

Ottimizzazione dei Garanti accademici

Colli Simone¹ and Merenda Saverio Mattia¹

¹ `simone.colli@studenti.unipr.it`

² `saveriomattia.merenda@studenti.unipr.it`

December 9, 2024

(MM:

- Modificare il nome del progetto su github e nel container docker da `pd-project` a `ottimizzazione-garanti-accademici`
- Scrivere le scritte dinamiche in italiano, ad esempio Figure 1, ecc

)

Abstract

Questo lavoro presenta l'analisi e l'implementazione di un sistema automatizzato per l'assegnazione dei garanti ai corsi universitari, in conformità ai requisiti ministeriali. L'obiettivo principale è garantire che ogni corso soddisfi i vincoli minimi di docenza, rispettando le regole di distribuzione tra diverse categorie di docenti e ottimizzando l'uso delle risorse disponibili.

Utilizzando la programmazione logica con Answer Set Programming (ASP) (Lifschitz, 2002), abbiamo modellato il problema attraverso fatti, regole e vincoli derivati dai dati ministeriali e universitari. Abbiamo implementato una serie di vincoli per rispettare i minimi richiesti di docenti per corso, evitando sovrapposizioni improprie tra gli incarichi dei docenti e considerando scenari realistici in cui un docente può assumere più ruoli parziali.

L'approccio è stato testato su un dataset reale contenente informazioni su corsi, SSD (Settori Scientifico Disciplinari) e docenti dell'Università degli Studi di Parma. I risultati dimostrano come il sistema possa trovare configurazioni ottimali che soddisfano i requisiti, massimizzando l'efficienza e mantenendo flessibilità nell'assegnazione dei docenti.

1 Introduzione

L'assegnazione dei garanti nei corsi universitari costituisce una questione fondamentale per la gestione ottimale delle risorse accademiche. Nell'ambito universitario, il *garante* è un docente responsabile di rappresentare e tutelare la qualità didattica di un corso, garantendo il rispetto dei requisiti disciplinari e istituzionali. I garanti possono appartenere a diverse categorie contrattuali: docenti a tempo indeterminato, docenti a tempo determinato e, in casi eccezionali, docenti a contratto.

La sfida principale consiste nel soddisfare i vincoli ministeriali relativi ai garanti, garantendo al contempo un'allocazione equilibrata e sostenibile delle risorse. Ogni corso deve essere supportato da un numero minimo di garanti, suddivisi tra le diverse fasce contrattuali, per assicurare un livello adeguato di competenza e rappresentatività. Inoltre, è indispensabile che almeno il 50% dei garanti afferisca al Settore Scientifico Disciplinare (SSD) caratterizzante del corso, al fine di garantire la coerenza tra l'offerta formativa e le competenze disciplinari.

Un'ulteriore complessità è rappresentata dall'impiego di docenti a contratto, il cui utilizzo deve essere limitato e subordinato alle sole situazioni in cui non sia possibile soddisfare i requisiti attraverso i docenti strutturati. La necessità di rispettare questi vincoli, combinata con la disponibilità limitata di personale e la necessità di bilanciare il carico di lavoro, rende questo problema una sfida organizzativa e computazionale significativa.

Un primo ostacolo affrontato in questo progetto riguarda la fase di pre-elaborazione dei dati forniti dall'università, descritta nella Sezione 2 di questo elaborato. I dati risultano disomogenei e incompleti, richiedendo una significativa pulizia e riorganizzazione prima di poter essere utilizzati efficacemente nel modello ASP. Questa fase ha richiesto lo sviluppo di strumenti dedicati per uniformare e validare i dati.

La Sezione 3 esplora il cuore del nostro approccio, ovvero la costruzione del modello ASP. Qui abbiamo implementato regole logiche per soddisfare i vincoli ministeriali e massimizzare l'efficienza dell'assegnazione dei garanti, tenendo conto delle limitazioni sulle fasce contrattuali e sull'impiego dei docenti a contratto.

Successivamente, nella Sezione 4, presentiamo i risultati ottenuti su un esempio ridotto, un test che ha permesso di validare il modello in un ambiente controllato e di analizzare la qualità delle soluzioni generate. Infine, la Sezione 5 descrive l'applicazione del modello su un dataset completo contenente tutti i corsi universitari, eseguendo un benchmark su larga scala per valutare la capacità del nostro approccio di risolvere problemi reali e complessi.

2 Preprocessing dei dati

2.1 Analisi preliminare

I dati analizzati, forniti dall'ufficio didattico dell'Università di Parma, erano organizzati in formato Excel con una struttura tabellare complessa, contenente numerosi campi. Per garantire un'analisi efficace e accurata, si è resa necessaria una fase di preprocessing, finalizzata a correggere eventuali incoerenze, migliorare la qualità complessiva del dataset e suddividerlo in sottoinsiemi più gestibili.

