle fasce 17.30-19.00, di per se questa comporta già un problema, infatti, se anche fosse infinita la disponibilità in quella fascia a causa delle sovrapposizioni si renderebbe comunque difficile trovare soluzione. Si aggiunge a ciò che in media le fasce che hanno più aule sono anche le fasce in cui ci sono più Indisponibilità.

**Piano di assegnazione spadi didattici (AULE E LAIB) didattica ed esercitazioni A.A. 22-23 1° pd**

Estrazione dati da template (fonte procedura orari).

Conteggio Aule	Lunedì					totale	Martedì					totale	Mercoledì					totale	Giovedì					totale	Venerdì					totale	aule utilizzate	lezioni totali
	L	ML	MS	S	XS		L	ML	MS	S	XS		L	ML	MS	S	XS		L	ML	MS	S	XS		L	ML	MS	S	XS			
08:30/10:00	3		1	2		6	5		2	1	8	1	1	2	2	1	7	2		5		7	2		2	1	3		2	27	30	
08:30/11:30	3			1		4	1		3	1	6	3		1		4	1		1		2	1		3				4	19	20		
10:00/11:30			2	1		3	5		1		6			2		2			2		2		1					1	13	14		
10:00/13:00	2		4	1		7	4		1		5	2		1	1	3	3		1	1	5	3			2		5	19	25			
11:30/13:00		1	1	1		3	1		1		2		1	3	1	5	1		1	1	2	1					1	13	13			
11:30/14:30			1			1	1		1		2	1		1		2	2			2		2	1	1			2	8	9			
13:00/14:30	1	1	1	1		3	1		1	1	3	3	1	1	2	7	2		4	1	7	2	1	2		5	20	25				
13:00/16:00	2	2	1			5	6		4	1	11	1		3		4	3		1	2	6	1	1	3	1		6	27	32			
13:00/17:30											1					1												1	1	1		
14:30/16:00	2	1	1			4	1		1		2	3		2		5	3		1	3	1	1	1			3	18	22				
14:30/17:30			1			1	3		1	3	8	1		1		2	1		1	2	4	2	2	1		5	18	20				
16:00/17:30	3	1				4	3		1		4			2		2	3		1		4	1		2		3	13	17				
16:00/19:00	6	1	1			8	3		2		5	3		6		9	2		1	2	5			2		2	19	29				
17:30/19:00	1		1	1		3	1		1	1	3			1		1											7	7	7			
numero aule utilizzate	10	3	11	7	1	32	15	1	12	5	1	34	9	2	16	4	1	32	10	2	12	5		29	8	4	12	5	29	78		
lezioni totali	23	5	14	9	1	52	35	2	22	5	1	65	19	3	25	5	2	54	23	2	22	7	0	54	12	6	15	6	0	39	264	

Figure 6.1: Statistiche Aule Assegnate anno precedente

Aule	Posti	lunedì					martedì					mercoledì					giovedì					venerdì									
		8:30-10:00	10:00-11:30	11:30-13:00	13:00-14:30	14:30-16:00	8:30-10:00	10:00-11:30	11:30-13:00	13:00-14:30	14:30-16:00	8:30-10:00	10:00-11:30	11:30-13:00	13:00-14:30	14:30-16:00	8:30-10:00	10:00-11:30	11:30-13:00	13:00-14:30	14:30-16:00	8:30-10:00	10:00-11:30	11:30-13:00	13:00-14:30	14:30-16:00					
Centrale 23	32																														
Centrale 25	32	1																													
Centrale 21A	47																														
Centrale 9B	47																														
Centrale 11B	47																														
Centrale 6C	25																														
Centrale 6D	25																														
Centrale 11	101																														
Centrale 13	100																														
Centrale 13A	51																														
Centrale 13B	51																														
Centrale 15	100																														
Centrale 15A	97																														
Centrale 17A	96																														
Centrale 19A	87																														
Centrale 5B	51																														
Centrale 7B	97																														
Centrale 8D	94																														
Centrale 9S	72																														
Centrale 11S	72																														
Centrale 7	150																														
Centrale 10	140																														
Centrale 2C	121																														
Centrale 4C	120																														
Centrale 10C	118																														
Centrale 2D	123																														
Centrale 3D	117																														

