

Relazione IALab - Answer Set Programming

Roger Ferrod, Pio Raffaele Fina, Lorenzo Tabasso

Dipartimento di Informatica, Università degli Studi di Torino
`roger.ferrod@edu.unito.it`,
`pio.fina@edu.unito.it`,
`lorenzo.tabasso@edu.unito.it`

1 Introduzione

Il problema del *timetabling* consiste nell'assegnare un determinato numero di corsi ad un numero finito di slot temporali e aule. Il problema è soggetto ad *integrity constraints*, da soddisfare tassativamente, e vincoli *soft*, i quali hanno associata una funzione di costo da minimizzare [2]. Nelle specifiche del progetto sono presentati alcuni vincoli "auspicabili" da soddisfare ove possibile e nel maggior numero possibile; abbiamo interpretato tali vincoli come condizioni opzionali piuttosto che *soft constraints*. Il *timetabling* è un particolare caso di problema di *scheduling* appartenente alla classe di complessità NP-complete [3]. Il linguaggio di modellazione ricopre un ruolo fondamentale nella codifica del problema, in quanto viene richiesto un certo grado di espressività al fine di codificare i molteplici vincoli imposti dalla specifica [2]. Gli approcci per la soluzione di questo problema possono essere vagamente classificati in metodi basati su *ricerca operativa*, metodi basati su *metaeuristiche* e *multi-agents cooperative search* [1].

In questo progetto utilizziamo un approccio basato sul paradigma dell'*Answer Set Programming*. Lo *scheduling* può essere risolto in ASP fornendo una codifica del problema attraverso predicati, funzioni, fatti e vincoli in stile dichiarativo, successivamente il solver enumererà uno o più *answer-set* che rappresenteranno le soluzioni ammissibili eliminando gli *answer-set* che non soddisfano i vincoli.

2 Specifica

Per semplicità e completezza si riportano di seguito i vincoli descritti nella specifica del problema.

Le lezioni del master si svolgono il venerdì (8 ore) e il sabato (4 o 5 ore) nell'unica aula assegnata al Master, per 24 settimane. Inoltre, sono previste due settimane full-time, la 7a e la 16a, con lezioni dal lunedì al sabato (8 ore al giorno da lunedì a venerdì, 4 o 5 ore al sabato (**H0**)). Il calendario dovrà tener conto dei seguenti vincoli, da considerarsi partizionati in rigidi (da soddisfare tassativamente) e auspicabili (da soddisfare ove possibile, e nel maggior numero possibile). Gli insegnamenti da collocare sono accompagnati dal docente titolare e dal numero di ore da svolgere.

I seguenti vincoli sono da considerarsi rigidi:

- H1** Lo stesso docente non può svolgere più di 4 ore di lezione in un giorno.
- H2** A ciascun insegnamento vengono assegnate minimo 2 e massimo 4 ore nello stesso giorno.
- H3** il primo giorno di lezione prevede che, nelle prime due ore, vi sia la presentazione del master.
- H4** Il calendario deve prevedere almeno 2 blocchi liberi di 2 ore ciascuno per eventuali recuperi di lezioni annullate o rinviate.
- H5** l'insegnamento "Project Management" deve concludersi non oltre la prima settimana full-time
- H6** La prima lezione dell'insegnamento "Accessibilità e usabilità nella progettazione multimediale" deve essere collocata prima che siano terminate le lezioni dell'insegnamento "Linguaggi di markup".
- H7** Le lezioni dei vari insegnamenti devono rispettare le seguenti propedeuticità, in particolare la prima lezione dell'insegnamento della colonna di destra deve essere successiva all'ultima. Le varie propedeuticità sono riportate nella Tabella 1.

I seguenti vincoli sono da considerarsi auspicabili:

- A1** La distanza tra la prima e l'ultima lezione di ciascun insegnamento non deve superare le 6 settimane.
- A2** La prima lezione degli insegnamenti "Crossmedia: articolazione delle scritture multimediali" e "Introduzione al social media management" devono essere collocate nella seconda settimana full-time nello stesso giorno.
- A3** le lezioni dei vari insegnamenti devono rispettare le seguenti propedeuticità, in particolare la prima lezione dell'insegnamento della colonna di destra deve essere successiva alle prime 4 ore di lezione del corrispondente insegnamento della colonna di sinistra. Le propedeuticità sono riportate in Tabella 2.
- A4** la distanza fra l'ultima lezione di "Progettazione e sviluppo di applicazioni web su dispositivi mobile I" e la prima di "Progettazione e sviluppo di applicazioni web su dispositivi mobile II" non deve superare le due settimane.

3 Rappresentazione e Codifica del Problema

In questa sezione analizziamo i principali aspetti della codifica del problema, sia da un punto di vista qualitativo che quantitativo. Il repository è così strutturato:

- **facts.cl**: file contenente predicati che esprimono conoscenza di tipo fattuale (numero di settimane, materie del master, professori, propedeuticità, ecc.).
- **generation.cl** e **generation2.cl**: due versioni alternative per la generazione dei predicati **calendar**.
- **master.cl**: file principale dove sono codificati i vincoli e i predicati ausiliari.
- **toy.cl**: istanza ridotta del problema principale, utilizzata durante la prima fase di testing dei vincoli.

