

# Intelligenza Artificiale e Laboratorio

Anno Accademico 2019/20

Giuseppe Mazzone	Mat. 938528
Sante Altamura	Mat. 941238

# Programmazione logica: Prolog e ASP

## Implementazione di strategie di ricerca con Prolog

L'approccio seguito per la realizzazione del progetto ha permesso la creazione di algoritmi per implementare strategie di ricerca non informate e basate su euristiche, applicandole al problema del labirinto. Si è inoltre scelto di limitare l'output degli algoritmi solamente al primo risultato, dato che è garantito essere ottimo. A livello implementativo tale politica è stata realizzata mediante l'eliminazione dei punti di scelta del backtracking considerati superflui, attraverso l'uso del *cut*.

Per testare le strategie di ricerca è possibile effettuare la consult del loader [`'loader.pl'`] ed eseguire uno dei seguenti comandi:

- `playid` per l'Iterative Deepening;
- `playida` per l'Iterative Deepening A\*;
- `playa` per l'A\*;

## Strategie non informate

### Iterative Deepening

L'implementazione dell'algoritmo risulta semplice e naturale grazie al meccanismo di backtracking offerto da Prolog. Alla base vi è una ricerca a profondità limitata standard che si basa su un limite fornito come variabile dinamica, inizialmente inizializzato a 1:

```
:-dynamic(limite/1).  
limite(1).
```

Inizialmente l'interprete prolog prende in considerazione la prima clausola a cui si dà il parametro di output "Soluzione" poi considera e cerca di dimostrare il predicato `ricercaPath`. Dopo c'è un cut in modo tale da rendere le scelte definitive, cioè, se è possibile una soluzione per `ricercaPath` allora ignora tutte le altre possibili scelte in cui aumenta di uno il limite

```
iterativeDeepening(Soluzione):-  
    ricercaPath(Soluzione),  
    !,  
    retract(limite(_)),  
    assert(limite(1)).
```

Nel passo base, effettua una ricerca in profondità con limite quello fornito dinamicamente sopra riportato:

```
ricercaPath(Soluzione):-  
    limite(N),  
    Limite is N,  
    write(Limite),nl,  
    iniziale(S),  
    ricerca_depth_limitata(S,[],Limite,Soluzione).
```

Se la `ricercaPath` non ricade nel passo base, cioè l'interprete prolog non è riuscito a trovare uno stato finale in base al limite corrente viene eseguita la seconda scelta della `ricercaPath` che permette di aumentare il limite di 1 e di controllare che non si sia giunti alla massima profondità:

```
ricercaPath(Soluzione):-  
    limite(N),  
    N1 is N+1,  
    massimaProfondita(D),  
    N1 =< D,  
    retract(limite(_)),  
    assert(limite(N1)),  
    ricercaPath(Soluzione).
```

Esplorare uno stato attraverso significa generare una azione valida per esso, applicarla, controllare di non aver scoperto una configurazione già precedentemente analizzata ed, in tal caso, ripetere il procedimento dall'inizio. Tale algoritmo è denominato `ricerca_depth_limitata` e viene invocato con il livello di taglio che,

durante la discesa in profondità nell'albero, non deve superare.

La sequenza di azioni che conducono dallo stato iniziale a quello finale viene elaborata mediante una strategia backward: una volta raggiunto un obiettivo la lista soluzione viene costruita percorrendo a ritroso il ramo che ha portato al goal,aggiungendo in testa alla lista tutte le azioni che vengono trovate durante il cammino.

## Strategie informate

### Euristiche implementate

#### Algoritmo IDA\*

Questo algoritmo permette di evolvere la ricerca di Iterative Deepening andando ad utilizzare una euristica per stimare il numero della soglia ad ogni passo di ricerca. Tale strategia utilizza una particolare struttura **nodo** per associare ad un determinato stato del problema la stima euristica **f(n)**:

`ida_nodo(Stato, StimaEuristica)`

dove:

- **Stato**: è uno stato proprio del dominio del problema dal quale vogliamo stimare la distanza;
- **StimaEuristica**: è la distanza stimata da Stato allo stato finale;

Tale struttura è fornita in maniera dinamica ed è definita dal seguente predicato extra-logico:

`:-dynamic(ida_nodo/2).`

Al primo passo di ricerca la soglia è data dalla stima euristica calcolata tra il nodo iniziale e il nodo finale, è importante notare la presenza del CUT che in questo caso ci permette di eliminare tutti gli altri percorsi dal ramo di risoluzione ottenendo quindi solo una soluzione:

```
idaStar(Sol):-  
    iniziale(S),  
    heuristic(S, _, EuristicaIniziale),  
    ida(S, Sol, [S], 0, EuristicaIniziale),  
    !.
```

Ad ogni iterazione viene effettuata la ricerca in profondità secondo la soglia determinata dall'euristica. Successivamente al passo iniziale, la stima euristica corrisponderà al minimo:

$$f(S) = g(S) + h(S)$$

per tutti i nodi che superano la soglia del passo precedente. Se la ricerca in profondità fallisce allora vengono presi in considerazione i predicati successivi, altrimenti, viene correttamente istanziato il parametro di output. Nel caso vengano considerati i predicati successivi, il funzionamento per il ricalcolo della soglia è il seguente: vengono prese le  $F(S)$  per tutti gli stati attraverso il predicato `findall` e tra queste vengono considerate solo quelle che superano la soglia precedente (attraverso la `exclude`) ed inserite in `OverEuristicaLista`; ordinate le euristiche dalla più piccola alla più grande, viene presa la soglia che corrisponde alla minima euristica trovata e si chiama ricorsivamente `ida` sulla nuova soglia.

```
ida(S, Sol, Visitati, CostoPercorsoS, Euristica):-
    ida_search(S, Sol, Visitati, CostoPercorsoS, Euristica);
    findall(FS, ida_nodo(_, FS), EuristicaLista),
    exclude(>=(Euristica), EuristicaLista, OverEuristicaLista),
    sort(OverEuristicaLista, ListaOrdinata),
    nth0(0, ListaOrdinata, NuovaEuristica),
    retractall(ida_nodo(_, _)),
    ida(S, Sol, Visitati, 0, NuovaEuristica).
```

Clausola che rappresenta il caso base della ricerca in profondità, se lo stato è finale.

```
ida_search(S, [], _, _, _):-
    finale(S).
```

Se lo stato non è finale, avviene una ricerca in profondità, con limite dettato dall'euristica. Viene scelta un'azione applicabile nello stato  $S$ , si trova uno stato nuovo che verifichi il predicato `trasforma` si verifica che non appartenga agli stati visitati e si calcola il costo per passare al nuovo stato; per questo tipo di esperimento il costo è sempre unitario. Il costo del percorso per il nuovo stato `CostoPercorsoNuovaS` è dato da:

$$CostoPercorsoS + Costo$$

mentre il costo del percorso per il nuovo stato `FNuovoS` è dato da:

$$CostoPercorsoNuovaS + CostoHeuristicNuovaS$$

viene asserito un predicato che mi permette di tracciare  $SNuovo$  con la sua  $F(SNuovo)$ ; se la profondità del nuovo stato è minore dell'euristica limite si chiama ricorsivamente `ida_search` sul nuovo stato.

```

ida_search(S, [Az|AltreAzioni], Visitati, CostoPercorsoS, Euristica):-
    applicabile(Az, S),
    trasforma(Az, S, NuovoS),
    \+member(NuovoS, Visitati),
    costo(S, NuovoS, Costo),
    CostoPercorsoNuovaS is CostoPercorsoS + Costo,
    heuristic(NuovoS, _, CostoHeuristicNuovaS),
    FNuovoS is CostoPercorsoNuovaS + CostoHeuristicNuovaS,
    assert(ida_nodo(NuovoS, FNuovoS)), FNuovoS <= Euristica,
    ida_search(NuovoS, AltreAzioni, [NuovoS|Visitati],
    CostoPercorsoNuovaS, Euristica).