Figure 6.2: Aule libere aggiunte al modello

```

1 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  Piccola:
2 Slot (lezioni) da assegnare: 262
3 Slot (blocchi da 1.5h) da assegnare: 399
4 Soglia di occupazione (media): 3.53
    
```

```

5 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  Media:
6 Slot (lezioni) da assegnare: 148
7 Slot (blocchi da 1.5h) da assegnare: 221
8 Soglia di occupazione (media): 0.96
9 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  MedioGrande:
10 Slot (lezioni) da assegnare: 77
11 Slot (blocchi da 1.5h) da assegnare: 122
12 Soglia di occupazione (media): 3.70
13 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  Grande:
14 Slot (lezioni) da assegnare: 52
15 Slot (blocchi da 1.5h) da assegnare: 79
16 Soglia di occupazione (media): 0.47

```

**Listing 6.6:** Statistiche Capienza Aule Solo Anno precedente

```

1 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  Piccola:
2 Slot (lezioni) da assegnare: 262
3 Slot (blocchi da 1.5h) da assegnare: 399
4 Soglia di occupazione (media): 1.83
5 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  Media:
6 Slot (lezioni) da assegnare: 148
7 Slot (blocchi da 1.5h) da assegnare: 221
8 Soglia di occupazione (media): 0.64
9 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  MedioGrande:
10 Slot (lezioni) da assegnare: 77
11 Slot (blocchi da 1.5h) da assegnare: 122
12 Soglia di occupazione (media): 1.65
13 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  Grande:
14 Slot (lezioni) da assegnare: 52
15 Slot (blocchi da 1.5h) da assegnare: 79
16 Soglia di occupazione (media): 0.34

```

**Listing 6.7:** Statistiche Capienza Aule Anno precedente + Libere

### 6.3 Problema dell'insufficienza di Aule

Per quanto riguarda gli Slot Aula, come emerso dalle statistiche, il modello si ritrova ad averne pochissime in più rispetto agli effettivi slot da Allocare, 19 in più per la precisione. Se il problema consistesse soltanto nell'allocare gli slot nelle aule, basandoci sui numeri, questo non sarebbe un gran problema. Tuttavia bisogna ricordarsi che il modello comprende un enorme elevato di hard constraint che limitano parecchio le opzioni di scelta

per l'allocazione. Pertanto, se anche numericamente il problema può essere risolto, questo non risulta vero nella pratica una volta presi in considerazione i restanti vincoli.

Da queste considerazioni, si può in fretta dedurre che, è normale che il software non riesca a trovare soluzioni in tempi ragionevoli, anzi, potrebbe lavorare settimane prima di rivelare che non esiste una soluzione valida<sup>4</sup>.

### 6.3.1 Soluzione

Per rispondere a questo problema si è deciso di aumentare il numero delle aule disponibili rispetto a quanto inizialmente negoziato, dopo averlo concordato con la commissione spazi, in modo tale da permettere al software di avere più libertà di scelta e quindi trovare soluzione in tempi brevi. La soluzione adottata ha comunque seguito un'euristica di base, evitando di eccedere troppo con le approssimazioni.

E stato innanzitutto considerato che esistono certe fasce Orarie in cui la situazione delle aule è più critica, tra queste troviamo:

- 10.00-11.30
- 11.30-13.00
- 14.30-16.00

Per questo motivo, in tali fasce si è deciso a priori che si sarebbero aggiunte molte meno aule fittizie rispetto alle altre fasce. Attraverso un approccio *trial and error*, con simulazioni consecutive si è andati alla ricerca di quei valori minimi che permettessero la generazione di una soluzione in qualche minuto.

Dai test effettuati si è giunti alla conclusione che i valori minimi fossero: 8 aule in più per tutte le fasce non critiche e 4 per quelle critiche. Per una visualizzazione migliore si confrontino i Listing 6.2 e 6.3.1 e le statistiche in Listing 6.2 e 6.3.1.

```
1 <GruppoLocaliNew>
2   <Nome>Aule piccole</Nome>
3   <Locale>Aula</Locale>
4   <Capienza>Piccola</Capienza>
5   <Disponibilita>
6     <Giorno>Lun</Giorno>
7     <Quantita_Fascia>20,12,10,20,4,14,25</Quantita_Fascia>
```