Insegnamento precedente	Insegnamento successivo
Fondamenti di ICT e Paradigmi di Programmazione	Ambienti di sviluppo e linguaggi client-side per il web
Ambienti di sviluppo e linguaggi client-side per il web	Progettazione e sviluppo di applicazioni web su dispositivi mobile I
Progettazione e sviluppo di applicazioni web su dispositivi mobile I	Progettazione e sviluppo di applicazioni web su dispositivi mobile II
Linguaggi di markup	Ambienti di sviluppo e linguaggi client-side per il web
Project Management	Marketing digitale
Marketing digitale	Tecniche e strumenti di Marketing digitale
Project Management	Strumenti e metodi di interazione nei Social media
Project Management	Progettazione grafica e design di interfacce
Acquisizione ed elaborazione di immagini statiche grafica	Elementi di fotografia digitale
Elementi di fotografia digitale	Acquisizione ed elaborazione di sequenze di immagini digitali
Acquisizione ed elaborazione di immagini statiche	Grafica 3D

Tabella 1. Tabella delle Propedeuticità obbligatorie.

Lo sviluppo del codice è stato guidato dai principi di *generalizzazione* ed *efficienza*. Dove possibile si è cercato di rendere il più generale possibile la codifica, in modo da agevolare eventuali estensioni future (aggiunta di corsi, professori, propedeuticità, ecc...). Spesso tali scelte hanno condizionato l'efficienza, soprattutto in fase di *grounding*.

3.1 Rappresentazione del tempo

Come già discusso nell'introduzione, il problema può essere naturalmente formulato in termini di *scheduling*. La dimensione temporale, qui intesa come **sequenza di eventi**, ricopre un ruolo fondamentale nella rappresentazione e codifica del problema. A livello concettuale, la componente temporale viene rappresentata da un modello gerarchico a tre livelli: settimane, giorni e ore, come illustrato in Figura 1.

Sebbene possa sembrare naturale modellare il vincolo H2 con blocchi dalla durata di due ore, si è deciso di avere una granularità più fine nel tempo, utilizzando slot di una singola ora. Tale soluzione si dimostra particolarmente adatta

Insegnamento precedente	Insegnamento successivo
Fondamenti di ICT e Paradigmi di Programmazione	Progettazione di basi di dati
Tecniche e strumenti di Marketing digitale	Introduzione al social media management
Comunicazione pubblicitaria e comunicazione pubblica	La gestione delle risorse umane
Tecnologie server-side per il web	Progettazione e sviluppo di applicazioni web sub dispositivi mobile

Tabella 2. Tabella delle Propedeuticità auspicabili.

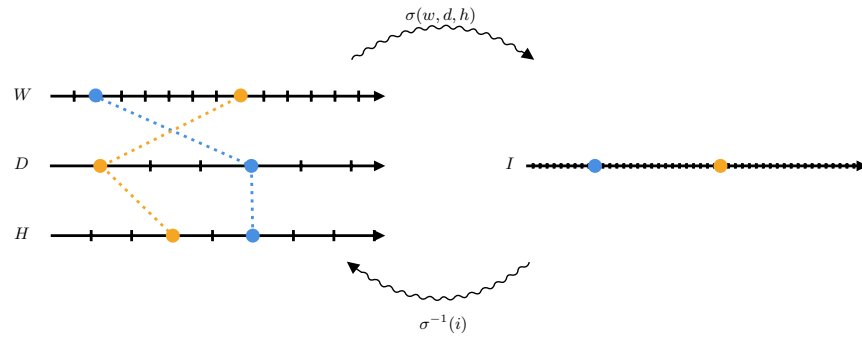


Figura 1. Modello concettuale della componente temporale.

ad implementare il vincolo H0, che prevede un numero variabile di ore al sabato. Considerando blocchi atomici da 2 ore, non è infatti possibile rappresentare i sabati con un numero dispari di ore; inoltre non è possibile escludere a priori la quinta ora del sabato, in quanto non si raggiungerebbe il monte ore finale.

Si è deciso di codificare un determinato istante di tempo con un numero naturale, seguendo i principi della *Numerazione di Gödel* e *Unix Time*. Più formalmente, lo schema di codifica si basa sulla seguente equazione:

$$\sigma(n_1, \dots, n_k) = \sum_{i=1}^k n_i \cdot 10^i \quad \forall n, i, k \in \mathbb{N} \quad (1)$$

L'input della funzione è una tupla (w, d, h) dove w, d, h rappresentano rispettivamente settimana, giorno e ora. L'equazione (1) si riduce dunque a:

$$\sigma(w, d, h) = w \cdot 10^2 + d \cdot 10 + h. \quad (2)$$

Il principale vantaggio di uno schema di codifica basato su (1) risiede nell'aver definito implicitamente una **relazione d'ordine totale** tra i timeslots, fondamentale per esprimere i vincoli temporali come H4, H5 e H7.