```

### Algoritmo A\*

Essendo questo algoritmo basato sulla ricerca in ampiezza e non essendo tale ricerca naturale per prolog la sua implementazione risulta più complessa, infatti si rende necessaria la creazione di una struttura dati ad hoc simile a quella precedente per associare ad un determinato stato del problema tutte le informazioni ad esso correlate. Si definisce così un nodo:

```
nodo(S, AzioniPerS, CostoAttuale, StimaEuristica)
```

dove:

- **S**: è uno stato proprio del dominio del problema;
- **AzioniPerS**: è la lista delle azioni che hanno portato a **Stato** dallo stato iniziale.
- **CostoAttuale**: è la distanza di **Stato** dallo stato iniziale;
- **StimaEuristica**: è la distanza stimata da **Stato** allo stato finale;

La clausola seguente permette di inizializzare il problema indicando lo stato iniziale e calcolando l'euristica iniziale a partire da tale stato S. Subito dopo viene invocato `astarRicerca` sul nodo di partenza, trovata una soluzione essa sarà l'unica ad essere prodotta grazie al CUT:

```

astar(Soluzione):-
    iniziale(S),
    heuristic(S, _, HeuristicS),
    astarRicerca([nodo(S, [], 0, HeuristicS)], [], Soluzione),
    !.

```

Nel caso base di `astarRicerca` se ci si trova nello stato finale, le azioni da restituire come parametro di output sono proprio le azioni per raggiungere `S`

```
astarRicerca([nodo(S,AzioniPerS,_,_)|_],_,AzioniPerS):-
    finale(S),!.
```

Mentre il caso induttivo permette effettivamente di eseguire la ricerca in ampiezza, che a differenza di quella standard, ordina la lista dei nuovi stati possibili secondo le  $F(S)$  di tali nodi privilegiando quelle con stima minore:

```
astarRicerca([nodo(S,AzioniPerS,CostoAttuale,
HeuristicAttuale)|CodaStati],
Visitati,Soluzione):-
    findall(Az,applicabile(Az,S),ListaAzioniApplicabili),
    generaStatiFigli(nodo(S,AzioniPerS,CostoAttuale,HeuristicAttuale),
[S|Visitati],ListaAzioniApplicabili,StatiFigli),
    append(CodaStati,StatiFigli,NuovaCoda),
    pedsort(a_star_comparator,NuovaCoda,CodaOrdinata),
    astarRicerca(CodaOrdinata,[S|Visitati],Soluzione).
```

Nel caso base la lista azioni applicabili è vuota allora, banalmente, la lista di stati figli è vuota.

```
generaStatiFigli(_,_,[],[]).
```

Nel caso induttivo, ad ogni chiamata viene posizionato in testa alla lista dei figli il nodo rappresentato dal costo del suo percorso e dal valore dell'euristica.

```
generaStatiFigli(nodo(S,AzioniPerS,CostoAttuale,HeuristicAttuale)
,Visitati,[Az|AltreAzioni],[nodo(SNuovo,[Az|AzioniPerS],
NuovoCostoPerS,NuovaEuristicaPerS)|AltriFigli]):-
    trasforma(Az,S,SNuovo), \+member(SNuovo,Visitati),!,
    costo(S,SNuovo,Costo), NuovoCostoPerS is CostoAttuale + Costo,
    heuristic(SNuovo,_,NuovaEuristicaPerS),
    generaStatiFigli(nodo(S,AzioniPerS,CostoAttuale
,HeuristicAttuale),Visitati,AltreAzioni,AltriFigli).
```

Fa riferimento al caso in cui `SNuovo` è già stato visitato ma è necessario proseguire la ricerca.

```
generaStatiFigli(Nodo,Visitati,[_|AltreAzioni],AltriFigli):-
    generaStatiFigli(Nodo,Visitati,AltreAzioni,AltriFigli).
```

## Mappe Utilizzate

- Per la creazione delle mappe è stato utilizzato un generatore di labirinti implementato in Java.
- Per la creazione delle mappe in prolog, viene generato dallo stesso tool un file testuale con le posizioni occupate.
- Le posizioni iniziali **I** e finali **F** sono state scelte a posteriori dai progettisti.