---

<sup>4</sup>Infatti in questo caso particolare, potendo rientrare numericamente nel problema, il software necessita di esplorare tutte le combinazioni possibili prima di poter determinare che il problema non ha soluzione.



```

8     </Disponibilita>
9     <Disponibilita>
10        <Giorno>Mar</Giorno>
11        <Quantita_Fascia>19,7,5,14,4,17,20</Quantita_Fascia>
12    </Disponibilita>
13    <Disponibilita>
14        <Giorno>Mer</Giorno>
15        <Quantita_Fascia>17,10,10,12,13,11,13</Quantita_Fascia>
16    </Disponibilita>
17    <Disponibilita>
18        <Giorno>Gio</Giorno>
19        <Quantita_Fascia>14,4,5,15,4,18,16</Quantita_Fascia>
20    </Disponibilita>
21    <Disponibilita>
22        <Giorno>Ven</Giorno>
23        <Quantita_Fascia>17,10,10,12,14,11,11</Quantita_Fascia>
24    </Disponibilita>
25 </GruppoLocaliNew>

```

Listing 6.8: Estratto Aule Simulazione finale

```

1 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  Piccola:
2 Slot (lezioni) da assegnare: 262
3 Slot (blocchi da 1.5h) da assegnare: 399
4 Soglia di occupazione (media): 0.91
5 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  Media:
6 Slot (lezioni) da assegnare: 148
7 Slot (blocchi da 1.5h) da assegnare: 221
8 Soglia di occupazione (media): 0.39
9 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  MedioGrande:
10 Slot (lezioni) da assegnare: 77
11 Slot (blocchi da 1.5h) da assegnare: 122
12 Soglia di occupazione (media): 0.41
13 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
  Grande:
14 Slot (lezioni) da assegnare: 52
15 Slot (blocchi da 1.5h) da assegnare: 79
16 Soglia di occupazione (media): 0.17

```

Listing 6.9: Statistiche Aule Anno precedente + Libere + fittizie

Dalle ultime statistiche nonostante tutte le soglie sono rispettate, c'è però uno squilibrio nell'occupazione delle aule. Le aule piccole sono praticamente tutte piene mentre delle aule grandi veramente poche sono occupate. In questo caso il problema potrebbe persistere, e la disponibilità delle aule piccole potrebbe non essere sufficiente a rispettare tutti gli altri vincoli.

Considerata la grande quantità di aule libere con capienza più grande, avrebbe allora senso

che questa siano sfruttate anche per insegnamenti con quantità di studenti minori rispetto alla capienza. Per far sì che questa opzione venga attuata dal software basterebbe utilizzare la funzione, implementata già nella tesi precedente, riguardo l'estensione sulla capienza delle aule.<sup>5</sup> Nonostante ciò dai test effettuati non si sono ottenuti vantaggi nell'utilizzare questa funzione con livelli alti di estensione, probabilmente perché il lavoro di calcolo veniva appesantito ulteriormente.

Come escamotage a questo dilemma, si è ottenuto lo stesso risultato modificando i range di capienza delle Aule. In un primo momento le capienze delle aule erano così definite:

```
1 if numStudenti == 0:
2     # per gli Insegnamenti di cui non conosco il numero di iscritti
3     return CapienzaLocale.Media
4 if numStudenti <= 100:
5     return CapienzaLocale.Piccola
6 if numStudenti <= 150:
7     return CapienzaLocale.Media
8 if numStudenti <= 200:
9     return CapienzaLocale.MedioGrande
10 return CapienzaLocale.Grande
```

Da questo si nota che, secondo le informazioni date al software, tutti gli Insegnamenti con meno di 100 studenti sono stati considerati da allocare in aule piccole, e che questi Insegnamenti comprendano 399 slot da assegnare. Se invece si stabilissero le soglie in maniera differente troveremmo un numero diverso di slot. Per questo motivo, attraverso altre simulazioni, sono state imposte nuove soglie col solo scopo di bilanciare le soglie di occupazione che si erano ottenute prima di questa modifica.

Le soglie considerate al fine di produrre il risultato finale riportate di seguito.

```
1 if numStudenti == 0:
2     # per gli Insegnamenti di cui non conosco il numero di iscritti
3     return CapienzaLocale.Media
4 if numStudenti <= 50:
5     return CapienzaLocale.Piccola
6 if numStudenti <= 100:
7     return CapienzaLocale.Media
8 if numStudenti <= 140:
9     return CapienzaLocale.MedioGrande
10 return CapienzaLocale.Grande
```