Note 1. L'applicazione $\sigma : W \times D \times H \rightarrow I$ è biettiva, è possibile dunque definire l'applicazione inversa $\sigma^{-1} : I \rightarrow W \times D \times H$. Tale proprietà permette facilmente di recuperare il numero di settimana w dal valore i :

$$\sigma_w^{-1}(i) = \left\lfloor \frac{i}{100} \right\rfloor$$

Tale proprietà è utilizzata, ad esempio, nella codifica del vincolo A4.

3.2 Predicato Calendar

Il predicato principale utilizzato nella codifica del problema è `calendar(W,D,H,I,S,P)`. Tale predicato rappresenta il singolo timeslot, associato alla materia **S** ed insegnata dal professore **P**, in programma nella settimana **W**, il giorno **D** nell'ora **H**. Il quarto argomento (**I**) implementa l'equazione 2 e rappresenta un identificatore univoco per il timeslot. La codifica basata sul predicato 6-ario `calendar` permette di consolidare l'informazione in un unico predicato, consentendo una lettura più immediata e un debugging più agevole.

L'esecuzione del programma segue quindi un approccio *generate-and-test*: nella prima fase si genera lo spazio di ricerca, successivamente, utilizzando i vincoli, si eliminano gli stati (in questo caso gli *answer-set*) che rappresentano soluzioni non ammissibili.

Versione 1 La prima codifica sviluppata (file `generation.cl`) prevedeva la generazione dei termini `calendar` utilizzando unicamente una *cardinality constraint* nella testa della regola (*head aggregates*) (Code 1).

```
0 {calendar(W, D, H, W*100+D*10+H, S, P)} 1 :- week(W), day(D),
   ↪ hour(H), subject(S,P,_).
```

Code 1: Cardinality constraint `calendar`.

L'utilizzo della *cardinality constraint* con upper bound 1 permette di generare **al più** un termine `calendar` per ogni combinazione di W , D e H , mentre il lower bound 0 fornisce la possibilità di non generare alcuno slot. Con la regola in Code 1 viene generato l'intero spazio di ricerca corrispondente al prodotto cartesiano dei termini grounded `week`, `day`, `hour`, `subject`. Nell'istanza del problema con 24 settimane e 28 materie (24 reali + 2 fittizie: *"Introduzione al master"* e *"Recupero"*), lo spazio degli stati conta 32.256 ($26 \cdot 6 \cdot 8 \cdot 28$) termini `calendar` grounded. In questo caso dunque, la generazione dello spazio di ricerca avviene in modo **non informato**, poiché non viene presa in considerazione la specifica del problema. Ad esempio, si considerino le seguenti istanze di `calendar`:

```
calendar(1, 1, 3, 113, ..., ...)
calendar(5, 6, 8, 568, ..., ...)
```

Queste non potranno mai far parte di una soluzione candidata: la settimana 1 non è full week e di conseguenza non è possibile avere lezione il lunedì (giorno 1) come indicato nel primo esempio; inoltre il sabato (giorno 6) prevede al più 5 ore di lezione, dunque il valore 8 (secondo esempio) non è accettabile.

Per questi motivi l'utilizzo della cardinality constraint deve essere accompagnata da una serie di vincoli che permettono l'eliminazione degli Answer Set incompatibili con il dato dominio.

Versione 2 Nella seconda versione (file `generation2.cl`), la generazione di `calendar` è stata suddivisa in 3 fasi:

- Generazione dei calendar dal lunedì al giovedì per le settimane full week


```
0 {calendar(W, D, H, W*100+D*10+H, S, P)} 1 :-
  → week(W), isFullWeek(W), in_week_hours(D, H), D <
  → 5, subject(S, P, _).
```
- Generazione dei calendar solo per il venerdì


```
0 {calendar(W, D, H, W*100+D*10+H, S, P)} 1 :-
  → week(W), in_week_hours(D, H), D = 5, subject(S,
  → P, _).
```
- Generazione dei calendar solo per il sabato


```
0 {calendar(W, D, H, W*100+D*10+H, S, P)} 1 :-
  → week(W), saturday_hours(D,H), subject(S, P, _).
```

Asserendo i fatti:

```
in_week_hours(1..5,1..8).
saturday_hours(6,1..5).
```

Generando *ex ante* sottospazi dell'intero spazio di ricerca, viene ridotta la complessità (temporale, ma soprattutto spaziale), ottimizzando la fase di *grounding*.

Sebbene la prima versione sia notevolmente più chiara e immediata da comprendere, la seconda permette di ridurre, di circa un terzo, il numero di termini `calendar` generati, passando da 32.256 a 10.528. Abbiamo dunque optato per l'utilizzo della seconda soluzione in quanto estremamente più efficiente.

3.3 Predicato Period

La codifica di alcuni vincoli (H6, A1 e A4) necessitano della conoscenza della prima e ultima lezione di una o più `subject`. Tale informazione è direttamente reperibile attraverso l'uso degli aggregati `#min` e `#max`:

```
#min{I: calendar(_, _, _, I, S, _)}, subject(S, _, _).
#max{I: calendar(_, _, _, I, S, _)}, subject(S, _, _).
```

Code 2: Aggregati per la prima e ultima lezione del corso `S`.