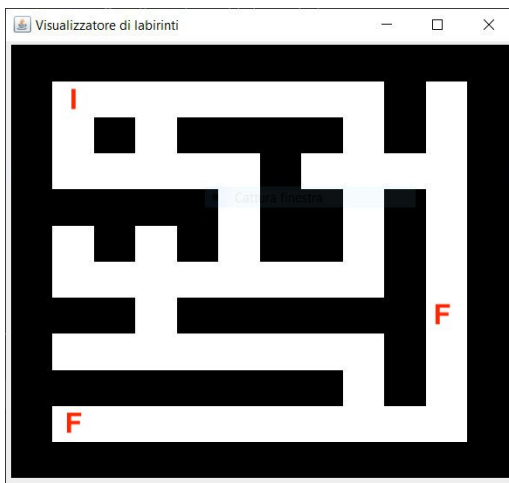


Figura 1: 10x10 A

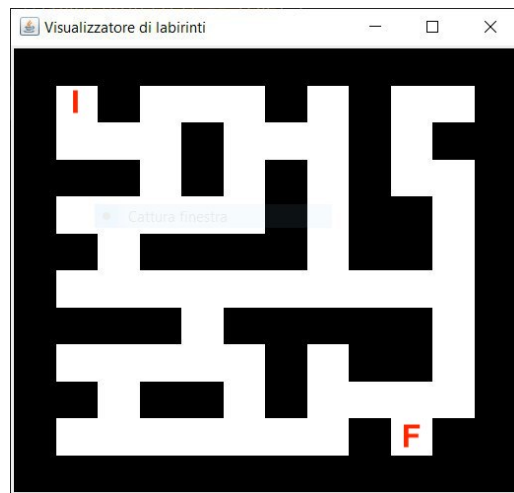


Figura 2: 10x10 B

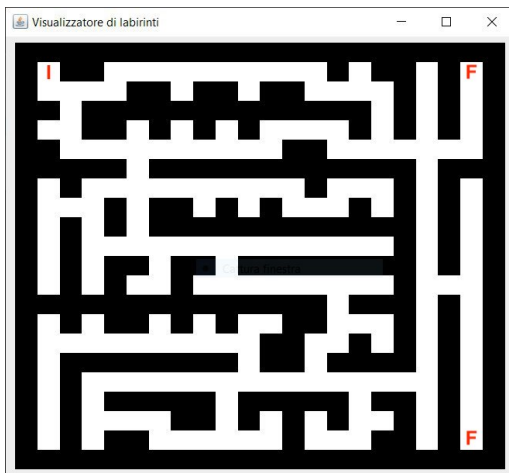


Figura 3: 20x20 A

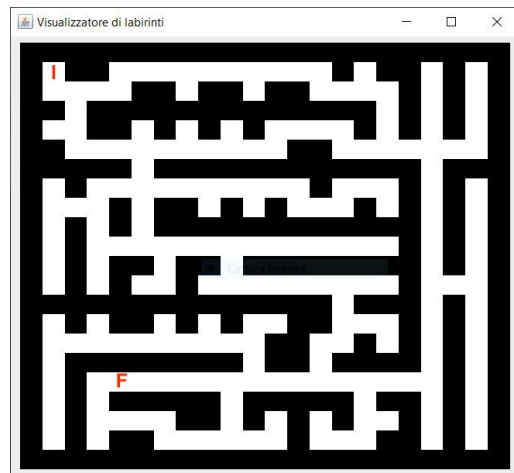


Figura 4: 20x20 B



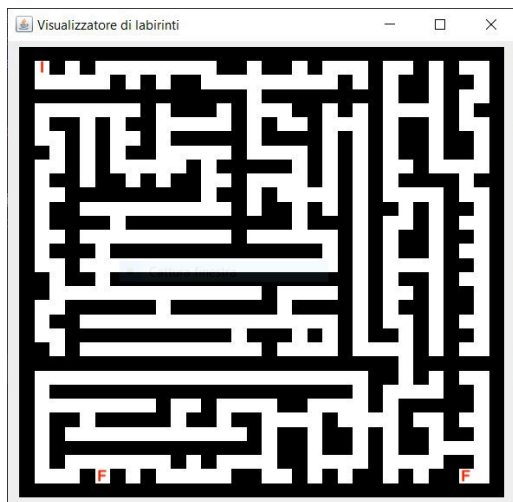


Figura 5: 30x30 A

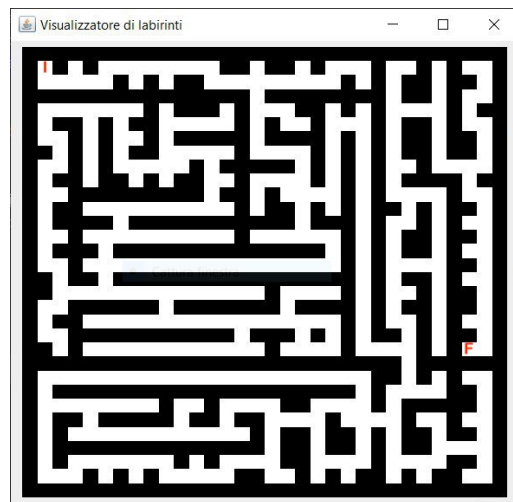


Figura 6: 30x30 B

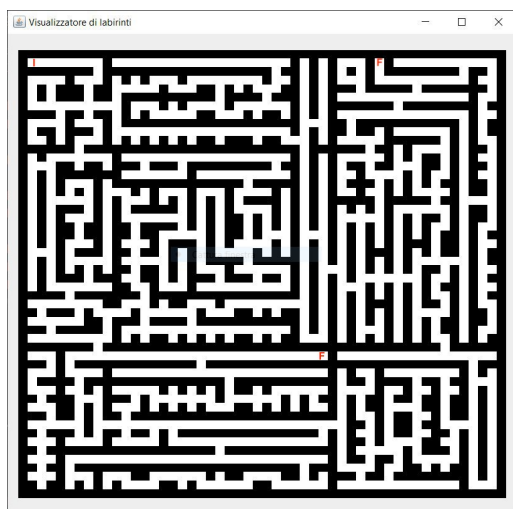


Figura 7: 50x50 A

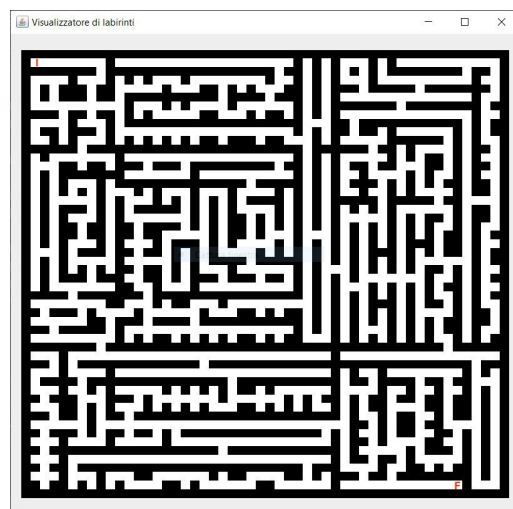


Figura 8: 50x50 B

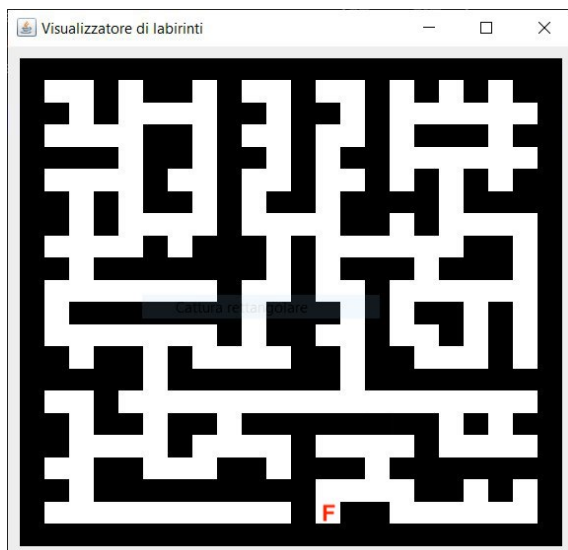
## Risultati Ottenuti

- In tutti i test effettuati, la strategia di ricerca informata  $A^*$  si è dimostrata superiore a tutte le altre: è stata la strategia più efficiente in termini di tempo. Questo vale sia per l'euristica basata sulla distanza di Manatthan sia per quella basata sulla distanza Euclidea, tenendo conto che sono due euristiche ammissibili per  $A^*$ , in quanto la rendono ammissibile. Intuitivamente, si può anche capire il perchè in questo caso  $A^*$  è computazionalmente ottimo: ha una **stima ottimistica** del costo del percorso attraverso ogni nodo considerato ed inoltre, trattandosi di una ricerca in ampiezza, non espande i nodi già espansi.
- L'Iterative Deepening  $A^*$  risulta avere tempi di computazione buoni se la ricerca avviene su mappe di piccole-medie dimensioni. Uno svantaggio rispetto ad  $A^*$  si può notare in relazione all'euristica utilizzata: la distanza euclidea, fa aumentare i tempi di esecuzione di IDA\* per tutte le mappe, risultando anche più oneroso dell'Iterative Deepening. Invece, utilizzando la distanza di Manatthan come euristica, il tempo di computazione risulta essere notevolmente ridotto.

Tutte le strategie implementate, rispettano la:

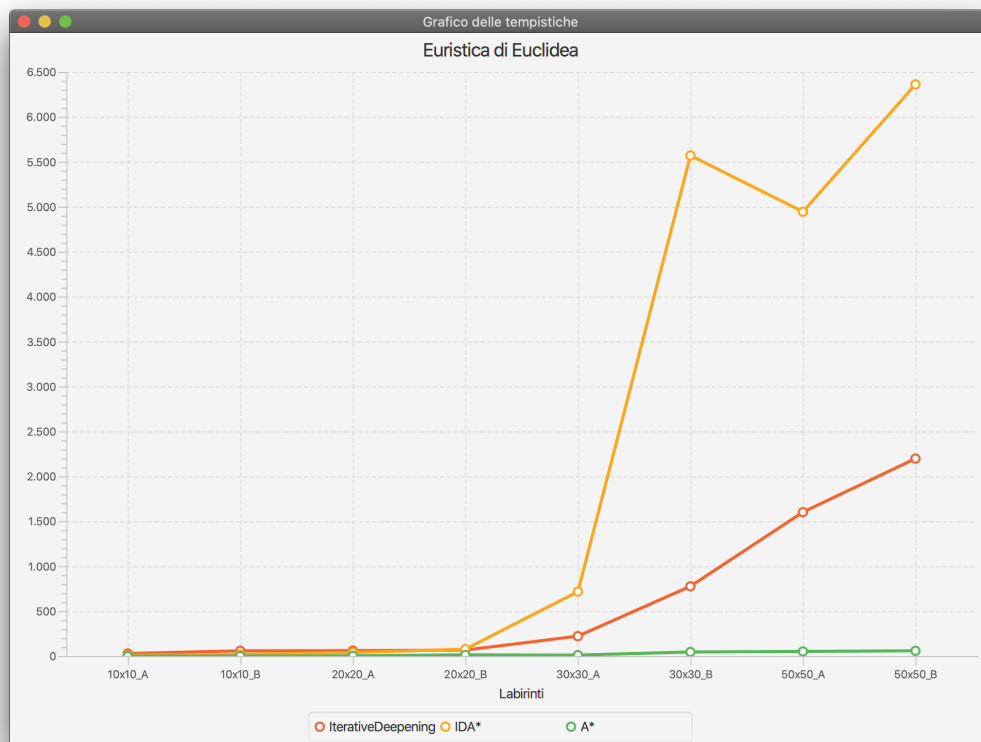
- **Correttezza;**
- **Completezza;**
- **Condizione di terminazione;**