**Listing 6.10:** Soglie Ottimali Aule

---

<sup>5</sup>Funzione che permette al programma di utilizzare aule di x livelli più grandi, in base al livello di estensione

```
1 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
   Piccola:
2 Slot (lezioni) da assegnare: 108
3 Slot (blocchi da 1.5h) da assegnare: 161
4 Soglia di occupazione (media): 0.37
5 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
   Media:
6 Slot (lezioni) da assegnare: 182
7 Slot (blocchi da 1.5h) da assegnare: 285
8 Soglia di occupazione (media): 0.50
9 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
   MedioGrande:
10 Slot (lezioni) da assegnare: 93
11 Slot (blocchi da 1.5h) da assegnare: 133
12 Soglia di occupazione (media): 0.45
13 LocaliHandler.createListsLocaliOfSlot(): statistiche Aule di capienza
   Grande:
14 Slot (lezioni) da assegnare: 156
15 Slot (blocchi da 1.5h) da assegnare: 242
16 Soglia di occupazione (media): 0.54
```

**Listing 6.11:** Statistiche Aule con Soglie Ottimali

Grazie a quest'ultimo intervento, è stato possibile trovare una soluzione temporanea al problema delle aule. Temporanea in quanto, con questo tipo di modellazione sono stati approssimati i dati reali, per cui si è reso possibile trovare una soluzione abbastanza accettabile con la consapevolezza che a posteriori, una volta scelta la soluzione del software questa debba essere modificata attraverso interazioni con i responsabili della Gestione Aule.

## 6.4 Problematica Max Slot Docenti

Una volta ottenuto il file jotform compilato ed estratte le informazioni da esso, si è generato il json finale contenente la descrizione dei vincoli per i Docenti, tra cui il massimo numero di slot al giorno e la distanza minima tra uno slot e l'altro. Innanzitutto non è stato possibile considerare la distanza minima come Hard constraint, poiché sono state svolte diverse prove senza la possibilità di trovare soluzioni. Per quanto riguarda, invece, il numero massimo di slot giornalieri, questi potevano essere considerati hard constraint, con la necessità di alcuni accorgimenti. Dall'analisi dei log prodotti dal software, si era infatti notato che alcuni docenti avessero specificato un valore di slot massimi giornalieri non sufficiente a coprire settimanalmente gli slot in cui essi erano presenti, conseguenza di ciò è stata la non risolubilità del modello.

Con un primo approccio è stato fatto in modo di riuscire a capire quali fossero i Docenti più problematici in tal senso.

### 6.4.1 Soluzione

Si introduce un concetto preliminare. Si definisce *criticità* di un Docente il valore calcolato come il numero di slot totali in cui è presente il docente diviso il numero di slot massimi giornalieri moltiplicato per 5<sup>6</sup>.

$$\text{criticità} = \frac{nSlotSettimanali}{nSlotMaxPerDay * N\_SLOT\_SETTIMANA} \quad (6.1)$$

In questo modo è chiaro che criticità maggiori di 1 renderanno impossibile la risoluzione del modello e valori molto vicini ad 1 lo renderà molto difficile<sup>7</sup>.

Attraverso modifiche del codice si è definita una soglia di criticità per cui se questa venisse superata da un qualche docente, allora il numero massimo di slot giornalieri verrebbe aumentato di uno in fase di esecuzione.

Con una serie di simulazioni in cui si andava a modificare il valore di soglia, si è scoperto che la soglia per cui si passava da un modello non soddisfacibile ad uno con soluzione, era una soglia pari a 0,8. Sapendo che se fosse stato modificato il valore di *maxSlotPerDay* per i docenti che superavano tale soglia, il problema si sarebbe risolto, allora si è optati per modificare i valori di scelta per tali docenti avendo cura di avvisarli, spiegando loro la necessità di tale operazione.

```

1 Baldi Mario con criticita': 0.5 a da allocare in: 10 slot
2 Benso Alfredo con criticita': 0.45 a da allocare in: 9 slot
3 Montuschi Paolo con criticita': 0.5 a da allocare in: 5 slot
4 Cagliero Luca con criticita': 0.55 a da allocare in: 11 slot
5 Garello Roberto con criticita': 0.53 a da allocare in: 8 slot
6 Vassio Luca con criticita': 0.46 a da allocare in: 7 slot
7 Cocuzza Matteo con criticita': 0.6 a da allocare in: 6 slot
8 Di Carlo Stefano con criticita': 0.6 a da allocare in: 6 slot
9 Lioy Antonio con criticita': 0.45 a da allocare in: 9 slot
10 De Russis Luigi con criticita': 0.66 a da allocare in: 10 slot
11 Zunino Claudio con criticita': 0.6 a da allocare in: 6 slot
12 Giorgis Fabrizio con criticita': 0.53 a da allocare in: 8 slot
13 Servetti Antonio con criticita': 0.73 a da allocare in: 11 slot