Durante la fase di *grounding* queste due operazioni, e più in generale l'uso degli aggregati, hanno un grande impatto in termini di complessità computazionale (sia spaziale che temporale). Al fine di rendere più efficiente il calcolo, è stato definito un unico predicato `period` (Code 3), utile a modellare più vincoli:

```
period(F, L, S) :- F = #min{I: calendar(_, _, _, I, S, _)},
                  L = #max{I: calendar(_, _, _, I, S, _)},
                  subject(S, _, _).
```

Code 3: Predicato `period`.

In questo modo si attua una strategia "computing once" asserendo, per ogni corso S , il primo (F) e l'ultimo (L) timeslot di lezione.

Note 2. Gli aggregati `#min` e `#max`, se applicati a insiemi vuoti, restituiscono rispettivamente le costanti `#sup` ed `#inf`. Le operazioni aritmetiche tra valori numerici e `#sup` o `#inf` non sono definite e costituiscono un errore. Nel nostro caso specifico, per via della cardinalità imposta all'aggregato `calendar` (Code 1), è possibile generare insiemi vuoti, di conseguenza le operazioni di `#min` e `#max` degenerano in `#sup` ed `#inf`. Per ovviare a tale problema è necessario escludere, con un vincolo, questi casi (`#min != #sup` e `#max != #inf`). Un esempio di tale accorgimento è presente nella codifica del predicato ausiliare `week_length`:

```
week_length(L/100-F/100, S) :- period(F, L, S), subject(S,
    ↪   _ , _), L!=#inf, F!=#sup.
```

3.4 Vincoli Rigidi

Vincoli H3 e H4 La codifica dei due vincoli è molto simile, pertanto la trattazione sarà comune. La specifica prevede che alcuni timeslots vengano riservati per la presentazione del master H3 e per dei blocchi di recupero H4. La modellazione dei due vincoli avviene in maniera implicita grazie al meccanismo di generazione dei timeslots, è dunque sufficiente aggiungere due corsi "fittizi" in `facts.cl`.

```
subject("Introduzione al Master", "", 2).
subject("Recupero", "", 4).
```

La presenza del vincolo

```
:- X != #count{I : calendar(_, _, _, I, S, _)}, subject(S, _,
    ↪   X).
```

garantisce che ogni predicato `subject(S,_,X)`, con X ore di insegnamento, abbia esattamente X timeslots nel calendario. In questo modo è possibile, ad esempio, definire le ore di "Introduzione al Master":

```
calendar(1, 5, 1, 151, "Introduzione al Master", "").
calendar(1, 5, 2, 152, "Introduzione al Master", "").
```

prestando però attenzione alla peculiarità di queste lezioni (poste nelle prime 2 ore della prima settimana). Va dunque eliminata la possibilità alle restanti lezioni di occupare gli slot con indice $I=151$ e $I=152$, questo viene garantito dalla presenza del seguente vincolo:

```
:- calendar(_, _, _, I1, "Introduzione al Master", _),
   calendar(_, _, _, I2, S, _), subject(S, _, _),
   I1 > I2,
   S != "Introduzione al Master".
```

Vincolo H7 Il vincolo di propedeuticità richiede che sia rispettata la Tabella 1, in particolare ogni elemento della prima colonna deve precedere il corrispondente corso della seconda colonna. Grazie alla rappresentazione di `propaedeutic(S1, S2)` è stato possibile generalizzare ed esprimere il vincolo nel seguente modo:

```
:- calendar(_, _, _, I1, S1, _), calendar(_, _, _, I2, S2, _),
   ↪ propaedeutic(S1, S2), I1 > I2.
```

Ossia, ogni slot associato al corso `S1` deve precedere ogni slot di `S2`.

3.5 Vincoli Auspicabili

Vincolo A3 Il terzo vincolo auspicabile rappresenta una variante del vincolo H7, in particolare si aggiunge l'individuazione della prima e quarta ora di ogni corso. Per adempiere a tale compito abbiamo utilizzato diversi predicati ausiliari, tra cui:

1. `gt(X, I, S)`: individua tutti gli slot (`X`) del corso `S` successivi ad uno slot dato (`I`)
2. `fourth_hour(X, S)`: calcola la quarta ora (`X`) del corso `S`
3. `first_right(X, S)`: calcola la prima ora (`X`) del corso `S`, restringendo però l'unificazione di `S` alla sola colonna di destra (Tabella 2)
4. `first_left(X, S)`: calcola la prima ora (`X`) del corso `S` unificando con la colonna di sinistra
5. `all_greater_than_1(X, S)`: individua tutti gli slot (`X`) del corso `S` successivi alla prima lezione

Siccome la linea temporale è discreta, è stato possibile applicare iterativamente il predicato `gt` per ordinare le istanze e ricavare le prime N ore di ogni corso. Ad esempio, la seconda ora di un corso è individuata dal minimo slot calcolato sull'insieme degli slot successivi alla prima ora, pertanto è possibile codificare il predicato `second_hour` nel seguente modo:

```
second_hour(X, S) :- X = #min{Y : all_greater_than_1(Y, S)},
   ↪ propaedeuticSoft(S, _).
```