Le strategie di ricerca sono state testate anche su mappe per le quali non è possibile raggiungere lo stato finale, come ad esempio:



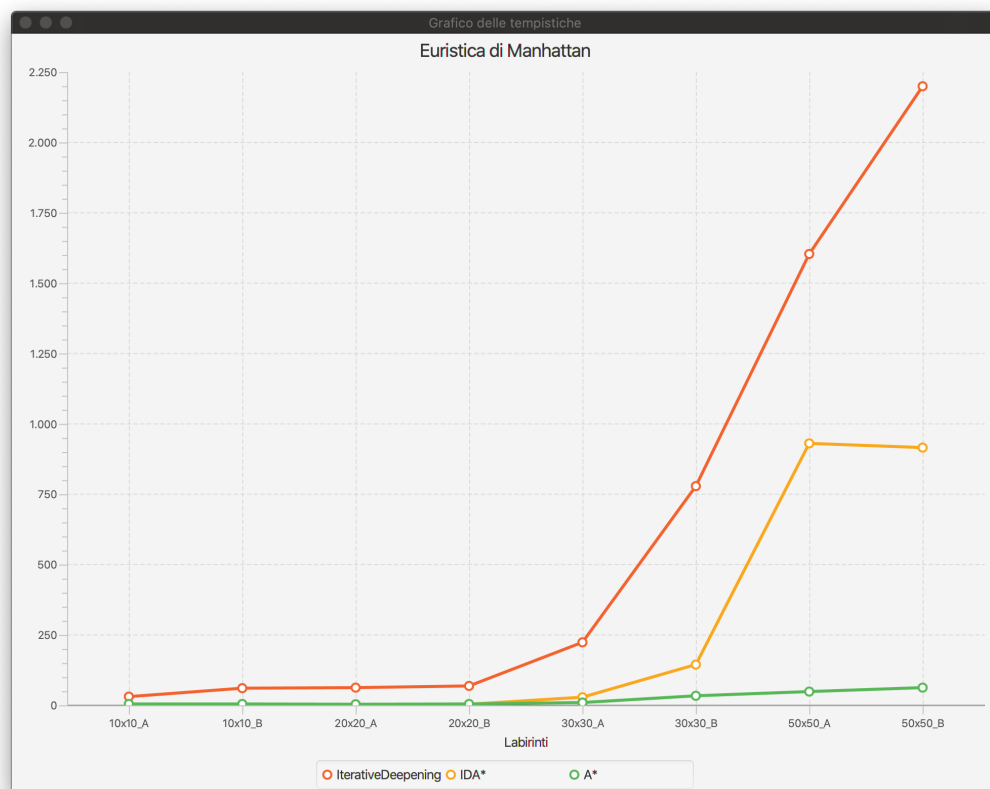
Strategia	Euclidea							
	10x10A	10x10B	20x20A	20x20B	30x30A	30x30B	50x50A	50x50B
Iterative Deepening	29	59	61	67	222	777	1602	2198
IDA*	9	19	40	77	717	5570	4945	6361
A*	0	1	3	14	11	46	52	59

Tabella 1: Inferenze eseguite



Strategia	Manhattan							
	10x10A	10x10B	20x20A	20x20B	30x30A	30x30B	50x50A	50x50B
Iterative Deepening	29	59	61	67	222	777	1602	2198
IDA*	0	1	1	2	27	143	929	914
A*	3	3	2	3	8	32	47	61

Tabella 2: Inferenze eseguite



# Constraint Satisfaction Problem con ASP (Answer Set Programming)

## Calendario delle lezioni

### Consegna

Formulare e risolvere il seguente problema di soddisfacimento di vincoli: generazione del **calendario del Master in Progettazione e Management del Multimedia per la Comunicazione** dell'Università di Torino che, con la sua storia iniziata nel 1996, è il più longevo d'Europa. Le lezioni del master si svolgono il venerdì (8 ore) e il sabato (4 o 5 ore) nell'unica aula assegnata al Master, per 23 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). Il calendario dovrà tener conto di alcuni vincoli, da considerarsi partizionati in rigidi (da soddisfare tassativamente) e auspicabili (da soddisfare ove possibile, e nel maggior numero possibile).

### Soluzione

Il file `Orario_Lezioni.cl` contiene la soluzione del problema di soddisfazione di vincoli scritto in clingo. Attraverso `CSPTtoEXCEL.py` scritto in Python è possibile eseguire il file `.cl` indicando il numero di soluzioni che si vogliono produrre; attraverso procedure di parsing viene restituito in output uno o più elaborati excel (in base al numero di soluzioni richieste) che contengono 24 fogli, ognuno contenente la programmazione delle lezioni per la specifica settimana in ordine di **Giorno** e **Ora** di **Inizio**.

### Scelte implementative

- **Predicati e Costanti**
  - `corso(Corso)`: `Corso` indica il nominativo univoco di ogni corso.
  - `nOreCorso(Num, Corso)`: `Num` indica il numero di ore previste per ogni `Corso`.
  - `docente(Docente)`: `Docente` è l'identificativo di un docente.
  - `insegna(Docente, Corso)`: `Docente` indica il nominativo del docente che insegna il corso `Corso`.
  - `settimana(Settimana)`: `Settimana` indica un numero identificativo per una specifica settimana, vengono quindi creati 24 predicati.

- `settimanaFull(Settimana)`: `Settimana` indica l'identificativo delle due settimane full previste nel calendario.
- `giorno(Giorno)`: `Giorno` definisce l'identificativo di ogni giorno della settimana, vengono quindi creati 6 predicati dal lunedì al sabato.
- `giornoFeriale(Giorno)`: `Giorno` definisce l'unico giorno in cui non è previsto l'orario full di 8 ore.
- `giornNum(Id,Giorno)`: `Id` definisce l'identificativo numerico per ogni giorno (utile per effettuare l'ordinamento).
- `orario(Inizio,Fine)`: `Inizio` contiene l'ora di inizio possibile mentre `Fine` contiene l'ora di fine, vengono creati 8 diversi predicati del tipo `orario(9,10)` `orario(10,11)` etc.

- **Modellazione del problema**

- Si è creato il predicato `ha_ore(Giorno,Ora)` che crea il prodotto cartesiano tra tutte le istanze di `Giorno` e di `Ora`.

```
8 {ha_ore(Giorno, Ora):orario(Ora,_)} 8:- giorno(Giorno),not
giornoFeriale(Giorno).
```

in questo caso vengono assegnate 8 ore per ogni `Giorno`, dove `Giorno` è verificato, per fallimento, non essere una occorrenza di `giornoFeriale(Giorno)`.

```
4 {ha_ore(Giorno,Ora):orario(Ora,_)} 5:- giornoFeriale(Giorno).
```

in questo caso vengono assegnate da 4 a 5 ore per ogni `Giorno`, dove `Giorno` è una occorrenza di `giornoFeriale(Giorno)` (banalmente solo per il sabato).