```

**Listing 6.12:** Estratto log criticità Docenti

<sup>6</sup>I giorni di lezione in una settimana

<sup>7</sup>Come visto per le aule, il fatto che il modello matematicamente sia possibile non da assoluta certezza che vi sia soluzione. Si deve infatti tenere a mente che la presenza di tutti gli altri hard constraint potrebbe rimuovere delle opzioni per cui il problema diventa irrisolvibile.

## Capitolo 7

# Conclusioni

La pianificazione dell'orario si è rivelata una sfida complessa a causa della varietà di vincoli presenti. Data la numerosità delle entità coinvolte, è diventato evidente come tali vincoli possano beneficiare di ulteriori miglioramenti per ottenere una rappresentazione sempre più accurata della realtà. Allo stesso tempo, l'introduzione di nuovi vincoli può complicare la risoluzione di un modello precedentemente funzionante.

In alcuni casi, è stato possibile risolvere queste situazioni facendo affidamento sull'intuito e rispettando la realtà su cui si basa il modello. Tuttavia, a volte è stato necessario fare dei compromessi, cioè introdurre approssimazioni nel modello al di là delle reali possibilità.

### 7.1 Valutazioni conclusive

La soluzione basata sulle *correlazioni* si è rivelata un ottimo espediente per risolvere completamente il problema della Sovrapposizione degli Insegnamenti. Questo approccio, infatti, non presenta alcuna approssimazione<sup>1</sup> e consente diversi metodi per un bilanciamento adeguato dei vincoli generati. I metodi per la definizione dei pesi sono stati applicati per la prima volta in questa tesi, di conseguenza, attraverso un'analisi dei risultati ottenuti, potrebbe essere plausibile implementare nuovi metodi o modificare quelli esistenti per generare valori ancora migliori.

La soluzione dei percorsi, a differenza della precedente, potrebbe non rivelarsi sufficientemente matura, sono stati infatti riscontrati diverse situazioni in cui la tale soluzione si è rilevata meno efficace.

---

<sup>1</sup>Permette di tenere in considerazione tutte le possibili sovrapposizione

- Per certi Insegnamenti non si avevano a disposizione i dati riguardanti le iscrizioni degli studenti, per cui non è stato possibile includerli in alcun percorso.
- Allo stesso modo, la mancanza di altri dati che non ha permesso la risoluzione di codici insegnamento in ID\_INC, ha fatto sì che anche in questo caso si trascurassero altri insegnamenti. Per quanto ispezionato, in questi casi il codice insegnamento non era presente nel file del GOF e si è notato che si trattava di Insegnamenti erogati a tutto l'ateneo come ad esempio le *Challenge*.
- Si è provato ad includere gli insegnamenti per cui mancavano i dati, come descritto sopra, cioè si inserivano gli insegnamenti nei percorsi in cui si sarebbero dovuti trovare, stimando il valore di intersezione studenti sulla base degli dati a disposizione. Con questo approccio il numero di *percorsi* generati non era sostenibile (quasi 600). Il software infatti impiegava diversi minuti solo per la generazione dei vincoli. Conseguentemente, l'esecuzione della ricerca di soluzione, a sua volta, aveva delle prestazioni molto peggiori.

Con più tempo a disposizione, potrebbe essere possibile recuperare i dati mancanti o determinare, una volta verificato che tale mancanza è giustificata, se sia corretto non considerare determinati insegnamenti/percorsi.

Nello stato conclusivo della tesi, i percorsi riportati nel file finale sono quelli che includono solo insegnamenti di cui sono disponibili tutti i dati necessari. I nodi senza dati vengono rimossi, ma si mantiene comunque il sottopercorso. Questo approccio consente di trascurare il minor numero possibile di informazioni e di ottenere un numero adeguato di percorsi, in modo che il software non incontri problemi durante l'esecuzione.

## 7.2 Sviluppi futuri

Vengono in questa sezione riportati alcuni spunti di riflessione per eventuali miglioramenti futuri.