In modo analogo sono stati definiti gli altri predicati ausiliari, arrivando all'implementazione del vincolo come segue:

```
:- fourth_hour(X, S1), first_right(Y, S2),
   ↪ propaedeuticSoft(S1, S2), Y < X.
```

L'utilizzo di due predicati diversi per il calcolo della prima ora (`first_left` e `first_right`) è dettato da esigenze di ottimizzazione delle performace: è infatti inutile calcolare la quarta ora dei corsi presenti nella colonna di destra, mentre è necessario calcolare la prima ora per entrambe le colonne. Di conseguenza i predicati `second_hour`, `third_hour` e `fourth_hour` unificheranno solamente con i corsi della prima colonna.

4 Esperimenti

Gli esperimenti che mostriamo in questa sezione sono stati eseguiti con la versione *clingo 5.4.0* sulle seguenti macchine:

M1 Intel Core i7-4278U (2 core con 2 thread), 16 GB RAM.
M2 Intel Core i7-7700K (4 core con 2 thread), 16 GB RAM.
M3 Intel Core i5-6200U (2 core con 2 thread), 8 GB RAM.

È possibile eseguire il calcolo in modalità *multithreading* specificando il flag `-t`, in questo modo il *solving* verrà processato parallelamente su più core; tuttavia, aumentando il numero di thread, aumenta anche il consumo di memoria centrale. Il *multithreading* non ha impatto sulla fase di *grounding* in quanto non è paralelizzabile.

Tutti gli esperimenti sono stati eseguiti con 3 thread, i risultati sono mostrati in Tabella 3.

	Tempo	RAM
M1	2 min 11 s	4 GB
M2	0 min 48 s	4 GB
M3	1 min 54 s	4 GB

Tabella 3. Statistiche di esecuzione esperimenti.

Nessuna delle tre macchine a nostra disposizione è riuscita a completare il calcolo con la variante `generation.cl`, poiché la quantità di memoria richiesta dalla fase di *grounding* eccedeva il limite del sistema in tutti i casi. È stato dunque necessario ricorrere ad un server remoto¹ per misurare l'effettivo consumo di memoria che ammonta a 34 GB, in fase di *grounding*, e 56 GB durante

¹ Intel Xeon Skylake (16 core con 1 thread), 64 GB RAM

il *solving* (utilizzando 7 thread). In questo caso il tempo di calcolo complessivo risulta essere di 1 ora e 10 minuti. Questo particolare esperimento sottolinea l'esigenza e l'utilità delle ottimizzazioni implementate in `generation2.cl`.

5 Risultati

A scopo di migliore leggibilità e interpretazione del risultato, è stato sviluppato uno script Python, il quale prende in input un file di testo contenente l'output di Clingo, e genera in output un file di Excel rappresentante il calendario finale del master.

Nella tabella sono state evidenziate alcune celle per verificare più facilmente il soddisfacimento dei vincoli. In particolare: il colore rosso è associato al vincolo **H4**, mentre i colori verde e giallo al vincolo **H7**.