- Si è creato il predicato `settimane(Settimana,Giorno)` che crea l'insieme dei giorni possibili per ogni settimana per tutte le istanze di `Settimana` e di `Giorno`.

```
24 {settimane(S,venerdi):settimana(S)} 24.
```

in questo caso vengono assegnati 24 predicati diversi per ogni `Settimana` per il giorno specifico `venerdi`.

```
24 {settimane(S,sabato):settimana(S)} 24.
```

in questo caso vengono assegnati 24 predicati diversi per ogni `Settimana` per il giorno specifico `sabato`.

```
6 {settimane(Settimana,Giorno): giorno(Giorno)} 6:-
    settimanaFull(Settimana).
```

in questo caso vengono assegnati 6 predicati diversi per ogni settimana (dove *Settimana* indica l'occorrenza in *settimanaFull(Settimana)* ) per il giorno specifico *venerdi*.

```
4 {ha_ore(Giorno,Ora):orario(Ora,_)} 5:- giornoFeriale(Giorno).
```

in questo caso vengono assegnate da 4 a 5 ore per ogni *Giorno*, dove *Giorno* è una occorrenza di *giornoFeriale(Giorno)* (banalmente solo per il sabato).

- **Vincoli**

- Un corso deve avere al massimo 4 ore in un giorno:

```
0 {slotLezione(Doc, Corso, Ora, Giorno, Settimana):
    orario(Ora,_)} 4:- insegna(Doc, Corso),
    settimane(Settimana, Giorno), ha_ore(Giorno, Ora).
```

Tale regola permette anche di assegnare effettivamente di assegnare da 0 a 4 ore di *orario(Ora,\_)* per ogni *Corso*, ogni *Settimana*, ogni *Giorno*.

- Il sabato può avere solo 4 o 5 ore:

```
:- slotLezione(Doc, Corso, Ora, Giorno, Settimana),
    not ha_ore(Giorno, Ora).
```

permette di eliminare dal modello tutti quei *slotLezione(Doc, Corso, Ora, Giorno, Settimana)* che non rispettano il predicato *ha\_ore(Giorno,Ora)* precedentemente creato.

- Non ci possono essere più insegnamenti in una stessa ora di uno stesso giorno:

```
:- slotLezione(_, Ins1, Ora, Giorno, Settimana),
    slotLezione(_, Ins2, Ora, Giorno, Settimana), Ins1 != Ins2.
```

permette di eliminare dal modello tutte le ore assegnate a più di un insegnamento.

- Ogni giorno non feriale ha 8 ore di lezione. ogni giorno feriale ha non meno di 4 ore di lezione. Per risolvere tale vincolo si è creato un predicato che contiene per ogni giorno di ogni settimana il numero di ore di lezione:

```
oreGiorno(Giorno, Settimana, N):-
  N = #count{Ora:slotLezione(_, _, Ora, Giorno, Settimana)},
  settimane(Settimana, Giorno).
```

e subito dopo si sono eliminati tutti i modelli che hanno il conteggio di ore per giorni feriali inferiore a 4 e per i giorni non feriali inferiore ad 8.

```
:- oreGiorno(Giorno, Settimana, N), giornoFeriale(Giorno), N<4.
```

```
:- oreGiorno(Giorno, Settimana, N), not giornoFeriale(Giorno),
   giorno(Giorno), N<8
```

- Un corso può avere minimo 2 e massimo 4 ore in uno stesso giorno, per soddisfare questo vincolo prima di tutto si è calcolato il numero di ore di lezione per ogni Corso in ogni giorno, settimana:

```
conteggioOreGiornaliere(Conteggio, Corso, Giorno, Settimana) :-
  Conteggio = #count{ Ora : slotLezione(_, Corso, Ora, Giorno, Settimana)},
```

e subito dopo, con l'utilizzo di un integrity constraint, si sono eliminati dal modello tutti quei `slotLezione(Doc, Corso, Ora, Giorno, Settimana)` che hanno il conteggio uguale a 1 (NB: l'aggregato `ha_ore` iniziale ha già imposto il numero massimo di 4 ore in un giorno per ogni corso.

```
:- conteggioOreGiornaliere(Conteggio, Corso, Giorno, Settimana),
   Conteggio==1.
```

- Un docente può tenere al più 4 ore di lezione al giorno, per soddisfare questo vincolo prima di tutto si è calcolato il numero di ore di lezione per ogni Docente in ogni giorno, settimana:



```
conteggioOreProfessore(Conteggio, Docente, Giorno, Settimana) :-
Conteggio = #count{ Ora : slotLezione(Docente, _, Ora, Giorno, Settimana)},
slotLezione(Docente, _, _, Giorno, Settimana), docente(Docente).
```

grazie al seguente integrity constraint, si sono eliminati i modelli che prevedevano più di 4 ore in un giorno per un Docente.

```
:- conteggioOreProfessore(Conteggio, Docente, Giorno, Settimana),
Conteggio>4.
```

- Il primo giorno di lezione (il venerdì della settimana 1) prevede che, nelle prime due ore, vi sia la presentazione al master (NB: La presentazione è schedulata come fosse un corso, con 2 ore da assegnare):

```
:- slotLezione(Doc,presentazione, Ora, Giorno, Settimana), Ora > 10.
:- slotLezione(Doc,presentazione, Ora, Giorno, Settimana),
    Giorno != venerdì.
:- slotLezione(Doc,presentazione, Ora, Giorno, Settimana),
    Settimana != 1.
```

permettono di eliminare dal modello tutti gli slot in cui la presentazione non è assegnata alla settimana 1, al giorno venerdì e che abbia ora maggiore di 10 (prevedendo che le due ore vengono assegnate alle uniche due ore che sono praticamente 9 e 10).

- Sono previsti due blocchi da 2 ore consecutive ciascuno per l'eventuale recupero di lezioni:

```
:- slotLezione(Doc, blocco_libero, Ora1, Giorno, Settimana),
    slotLezione(Doc, blocco_libero, Ora2, Giorno, Settimana),
    Ora1 != Ora2, |Ora1 - Ora2| != 1.
```

tale vincolo permette di eliminare dal modello tutti quegli assegnamenti in cui i due slot non sono consecutivi (NB: La presentazione è schedulata come fosse un corso, con 4 ore da assegnare)

- Ogni corso ha un numero massimo di ore di insegnamento, definite da `nOreCorso(NOre, Corso)`, per soddisfare questo vincolo prima di tutto si è calcolato il numero di ore di lezione per ogni Corso in tutti gli slot assegnati:

```

conteggioOreTotali(Corso, Conteggio) :-
Conteggio = #count{ Settimana, Giorno, Ora :
slotLezione(_, Corso, Ora, Giorno, Settimana)}, corso(Corso).

```

attraverso il seguente integrity constraint, si sono eliminati tutti i modelli che non prevedevano lo stesso numero di ore.

```

:- corso(Corso), nOreCorso(TotOre, Corso),
conteggioOreTotali(Corso, N), N != TotOre.

```

- Sono richiesti diversi vincoli che richiedono l'utilizzo di primo giorno di lezione e ultimo giorno di lezione, per questo si sono creati dei predicati ad hoc che specificano rispettivamente il primo giorno e la prima ora, e l'ultimo giorno e l'ultima ora:

```

primaSettimana(Giorno,Inizio,MinSettimana,Corso):-
#min {Settimana:lezione(_, _, _, Corso, Settimana, _)} = MinSettimana,
lezione(GG, Inizio,_,Corso,MinSettimana,_),giorNum(Giorno,GG).

```

```

primoGiorno(MinGiorno,Inizio,Settimana,Corso):-
#min {Giorno:primaSettimana(Giorno,_, _, Corso)} = MinGiorno,
primaSettimana(MinGiorno,Inizio, Settimana, Corso).

```

```

primaLezione(Giorno,MinInizio,Settimana,Corso):-
#min {Inizio:primoGiorno(_,Inizio, _, Corso)} = MinInizio,
primoGiorno(Giorno,MinInizio, Settimana, Corso).

```

```

ultimaSettimana(Giorno,Inizio,MinSettimana,Corso):-
#max {Settimana:lezione(_, _, _, Corso, Settimana, _)} = MinSettimana,
lezione(GG, Inizio,_,Corso,MinSettimana,_),giorNum(Giorno,GG).