Il dataset fornito includeva informazioni relative ai docenti, ai ricercatori e alle coperture degli insegnamenti nei diversi corsi di laurea. Dopo un'attenta analisi preliminare, sono stati selezionati esclusivamente i campi considerati rilevanti per l'analisi. Nel file contenente i dati di docenti e ricercatori, sono state prese in considerazione solo le informazioni riguardanti la matricola, la fascia contrattuale e l'SSD. Per quanto riguarda il file sulle coperture, sono stati mantenuti i campi relativi alla matricola del docente, al codice del tipo di corso, al codice del di laurea e all'SSD.

Questa selezione dei dati è stata essenziale per ridurre la complessità e concentrarsi sugli elementi chiave necessari per costruire il modello ASP e verificare il rispetto dei vincoli ministeriali. Tutto questo è stato realizzato utilizzando diversi script in Python, supportati dai moduli `pandas`¹ e `openpyxl`², essenziali per la manipolazione e la gestione dei dataset.

La maggior parte delle operazioni di preprocessing ha riguardato il file contenente le coperture degli insegnamenti, articolandosi in diverse attività principali. In primo luogo, abbiamo gestito le righe contenenti una o più colonne vuote, eliminando quelle completamente prive di informazioni, in quanto non rilevanti per l'analisi. Le righe in cui mancavano i dati relativi al docente (i.e., matricola, nome e cognome) sono state invece isolate in un file separato. Questa scelta è stata necessaria per identificare e analizzare in modo specifico gli insegnamenti non ancora assegnati ad alcun docente.

Un ulteriore passo fondamentale ha riguardato la scomposizione del dataset originale, suddividendolo in sottoinsiemi più piccoli e gestibili. Questa operazione ha permesso di affrontare con maggiore precisione i singoli aspetti del problema, semplificando il successivo trattamento dei dati. Infine, sono stati identificati e gestiti in modo univoco e automatizzato i casi particolari, consentendo di creare un dataset uniforme e privo di anomalie, adatto all'analisi e all'implementazione del modello ASP.

Per migliorare ulteriormente la gestibilità dei dati, abbiamo proceduto con la creazione di sottoinsiemi minimali e distinti. Il primo passo è consistito nella separazione delle righe relative agli insegnamenti tenuti da docenti o ricercatori da quelle tenute dai docenti a contratto.

Questa suddivisione è stata effettuata utilizzando il file dei docenti fornito, sfruttando la colonna contenente la matricola come criterio discriminante. In particolare, le righe associate ai docenti a contratto sono state estratte e archiviate in un file separato, al fine di garantire un maggiore ordine e facilitare eventuali ispezioni manuali. Parallelamente, le righe relative ai docenti a tempo indeterminato e determinato sono state salvate anch'esse in un altro file, per gli stessi identici motivi.

Una volta completata la suddivisione, è stata apportata una modifica ai valori nella colonna relativa al codice del tipo di corso di laurea. In particolare, tutte le occorrenze del valore L sono state sostituite con LT, così da esplicitare i corsi di laurea triennale e migliorare la chiarezza dei dati. Successivamente, abbiamo proceduto con l'identificazione dei casi particolari utilizzando un controllo incrociato tra due colonne specifiche. Questo processo ha permesso di individuare e gestire in modo univoco i corsi di laurea che presentano requisiti particolari, come illustrato nella Figura 1. Per implementare questa logica, abbiamo adattato il processo di modifica dei dati al

¹<https://pandas.pydata.org/>

²<https://openpyxl.readthedocs.io/en/stable/>

corso specifico da verificare. Ad esempio, per la Laurea Triennale in Servizio Sociale, abbiamo sostituito il codice del tipo di corso precedentemente identificato come LT con LTSS.

| Corsi di studio | Docenza di riferim. (minimo) | Professori a tempo indetermin. (minimo) | Ricercatori | Docenti in convenz. art. 6, c. 11, L. 240/2010, oppure docenti art. 1, c. 12, L. 230/2005, oppure docenti a contr. art. 23, L. 240/2010 (massimo) | | Docenti di università straniere per CdS interateneo (art. 6, c. 11; art. 23, c. 3, L. 240/2010) |
|--|------------------------------|---|-------------|---|----------------------------|---|
| | | | | Tot. | di cui docenti a contratto | |
| LT | 9 | 5 | 4 | 3 | 2 | 4 |
| LM | 6 | 4 | 2 | 2 | 1 | 3 |
| LMCU 5 anni | 15 | 8 | 7 | 5 | 3 | 7 |
| LMCU 6 anni | 18 | 10 | 8 | 6 | 4 | 9 |
| LT Servizio Sociale LT Scienze Motorie | 5 | 3 | 2 | 2 | 1 | 2 |
| LT Prof. sanitarie LT a orient. profess. LM Servizio Sociale LM Scienze Motorie | 4 | 2 | 2 | 1 | 1 | 2 |
| LM Infermieristica | 3 | 1 | 2 | 1 | 1 | 1 |

Figure 1: Casi particolari

2.2 Analisi post incontro con l'ufficio

(MM: fare passaggio) Venerdì 6 dicembre 2024 abbiamo svolto un incontro con l'ufficio incaricato della gestione dei dati accademici, durante il quale sono stati affrontati alcuni punti critici emersi nella fase di analisi preliminare. Questo confronto ci ha consentito di ottenere chiarimenti fondamentali e di affinare ulteriormente il processo di preprocessing dei dati sopra descritto.