week	day	ora 1	ora 2	ora 3	ora 4	ora 5	ora 6	ora 7	ora 8
1	ven	Introduzione al Master:	Introduzione al Master:	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato	Project Management: Muzzetto	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato	Project Management: Muzzetto	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato
	sab	Recupero:	Project Management: Muzzetto	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato	I vincoli giuridici del progetto: diritto dei media: Travostino	Project Management: Muzzetto			
2	ven	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato	I vincoli giuridici del progetto: diritto dei media: Travostino	Project Management: Muzzetto	Project Management: Muzzetto	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Project Management: Muzzetto	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi
	sab	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato			
3	ven	Accessibilità e usabilità nella progettazione multimediale: Gena	Acquisizione ed elaborazione del suono: Valle	Acquisizione ed elaborazione del suono: Valle	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Accessibilità e usabilità nella progettazione multimediale: Gena	Accessibilità e usabilità nella progettazione multimediale: Gena	Progettazione di basi di dati: Mazzei	Progettazione di basi di dati: Mazzei
	sab	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Progettazione di basi di dati: Mazzei	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi			
4	ven	Progettazione di basi di dati: Mazzei	Risorse digitali per il progetto: collaborazione e documentazione: Boniolo	Recupero:		Accessibilità e usabilità nella progettazione multimediale: Gena	Accessibilità e usabilità nella progettazione multimediale: Gena	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Progettazione di basi di dati: Mazzei
	sab	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	Progettazione di basi di dati: Mazzei	Accessibilità e usabilità nella progettazione multimediale: Gena		Fondamenti di ICT e Paradigmi di Programmazione: Pozzato			
5	ven	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato	Acquisizione ed elaborazione del suono: Valle	Acquisizione ed elaborazione del suono: Valle	Progettazione di basi di dati: Mazzei	Progettazione di basi di dati: Mazzei	Risorse digitali per il progetto: collaborazione e documentazione: Boniolo	Progettazione di basi di dati: Mazzei	Progettazione di basi di dati: Mazzei
	sab	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato		La gestione delle risorse umane: Lombardo	Project Management: Muzzetto				
6	ven	Acquisizione ed elaborazione del suono: Valle	Acquisizione ed elaborazione del suono: Valle	I vincoli giuridici del progetto: diritto dei media: Travostino	Risorse digitali per il progetto: collaborazione e documentazione: Boniolo	Progettazione di basi di dati: Mazzei	Acquisizione ed elaborazione del suono: Valle	Progettazione di basi di dati: Mazzei	Progettazione di basi di dati: Mazzei
	sab	Progettazione di basi di dati: Mazzei	Project Management: Muzzetto	Project Management: Muzzetto	Project Management: Muzzetto	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato			
7	lun	Risorse digitali per il progetto: collaborazione e documentazione: Boniolo	Risorse digitali per il progetto: collaborazione e documentazione: Boniolo	Project Management: Muzzetto	Risorse digitali per il progetto: collaborazione e documentazione: Boniolo	Project Management: Muzzetto			I vincoli giuridici del progetto: diritto dei media: Travostino
	mar	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato	Acquisizione ed elaborazione del suono: Valle	Recupero:		La gestione delle risorse umane: Lombardo	I vincoli giuridici del progetto: diritto dei media: Travostino	La gestione delle risorse umane: Lombardo	I vincoli giuridici del progetto: diritto dei media: Travostino
	mer	La gestione delle risorse umane: Lombardo	I vincoli giuridici del progetto: diritto dei media: Travostino	Acquisizione ed elaborazione del suono: Valle	Fondamenti di ICT e Paradigmi di Programmazione: Pozzato	Comunicazione pubblicitaria e comunicazione pubblica: Gabardi	La gestione delle risorse umane: Lombardo	La gestione delle risorse umane: Lombardo	I vincoli giuridici del progetto: diritto dei media: Travostino
	gio	Recupero:	I vincoli giuridici del progetto: diritto dei media: Travostino	La gestione delle risorse umane: Lombardo	Acquisizione ed elaborazione del suono: Valle	I vincoli giuridici del progetto: diritto dei media: Travostino	La gestione delle risorse umane: Lombardo	Risorse digitali per il progetto: collaborazione e documentazione: Boniolo	Risorse digitali per il progetto: collaborazione e documentazione: Boniolo

	ven	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Risorse digitali per il progetto: collaborazione e documentazione: Boniolo	Progettazione di basi di dati: Mazzei	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Progettazione di basi di dati: Mazzei	Progettazione di basi di dati: Mazzei
	sab	Progettazione di basi di dati: Mazzei	Project Management: Muzzetto		La gestione delle risorse umane: Lombardo	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio			
8	ven	Marketing digitale: Muzzetto	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Marketing digitale: Muzzetto	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Accessibilità e usabilità nella progettazione multimediale: Gena	Marketing digitale: Muzzetto	Progettazione di basi di dati: Mazzei	Accessibilità e usabilità nella progettazione multimediale: Gena
	sab	Strumenti e metodi di interazione nei Social media: Giordani	Marketing digitale: Muzzetto	Strumenti e metodi di interazione nei Social media: Giordani	Accessibilità e usabilità nella progettazione multimediale: Gena	Strumenti e metodi di interazione nei Social media: Giordani			
9	ven	Accessibilità e usabilità nella progettazione multimediale: Gena	Accessibilità e usabilità nella progettazione multimediale: Gena	Accessibilità e usabilità nella progettazione multimediale: Gena	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Accessibilità e usabilità nella progettazione multimediale: Gena	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Progettazione di basi di dati: Mazzei	Strumenti e metodi di interazione nei Social media: Giordani
	sab	Strumenti e metodi di interazione nei Social media: Giordani	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Strumenti e metodi di interazione nei Social media: Giordani	Accessibilità e usabilità nella progettazione multimediale: Gena	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta			
10	ven	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	La gestione delle risorse umane: Lombardo	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Risorse digitali per il progetto: collaborazione e documentazione: Boniolo	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Marketing digitale: Muzzetto	Marketing digitale: Muzzetto	Marketing digitale: Muzzetto
	sab	Strumenti e metodi di interazione nei Social media: Giordani	Strumenti e metodi di interazione nei Social media: Giordani	Strumenti e metodi di interazione nei Social media: Giordani	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Strumenti e metodi di interazione nei Social media: Giordani			
11	ven	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Marketing digitale: Muzzetto	La gestione della qualità: Tomatis	Marketing digitale: Muzzetto	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Marketing digitale: Muzzetto
	sab	Linguaggi di markup: Gena	Strumenti e metodi di interazione nei Social media: Giordani		Linguaggi di markup: Gena	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta			
12	ven	Linguaggi di markup: Gena	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Linguaggi di markup: Gena	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Ambienti di sviluppo e linguaggi client-side per il web: Micalizio	Linguaggi di markup: Gena	Linguaggi di markup: Gena
	sab	Strumenti e metodi di interazione nei Social media: Giordani	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Strumenti e metodi di interazione nei Social media: Giordani			
13	ven	La gestione della qualità: Tomatis	Linguaggi di markup: Gena	La gestione della qualità: Tomatis	Linguaggi di markup: Gena	La gestione della qualità: Tomatis	La gestione della qualità: Tomatis	Linguaggi di markup: Gena	Linguaggi di markup: Gena
	sab	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Linguaggi di markup: Gena			
14	ven	La gestione della qualità: Tomatis	La gestione della qualità: Tomatis	La gestione della qualità: Tomatis	Linguaggi di markup: Gena	La gestione della qualità: Tomatis	Linguaggi di markup: Gena	Linguaggi di markup: Gena	Linguaggi di markup: Gena
	sab	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Linguaggi di markup: Gena	Strumenti e metodi di interazione nei Social media: Giordani	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta	Acquisizione ed elaborazione di immagini statiche-grafica: Zanchetta			