```

```

ultimoGiorno(MinGiorno,Inizio,Settimana,Corso):-
#max {Giorno:ultimaSettimana(Giorno,_, _, Corso)} = MinGiorno,
ultimaSettimana(MinGiorno,Inizio, Settimana, Corso).

```

```

ultimaLezione(Giorno,MinInizio,Settimana,Corso):-
#max {Inizio:ultimoGiorno(_,Inizio, _, Corso)} = MinInizio,
ultimoGiorno(Giorno,MinInizio, Settimana, Corso).

```

Grazie all'utilizzo di un predicato `propedeuticita(M1,M2)`, dove M1 indica il corso che deve terminare prima dell'inizio del corso M2, l'integrity constraint per i due insegnamenti è stato così implementato:

```
:- propedeuticita(M1,M2), ultimaLezione(_, _, Settimana1, M1),  
primaLezione(_, _, Settimana2, M2), Settimana2<Settimana1.  
:- propedeuticita(M1,M2), ultimaLezione(Giorno1, _, Settimana, M1),  
primaLezione(Giorno2, _, Settimana, M2), Giorno2<=Giorno1.
```

in questo modo la prima lezione di M2 inizierà solo dopo almeno un giorno dalla fine del corso M2.

- Il corso 'projectManagement' deve concludersi non oltre la prima settimana full-time (ovvero la sesta settimana):

```
:- ultimaLezione(_, _,Settimana, project_management),  
   Settimana > 7.
```

in questo modo si eliminano tutti i modelli che hanno come ultima lezione per 'projectManagement' una Settimana che va oltre la sesta.

- La prima lezione del corso 'Accessibilità e usabilità nella progettazione multimediale' deve essere collocata prima che siano terminate le lezioni del corso 'Linguaggi di markup'

```
:- ultimaLezione(_, _, Settimana, linguaggi_di_markup),  
primaLezione(_, _,Settimana2,  
accessibilita_e_usabilita_nella_progettazione_multimediale),  
Settimana2>Settimana.
```

- la distanza tra la prima e l'ultima lezione di ciascun Corso non deve superare le 6 settimane:

```
:- primaLezione(_, _, Settimana1, M),  
   ultimaLezione(_, _, Settimana2, M),  
   (Settimana2-Settimana1)>6.
```

- La prima lezione degli insegnamenti 'Crossmedia: articolazione delle scritture multimediali' e 'Introduzione al social media management' devono essere collocate nella seconda settimana full-time.

```
:- primaLezione(_, _,Settimana,
crossmedia_articolazione_delle_scritture_multimediali), Settimana!=16.
:- primaLezione(_, _,Settimana,
introduzione_al_social_media_management), Settimana!=16.
```

- 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.

```
:- primaLezione(_, _, Settimana2,
progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii),
ultimaLezione(_, _, Settimana1,
progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i),
(Settimana2-Settimana1)>2.
```

- La prima lezione dell'insegnamento M2 (colonna di destra della tabella) deve essere successiva alle prime 4 ore di lezione del corrispondente insegnamento M1 (colonna di sinistra). Per poter codificare tale vincolo si è prima di tutto creato un conteggio che permette di calcolare il numero di ore nella prima lezione del corso M1, per poi successivamente eliminare tutti i modelli con conteggio minore di 4. In questo modo, sappiamo che dopo il primo giorno di lezione dell'insegnamento M1, esso avrà già fatto le prime 4 ore. Successivamente, si è creato un predicato che individua il secondo giorno di lezione di ogni corso; grazie a quest'ultimo predicato si è imposto che la prima lezione del corso M2 deve iniziare dopo la prima lezione, e prima della seconda lezione del corso M1.

```
contOrePG(M1,Conteggio):-
Conteggio == #count{ Ora:primoGiorno(Giorno,Ora,_,M1)},
primaLezione(Giorno,_,_,M1), propedeuticit4(M1,_).
:- contOrePG(M1,Conteggio), Conteggio<4.
```

In questo modo eliminiamo tutti i modelli che prevedono per il corso M1 meno di 4 ore nella prima lezione.

```

lezTemp(Giorno, Inizio, Corso, Settimana):-
giorNum(Ng,Giorno),lezione(Giorno, Inizio, _, Corso, Settimana, _),
not primaLezione(Ng,_,Settimana,Corso).

settimanaSecondaLezione(Giorno,Inizio,MinSettimana,Corso):-
#min {Settimana:lezTemp(_, _, Corso, Settimana)} = MinSettimana,
lezTemp(GG, Inizio,Corso,MinSettimana),giorNum(Giorno,GG).

secondoGiornoLezione(MinGiorno,Inizio,Settimana,Corso):-
#min {Giorno:settimanaSecondaLezione(Giorno,_, _, Corso)} = MinGiorno,
settimanaSecondaLezione(MinGiorno,Inizio, Settimana, Corso).

```

Per creare quindi il predicato `giornoSecondaLezione` si è prima di tutto creato un predicato che ha eliminato dall'insieme delle lezioni il primo giorno di lezione (`lezTemp`), subito dopo si è effettuato il calcolo del secondo giorno analogamente a come si è ricavato il primo giorno di lezione, in quanto quest'ultimo non sarà più presente nell'insieme.

```

:- propedeuticit4(M1,M2), primaLezione(_,_,Settimana1,M1),
primaLezione(_, _, Settimana, M2), Settimana<=Settimana1.
:- propedeuticit4(M1,M2), secondoGiornoLezione(_, _, Settimana2, M1),
primaLezione(_, _, Settimana, M2), Settimana2<=Settimana.

```

questo permette di eliminare tutti quei modelli in cui la prima lezione del corso M2 inizia prima o nella stessa settimana della prima lezione, e dopo o la stessa settimana della seconda lezione del corso M1.

## Risultati

Le seguenti tabelle rappresentano un possibile modello restituito da clingo.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	profPresent	presentazione
venerdi	10	11	profPresent	presentazione
venerdi	11	12	muzzetto	project_management
venerdi	12	13	muzzetto	project_management
venerdi	14	15	muzzetto	project_management
venerdi	15	16	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	16	17	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	17	18	muzzetto	project_management
sabato	14	15	muzzetto	project_management
sabato	15	16	muzzetto	project_management
sabato	16	17	muzzetto	project_management
sabato	17	18	muzzetto	project_management

Figura 9: Tabella di una possibile soluzione nella prima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	muzzetto	project_management
venerdi	10	11	muzzetto	project_management
venerdi	11	12	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	12	13	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	14	15	muzzetto	project_management
venerdi	15	16	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	16	17	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	17	18	muzzetto	project_management
sabato	9	10	muzzetto	project_management
sabato	15	16	muzzetto	project_management
sabato	16	17	tomatis	la_gestione_della_qualita
sabato	17	18	tomatis	la_gestione_della_qualita

Figura 10: Tabella di una possibile soluzione nella seconda settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
venerdi	10	11	boniolo	risorse_digitali_per_il_progetto_collaborazione_e_documentazione
venerdi	11	12	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
venerdi	12	13	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
venerdi	14	15	tomatis	la_gestione_della_qualita
venerdi	15	16	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
venerdi	16	17	boniolo	risorse_digitali_per_il_progetto_collaborazione_e_documentazione
venerdi	17	18	tomatis	la_gestione_della_qualita
sabato	9	10	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
sabato	14	15	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
sabato	15	16	muzzetto	marketing_digitale
sabato	16	17	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
sabato	17	18	muzzetto	marketing_digitale

Figura 11: Tabella di una possibile soluzione nella terza settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
venerdi	10	11	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	11	12	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	12	13	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	14	15	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	15	16	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	16	17	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	17	18	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
sabato	9	10	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
sabato	14	15	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
sabato	15	16	boniolo	risorse_digitali_per_il_progetto_collaborazione_e_documentazione
sabato	16	17	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
sabato	17	18	boniolo	risorse_digitali_per_il_progetto_collaborazione_e_documentazione