### 7.2.1 Penalità Docenti

Si è notato, attraverso analisi delle varie soluzioni anche provvisorie, che il software, soprattutto in simulazioni con esecuzioni poco esaustive, ha avuto tendenza a migliorare tutte le penalità riguardo alcuni Docenti, penalizzando totalmente altri. Si vuole, per un futuro fare in modo che le penalità siano distribuite equamente su tutti i docenti. Matematicamente le somme di penalità e bonus di ciascun Docente non dovrebbero distanziarsi tra di loro. Ad esempio valorizzando maggiormente, a run time, il soddisfacimento di penalità o bonus che aiutino a mantenere una minore distanza media tra i valori o penalizzando in maniera crescente in base al numero di scelte non gradite ai docenti.

### 7.2.2 Insegnamenti a Scelta e Fasce Orarie Protette

Per il secondo ed il terzo anno di lauree triennali esistono delle fasce orari protette secondo cui soltanto gli Insegnamenti a scelta possono finire, dove, quindi, tutti gli Insegnamenti Obbligatori non posso andare. Nella attuale soluzione è stata fatta una query al Db per identificare gli ID\_INC degli insegnamenti Obbligatori per poi inserirli manualmente nel codice al fine di far generare i vincoli necessari. Sarebbe utile per uno sviluppo futuro automatizzare la procedura dell'attuale problema ed eventualmente pensare ad un'estensione che possa includere l'aggiunta di nuovi casi simili.

### 7.2.3 Rifinire le funzioni per il calcolo delle penalità

Come affrontato in altri capitoli, per una definizione più puntuale dei pesi dei *soft constraint*, in particolare per le sovrapposizioni di insegnamento e per la gestione degli studenti, sono state utilizzate delle funzioni che sfruttavano il numero di studenti in comune tra Insegnamenti. Tali funzioni hanno però una dipendenza lineare rispetto l'intersezione, pertanto, si suggerisce una rivisitazione delle funzioni sfruttando i risultati qui ottenuti, valutando possibili soluzioni con dipendenza asintotica.

### 7.2.4 Aule

Rimane il compito di riuscire a trovare una soluzione che riesca a tener conto della distanza tra Aule, ovvero il requisito per cui si vorrebbero minimizzare gli spostamenti. Avendo già visto come le aule, però, siano già un grande problema, la quantità troppo stringente costringe a delle approssimazione e approssimando il numero di aule verrebbe anche meno qualsiasi soluzione per la minimizzazione degli spostamenti. Pertanto, si conclude che, prima di poter affrontare il nuovo obiettivo si debba risolvere l'approssimazione precedente, che tra le altre cose, non è un problema di malfunzionamento del programma, ma una limitazione strutturale intrinseca nella realtà stessa.

### 7.2.5 Estensione API web

L'API web per la visualizzazione delle Sovrapposizioni di Insegnamenti, permette di mostrare gli slot in cui il docente risulta indisponibile sulla base del solo orario generato, ad esempio dovuta ad altra lezione in tale fascia oraria. Come sviluppo futuro potrebbe essere creata una funzione che, sulla base del json docenti, inserisca anche le indisponibilità richieste.

# Bibliografia

- [1] Francesco Lasagno. *Modellazione e ottimizzazione della pianificazione della didattica*. 2021/2022.
- [2] Paolo Serafini. «Programmazione lineare intera Metodi risolutivi». ita. In: *Ricerca Operativa*. Germany: Springer, 2009. ISBN: 884700845X.
- [3] Luigi De Giovanni and Giacomo Zambelli. *Introduzione al metodo del semplice*. Tech. rep. 2022. URL: <https://www.math.unipd.it/~luigi/courses/rodid/m01.simplex.pdf>.
- [4] Gearhart Jared L., Adair Kristin L., Detry Richard J., Durfee Justin D., Jones Katherine A., and Martin Nathaniel. *Comparison of Open-Source Linear Programming Solvers*. Sandia National Laboratories Internal Report. 2013.
- [5] IBM. *Tutorial: Linear Programming, (CPLEX Part 1)*. 2022. URL: [https://ibmdecisionoptimization.github.io/tutorials/html/Linear\\_Programming.html](https://ibmdecisionoptimization.github.io/tutorials/html/Linear_Programming.html).
- [6] Wikipedia. *Web scraping — Wikipedia, L'enciclopedia libera*. [Online; in data 11-luglio-2023]. 2023. URL: [https://it.wikipedia.org/w/index.php?title=Web\\_scraping&oldid=134058255](https://it.wikipedia.org/w/index.php?title=Web_scraping&oldid=134058255).