15	ven		Linguaggi di markup: Gena	Tecniche e strumenti di Marketing digitale: Zanchetta	Linguaggi di markup: Gena	Linguaggi di markup: Gena	Grafica 3D: Gribaudo	Linguaggi di markup: Gena	Grafica 3D: Gribaudo
	sab	Tecniche e strumenti di Marketing digitale: Zanchetta	Tecniche e strumenti di Marketing digitale: Zanchetta	Tecniche e strumenti di Marketing digitale: Zanchetta	Progettazione grafica e design di interfacce: Terranova	Progettazione grafica e design di interfacce: Terranova			
16	lun	Crossmedia: articolazione delle scritture multimediali: Taddeo	Introduzione al social media management: Suppini	Elementi di fotografia digitale: Vargiu	Crossmedia: articolazione delle scritture multimediali: Taddeo	Crossmedia: articolazione delle scritture multimediali: Taddeo	Crossmedia: articolazione delle scritture multimediali: Taddeo	Elementi di fotografia digitale: Vargiu	Grafica 3D: Gribaudo
	mar	Crossmedia: articolazione delle scritture multimediali: Taddeo	Elementi di fotografia digitale: Vargiu	Introduzione al social media management: Suppini	Introduzione al social media management: Suppini	Introduzione al social media management: Suppini	Crossmedia: articolazione delle scritture multimediali: Taddeo	Crossmedia: articolazione delle scritture multimediali: Taddeo	Crossmedia: articolazione delle scritture multimediali: Taddeo
	mer	Crossmedia: articolazione delle scritture multimediali: Taddeo	Introduzione al social media management: Suppini	Crossmedia: articolazione delle scritture multimediali: Taddeo	Crossmedia: articolazione delle scritture multimediali: Taddeo	Introduzione al social media management: Suppini	Crossmedia: articolazione delle scritture multimediali: Taddeo	Introduzione al social media management: Suppini	Progettazione grafica e design di interfacce: Terranova
	gio	Crossmedia: articolazione delle scritture multimediali: Taddeo	Crossmedia: articolazione delle scritture multimediali: Taddeo	Introduzione al social media management: Suppini	Crossmedia: articolazione delle scritture multimediali: Taddeo	Progettazione grafica e design di interfacce: Terranova	Crossmedia: articolazione delle scritture multimediali: Taddeo	Introduzione al social media management: Suppini	Introduzione al social media management: Suppini
	ven	Crossmedia: articolazione delle scritture multimediali: Taddeo	Introduzione al social media management: Suppini	Introduzione al social media management: Suppini	Crossmedia: articolazione delle scritture multimediali: Taddeo	Crossmedia: articolazione delle scritture multimediali: Taddeo	La gestione della qualità: Tomatis	Crossmedia: articolazione delle scritture multimediali: Taddeo	Grafica 3D: Gribaudo
	sab	Introduzione al social media management: Suppini		Introduzione al social media management: Suppini	Elementi di fotografia digitale: Vargiu	Grafica 3D: Gribaudo			
17	ven	Progettazione grafica e design di interfacce: Terranova	Progettazione grafica e design di interfacce: Terranova	Semiologia e multimedialità: Santangelo	Progettazione grafica e design di interfacce: Terranova	Elementi di fotografia digitale: Vargiu	Progettazione grafica e design di interfacce: Terranova	Grafica 3D: Gribaudo	Elementi di fotografia digitale: Vargiu
	sab	Grafica 3D: Gribaudo	Grafica 3D: Gribaudo	Grafica 3D: Gribaudo	Elementi di fotografia digitale: Vargiu	Grafica 3D: Gribaudo			
18	ven	Tecnologie server-side per il web: Damiano	Tecnologie server-side per il web: Damiano	Elementi di fotografia digitale: Vargiu	Progettazione grafica e design di interfacce: Terranova	Progettazione grafica e design di interfacce: Terranova	Tecnologie server-side per il web: Damiano	Tecnologie server-side per il web: Damiano	Progettazione e sviluppo di applicazioni web su dispositivi mobile I: Pozzato
	sab	Grafica 3D: Gribaudo	Grafica 3D: Gribaudo	Tecniche e strumenti di Marketing digitale: Zanchetta	Progettazione e sviluppo di applicazioni web su dispositivi mobile I: Pozzato	Tecnologie server-side per il web: Damiano			
19	ven	Progettazione e sviluppo di applicazioni web su dispositivi mobile I: Pozzato	Progettazione e sviluppo di applicazioni web su dispositivi mobile I: Pozzato	Progettazione e sviluppo di applicazioni web su dispositivi mobile I: Pozzato		Progettazione e sviluppo di applicazioni web su dispositivi mobile I: Pozzato	Grafica 3D: Gribaudo	Grafica 3D: Gribaudo	Grafica 3D: Gribaudo
	sab	Tecniche e strumenti di Marketing digitale: Zanchetta	Elementi di fotografia digitale: Vargiu	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Grafica 3D: Gribaudo			
20	ven	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Tecniche e strumenti di Marketing digitale: Zanchetta	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Progettazione e sviluppo di applicazioni web su dispositivi mobile I: Pozzato	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Progettazione e sviluppo di applicazioni web su dispositivi mobile I: Pozzato	Grafica 3D: Gribaudo	Grafica 3D: Gribaudo