Figura 12: Tabella di una possibile soluzione nella quarta settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	tomatis	la_gestione_della_qualita
venerdi	10	11	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	11	12	tomatis	la_gestione_della_qualita
venerdi	12	13	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	14	15	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
venerdi	15	16	zanchetta	acquisizione_ed_elaborazione_di_immagini_statiche_grafica
venerdi	16	17	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
venerdi	17	18	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
sabato	14	15	mazzei	progettazione_di_basi_di_dati
sabato	15	16	gena	accessibilita_e_usabilita_nella_progettazione_multimediale
sabato	16	17	mazzei	progettazione_di_basi_di_dati
sabato	17	18	gena	accessibilita_e_usabilita_nella_progettazione_multimediale

Figura 13: Tabella di una possibile soluzione nella quinta settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	vargiu	elementi_di_fotografia_digitale
venerdi	10	11	vargiu	elementi_di_fotografia_digitale
venerdi	11	12	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	12	13	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	14	15	vargiu	elementi_di_fotografia_digitale
venerdi	15	16	travostino	i_vincoli_giuridici_del_progetto_diritto_dei_media
venerdi	16	17	travostino	i_vincoli_giuridici_del_progetto_diritto_dei_media
venerdi	17	18	vargiu	elementi_di_fotografia_digitale
sabato	14	15	vargiu	elementi_di_fotografia_digitale
sabato	15	16	vargiu	elementi_di_fotografia_digitale
sabato	16	17	vargiu	elementi_di_fotografia_digitale
sabato	17	18	vargiu	elementi_di_fotografia_digitale

Figura 14: Tabella di una possibile soluzione nella sesta settimana.



Giorno	Inizio	Fine	Docente	Corso
lunedì	9	10	giordani	strumenti_e_metodi_di_interazione_nei_social_media
lunedì	10	11	travostino	i_vincoli_giuridici_del_progetto_diritto_dei_media
lunedì	11	12	giordani	strumenti_e_metodi_di_interazione_nei_social_media
lunedì	12	13	giordani	strumenti_e_metodi_di_interazione_nei_social_media
lunedì	14	15	travostino	i_vincoli_giuridici_del_progetto_diritto_dei_media
lunedì	15	16	giordani	strumenti_e_metodi_di_interazione_nei_social_media
lunedì	16	17	mazzei	progettazione_di_basi_di_dati
lunedì	17	18	mazzei	progettazione_di_basi_di_dati
martedì	9	10	boniolo	risorse_digitali_per_il_progetto_collaborazione_e_documentazione
martedì	10	11	tomatis	la_gestione_della_qualita
martedì	11	12	tomatis	la_gestione_della_qualita
martedì	12	13	travostino	i_vincoli_giuridici_del_progetto_diritto_dei_media
martedì	14	15	travostino	i_vincoli_giuridici_del_progetto_diritto_dei_media
martedì	15	16	boniolo	risorse_digitali_per_il_progetto_collaborazione_e_documentazione
martedì	16	17	boniolo	risorse_digitali_per_il_progetto_collaborazione_e_documentazione
martedì	17	18	boniolo	risorse_digitali_per_il_progetto_collaborazione_e_documentazione
mercoledì	9	10	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
mercoledì	10	11	giordani	strumenti_e_metodi_di_interazione_nei_social_media
mercoledì	11	12	mazzei	progettazione_di_basi_di_dati
mercoledì	12	13	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
mercoledì	14	15	boniolo	risorse_digitali_per_il_progetto_collaborazione_e_documentazione
mercoledì	15	16	giordani	strumenti_e_metodi_di_interazione_nei_social_media
mercoledì	16	17	boniolo	risorse_digitali_per_il_progetto_collaborazione_e_documentazione
mercoledì	17	18	mazzei	progettazione_di_basi_di_dati
giovedì	9	10	muzzetto	marketing_digitale
giovedì	10	11	valle	acquisizione_ed_elaborazione_del_suono
giovedì	11	12	valle	acquisizione_ed_elaborazione_del_suono
giovedì	12	13	mazzei	progettazione_di_basi_di_dati
giovedì	14	15	giordani	strumenti_e_metodi_di_interazione_nei_social_media
giovedì	15	16	giordani	strumenti_e_metodi_di_interazione_nei_social_media
giovedì	16	17	muzzetto	marketing_digitale
giovedì	17	18	mazzei	progettazione_di_basi_di_dati
venerdì	9	10	muzzetto	marketing_digitale
venerdì	10	11	muzzetto	marketing_digitale
venerdì	11	12	tomatis	la_gestione_della_qualita
venerdì	12	13	tomatis	la_gestione_della_qualita
venerdì	14	15	vargiu	elementi_di_fotografia_digitale
venerdì	15	16	vargiu	elementi_di_fotografia_digitale
venerdì	16	17	mazzei	progettazione_di_basi_di_dati
venerdì	17	18	mazzei	progettazione_di_basi_di_dati
sabato	14	15	mazzei	progettazione_di_basi_di_dati
sabato	15	16	mazzei	progettazione_di_basi_di_dati
sabato	16	17	mazzei	progettazione_di_basi_di_dati
sabato	17	18	mazzei	progettazione_di_basi_di_dati

Figura 15: Tabella di una possibile soluzione nella settima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	10	11	muzzetto	marketing_digitale
venerdi	11	12	nonDefinito	blocco_libero
venerdi	12	13	nonDefinito	blocco_libero
venerdi	14	15	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	15	16	muzzetto	marketing_digitale
venerdi	16	17	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	17	18	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
sabato	9	10	mazzei	progettazione_di_basi_di_dati
sabato	14	15	mazzei	progettazione_di_basi_di_dati
sabato	15	16	mazzei	progettazione_di_basi_di_dati
sabato	16	17	mazzei	progettazione_di_basi_di_dati

Figura 16: Tabella di una possibile soluzione nella ottava settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	10	11	muzzetto	marketing_digitale
venerdi	11	12	travostino	i_vincoli_giuridici_del_progetto_diritto_dei_media
venerdi	12	13	muzzetto	marketing_digitale
venerdi	14	15	travostino	i_vincoli_giuridici_del_progetto_diritto_dei_media
venerdi	15	16	pozzato	fondamenti_di_ict_e_Paradigmi_di_programmazione
venerdi	16	17	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
venerdi	17	18	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
sabato	9	10	mazzei	progettazione_di_basi_di_dati
sabato	14	15	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
sabato	15	16	mazzei	progettazione_di_basi_di_dati
sabato	17	18	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali

Figura 17: Tabella di una possibile soluzione nella nona settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	valle	acquisizione_ed_elaborazione_del_suono
venerdi	10	11	valle	acquisizione_ed_elaborazione_del_suono
venerdi	11	12	valle	acquisizione_ed_elaborazione_del_suono
venerdi	12	13	valle	acquisizione_ed_elaborazione_del_suono
venerdi	14	15	giordani	strumenti_e_metodi_di_interazione_nei_social_media
venerdi	15	16	gena	linguaggi_di_markup
venerdi	16	17	giordani	strumenti_e_metodi_di_interazione_nei_social_media
venerdi	17	18	gena	linguaggi_di_markup
sabato	9	10	valle	acquisizione_ed_elaborazione_del_suono
sabato	14	15	gena	linguaggi_di_markup
sabato	15	16	gena	linguaggi_di_markup
sabato	16	17	gena	linguaggi_di_markup
sabato	17	18	valle	acquisizione_ed_elaborazione_del_suono

Figura 18: Tabella di una possibile soluzione nella decima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
venerdi	10	11	gena	linguaggi_di_markup
venerdi	11	12	travostino	i_vincoli_giuridici_del_progetto_diritto_dei_media
venerdi	12	13	travostino	i_vincoli_giuridici_del_progetto_diritto_dei_media
venerdi	14	15	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
venerdi	15	16	gena	linguaggi_di_markup
venerdi	16	17	gena	linguaggi_di_markup
venerdi	17	18	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
sabato	9	10	gena	linguaggi_di_markup
sabato	14	15	gena	linguaggi_di_markup
sabato	15	16	gena	linguaggi_di_markup
sabato	16	17	valle	acquisizione_ed_elaborazione_del_suono
sabato	17	18	valle	acquisizione_ed_elaborazione_del_suono