Un argomento importante di cui abbiamo discusso, riguarda la corretta gestione di situazioni in cui la numerosità delle immatricolazioni ai corsi di laurea risultano superiori alla numerosità massima teorica.

Un aspetto cruciale emerso riguarda la definizione delle preferenze relative ai docenti incaricati del ruolo di garante. È stata delineata una chiara categorizzazione ed preferenza per la valutazione, includendo in ordine: i docenti con copertura al 100% per un singolo corso, docenti a contratto con copertura esclusivamente al 100% per un singolo corso, e docenti a contratto con copertura al 100% per due corsi distinti.

Relativamente alla scelta dei garanti è stata esposta una preferenza secondaria, che implica la copertura del ruolo di garante per un corso da parte del presidente di quel corso di laurea.

Questi chiarimenti e criteri definiti durante l'incontro sono stati integrati nel processo di preprocessing, consentendo di riorganizzare i dati in modo coerente con le preferenze e i vincoli discussi. Sono stati implementati controlli aggiuntivi per garantire la consistenza dei dati elaborati, migliorando così l'efficacia del modello ASP nella fase di assegnazione.

3 ASP

In questa sezione approfondiamo l'organizzazione del progetto, evidenziando come sia stato strutturato per garantire chiarezza e modularità. Per facilitare l'ordine visivo e semplificare eventuali ispezioni manuali, il progetto è stato suddiviso in più file sorgenti, ciascuno dedicato a un aspetto specifico del dataset e delle regole.

Abbiamo separato i fatti e le regole in due file principali: uno dedicato ai docenti (Sezione 3.1) e un altro focalizzato sulle coperture (Sezione 3.2). Per maggiore praticità, è stato creato un file specifico per i docenti a contratto (Sezione 3.3) e un altro per i limiti ministeriali (Sezione 3.4).

Il file principale, il main, integra tutte le regole relative ai garanti, inclusi i vincoli e le priorità definite per l'ottimizzazione del sistema (Sezione 3.5).

È importante sottolineare che tutti questi file vengono generati dinamicamente dal programma di preprocessing scritto in Python. Questa fase, gestita dal file main del preprocessing, automatizza la creazione delle regole e dei fatti ASP in base ai dati forniti, garantendo così una configurazione personalizzata e facilmente aggiornabile.

3.1 Docenti

```

1 def saluta(nome):
2     print(f"Ciao, {nome}!")
3
4 saluta("Mondo")

```

Listing 1: Esempio di codice Python

3.2 Coperture

3.3 Docenti a contratto

(MM: spiegare qui che li usiamo come jolly)

3.4 Ministeriale

3.5 Vincoli

3.6 Gestione delle priorità

3.7 Ottimizzazioni

4 Esempio giocattolo

Per validare il modello sviluppato e testare l'efficacia dell'approccio proposto, abbiamo costruito un esempio giocattolo sfruttando il corso di laurea triennale di informatica ed il corso di laurea magistrale in Scienze Informatiche.

Il dataset generato ci ha accompagnato nel processo di testing del codice ASP proposto; permettendoci di verificare se fosse in grado di gestire correttamente la generazione dei termini, i vincoli e le preferenze specificate.

I risultati ottenuti sono stati utilizzati per ottimizzare ulteriormente il codice, permettendoci di identificare eventuali aree di miglioramento relative alla gestione dei vincoli e all'efficienza della generazione dei termini.

(MM: spiegare i comandi utilizzati per generare il dataset (il comando python))

(MM: specificare il comando utilizzato per lanciare l'analisi con clingo)

(MM: stampare l'output andando a commentare il risultato)

5 Valutazione sperimentale

5.1 Dataset piccolo (dipartimento SMFI)

5.2 Dataset grande (tutti i dipartimenti)

6 Conclusioni

Acknowledgments

Questo progetto è stato realizzato nel corso di Programmazione Dichiarativa (a.a. 2024-25), presso l'Università degli Studi di Parma.

References

Lifschitz, Vladimir (2002). “Answer set programming and plan generation”, *Artificial Intelligence*, Vol. 138 No. 1. Knowledge Representation and Logic Programming, pp. 39–54. ISSN: 0004-3702. DOI: [https://doi.org/10.1016/S0004-3702\(02\)00186-8](https://doi.org/10.1016/S0004-3702(02)00186-8). **available at:** <https://www.sciencedirect.com/science/article/pii/S0004370202001868>.