	sab	Grafica 3D: Gribaudo	Tecnologie server-side per il web: Damiano	Semiologia e multimedialità: Santangelo	Semiologia e multimedialità: Santangelo	Progettazione e sviluppo di applicazioni web su dispositivi mobile I: Pozzato			
21	ven	Progettazione e sviluppo di applicazioni web su dispositivi mobile I: Pozzato	Tecniche e strumenti di Marketing digitale: Zanchetta	Tecnologie server-side per il web: Damiano	Tecniche e strumenti di Marketing digitale: Zanchetta	Semiologia e multimedialità: Santangelo	Semiologia e multimedialità: Santangelo	Semiologia e multimedialità: Santangelo	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli
	sab	Progettazione e sviluppo di applicazioni web su dispositivi mobile II: Schifanella	Semiologia e multimedialità: Santangelo	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Tecniche e strumenti di Marketing digitale: Zanchetta	Semiologia e multimedialità: Santangelo			
22	ven	Tecnologie server-side per il web: Damiano	Progettazione e sviluppo di applicazioni web su dispositivi mobile II: Schifanella	Tecnologie server-side per il web: Damiano	Tecnologie server-side per il web: Damiano	Progettazione e sviluppo di applicazioni web su dispositivi mobile II: Schifanella	Progettazione e sviluppo di applicazioni web su dispositivi mobile II: Schifanella		Progettazione e sviluppo di applicazioni web su dispositivi mobile II: Schifanella
	sab	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Tecnologie server-side per il web: Damiano	Tecnologie server-side per il web: Damiano	Tecnologie server-side per il web: Damiano	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli			
23	ven	Tecnologie server-side per il web: Damiano	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Tecnologie server-side per il web: Damiano	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Semiologia e multimedialità: Santangelo	Tecnologie server-side per il web: Damiano	Semiologia e multimedialità: Santangelo	Progettazione e sviluppo di applicazioni web su dispositivi mobile II: Schifanella
	sab	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Progettazione e sviluppo di applicazioni web su dispositivi mobile II: Schifanella			
24	ven	Tecnologie server-side per il web: Damiano	Progettazione e sviluppo di applicazioni web su dispositivi mobile II: Schifanella	Tecnologie server-side per il web: Damiano	Progettazione e sviluppo di applicazioni web su dispositivi mobile II: Schifanella	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Tecnologie server-side per il web: Damiano	Progettazione e sviluppo di applicazioni web su dispositivi mobile II: Schifanella	
	sab	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Tecnologie server-side per il web: Damiano	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli	Acquisizione ed elaborazione di sequenze di immagini digitali: Ghidelli			

Riferimenti bibliografici

- [1] Hamed Babaei, Jaber Karimpour e Amin Hadidi. «A survey of approaches for university course timetabling problem». In: *Computers & Industrial Engineering* 86 (2015). Applications of Computational Intelligence and Fuzzy Logic to Manufacturing and Service Systems, pp. 43–59. ISSN: 0360-8352. DOI: <https://doi.org/10.1016/j.cie.2014.11.010>. URL: <http://www.sciencedirect.com/science/article/pii/S0360835214003714>.
- [2] MUTSUNORI BANBARA et al. «Answer set programming as a modeling language for course timetabling». In: *Theory and Practice of Logic Programming* 13 (lug. 2013). DOI: 10.1017/S1471068413000495.
- [3] J.D. Ullman. «NP-complete scheduling problems». In: *Journal of Computer and System Sciences* 10.3 (1975), pp. 384–393. ISSN: 0022-0000. DOI: [https://doi.org/10.1016/S0022-0000\(75\)80008-0](https://doi.org/10.1016/S0022-0000(75)80008-0). URL: <http://www.sciencedirect.com/science/article/pii/S0022000075800080>.