Figura 19: Tabella di una possibile soluzione nella undicesima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
venerdi	10	11	nonDefinito	blocco_libero
venerdi	11	12	nonDefinito	blocco_libero
venerdi	12	13	gribaudo	grafica_3d
venerdi	14	15	gena	linguaggi_di_markup
venerdi	15	16	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
venerdi	16	17	gribaudo	grafica_3d
venerdi	17	18	gena	linguaggi_di_markup
sabato	9	10	gena	linguaggi_di_markup
sabato	14	15	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
sabato	15	16	gena	linguaggi_di_markup
sabato	16	17	gena	linguaggi_di_markup
sabato	17	18	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali

Figura 20: Tabella di una possibile soluzione nella dodicesima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	giordani	strumenti_e_metodi_di_interazione_nei_social_media
venerdi	10	11	giordani	strumenti_e_metodi_di_interazione_nei_social_media
venerdi	11	12	giordani	strumenti_e_metodi_di_interazione_nei_social_media
venerdi	12	13	giordani	strumenti_e_metodi_di_interazione_nei_social_media
venerdi	14	15	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
venerdi	15	16	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
venerdi	16	17	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
venerdi	17	18	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
sabato	9	10	gribaudo	grafica_3d
sabato	14	15	terranova	progettazione_grafica_e_design_di_interfacce
sabato	15	16	gribaudo	grafica_3d
sabato	16	17	terranova	progettazione_grafica_e_design_di_interfacce
sabato	17	18	terranova	progettazione_grafica_e_design_di_interfacce

Figura 21: Tabella di una possibile soluzione nella tredicesima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	terranova	progettazione_grafica_e_design_di_interfacce
venerdi	10	11	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
venerdi	11	12	terranova	progettazione_grafica_e_design_di_interfacce
venerdi	12	13	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
venerdi	14	15	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
venerdi	15	16	santangelo	semiologia_e_multimedialita
venerdi	16	17	santangelo	semiologia_e_multimedialita
venerdi	17	18	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
sabato	9	10	gribaudo	grafica_3d
sabato	14	15	gribaudo	grafica_3d
sabato	15	16	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
sabato	16	17	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
sabato	17	18	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali

Figura 22: Tabella di una possibile soluzione nella quattordicesima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	zanchetta	tecniche_e_strumenti_di_marketing_digitale
venerdi	10	11	zanchetta	tecniche_e_strumenti_di_marketing_digitale
venerdi	11	12	gribaudo	grafica_3d
venerdi	12	13	gena	linguaggi_di_markup
venerdi	14	15	gena	linguaggi_di_markup
venerdi	15	16	zanchetta	tecniche_e_strumenti_di_marketing_digitale
venerdi	16	17	gribaudo	grafica_3d
venerdi	17	18	zanchetta	tecniche_e_strumenti_di_marketing_digitale
sabato	9	10	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
sabato	14	15	ghidelli	acquisizione_ed_elaborazione_di_sequenze_di_immagini_digitali
sabato	16	17	terranova	progettazione_grafica_e_design_di_interfacce
sabato	17	18	terranova	progettazione_grafica_e_design_di_interfacce

Figura 23: Tabella di una possibile soluzione nella quindicesima settimana.

Giorno	Inizio	Fine	Docente	Corso
lunedì	9	10	gena	linguaggi_di_markup
lunedì	10	11	gena	linguaggi_di_markup
lunedì	11	12	suppini	introduzione_al_social_media_management
lunedì	12	13	suppini	introduzione_al_social_media_management
lunedì	14	15	santangelo	semiologia_e_multimedialita
lunedì	15	16	suppini	introduzione_al_social_media_management
lunedì	16	17	suppini	introduzione_al_social_media_management
lunedì	17	18	santangelo	semiologia_e_multimedialita
martedì	9	10	gribaudo	grafica_3d
martedì	10	11	gribaudo	grafica_3d
martedì	11	12	gribaudo	grafica_3d
martedì	12	13	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
martedì	14	15	suppini	introduzione_al_social_media_management
martedì	15	16	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
martedì	16	17	gribaudo	grafica_3d
martedì	17	18	suppini	introduzione_al_social_media_management
mercoledì	9	10	gribaudo	grafica_3d
mercoledì	10	11	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
mercoledì	11	12	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
mercoledì	12	13	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
mercoledì	14	15	gribaudo	grafica_3d
mercoledì	15	16	gribaudo	grafica_3d
mercoledì	16	17	gribaudo	grafica_3d
mercoledì	17	18	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
giovedì	9	10	gribaudo	grafica_3d
giovedì	10	11	gribaudo	grafica_3d
giovedì	11	12	suppini	introduzione_al_social_media_management
giovedì	12	13	suppini	introduzione_al_social_media_management
giovedì	14	15	suppini	introduzione_al_social_media_management
giovedì	15	16	suppini	introduzione_al_social_media_management
giovedì	16	17	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
giovedì	17	18	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
venerdì	9	10	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
venerdì	10	11	taddeo	crossmedia_articolazione_delle_scritture_multimediali
venerdì	11	12	taddeo	crossmedia_articolazione_delle_scritture_multimediali
venerdì	12	13	taddeo	crossmedia_articolazione_delle_scritture_multimediali
venerdì	14	15	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
venerdì	15	16	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
venerdì	16	17	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
venerdì	17	18	taddeo	crossmedia_articolazione_delle_scritture_multimediali
sabato	14	15	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
sabato	15	16	santangelo	semiologia_e_multimedialita
sabato	16	17	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
sabato	17	18	santangelo	semiologia_e_multimedialita

Figura 24: Tabella di una possibile soluzione nella sedicesima settimana.



Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
venerdi	10	11	santangelo	semiologia_e_multimedialita
venerdi	11	12	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
venerdi	12	13	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
venerdi	14	15	santangelo	semiologia_e_multimedialita
venerdi	15	16	suppini	introduzione_al_social_media_management
venerdi	16	17	suppini	introduzione_al_social_media_management
venerdi	17	18	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
sabato	9	10	zanchetta	tecniche_e_strumenti_di_marketing_digitale
sabato	14	15	zanchetta	tecniche_e_strumenti_di_marketing_digitale
sabato	15	16	gribaudo	grafica_3d
sabato	16	17	gribaudo	grafica_3d
sabato	17	18	zanchetta	tecniche_e_strumenti_di_marketing_digitale

Figura 25: Tabella di una possibile soluzione nella diciassettesima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	damiano	tecnologie_server_side_per_il_web
venerdi	10	11	damiano	tecnologie_server_side_per_il_web
venerdi	11	12	damiano	tecnologie_server_side_per_il_web
venerdi	12	13	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
venerdi	14	15	santangelo	semiologia_e_multimedialita
venerdi	15	16	damiano	tecnologie_server_side_per_il_web
venerdi	16	17	micalizio	ambienti_di_sviluppo_e_linguaggi_client_side_per_il_web
venerdi	17	18	santangelo	semiologia_e_multimedialita
sabato	14	15	lombardo	la_gestione_delle_risorse_umane
sabato	15	16	lombardo	la_gestione_delle_risorse_umane
sabato	16	17	taddeo	crossmedia_articolazione_delle_scritture_multimediali
sabato	17	18	taddeo	crossmedia_articolazione_delle_scritture_multimediali

Figura 26: Tabella di una possibile soluzione nella diciottesima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	lombardo	la_gestione_delle_risorse_umane
venerdi	10	11	lombardo	la_gestione_delle_risorse_umane
venerdi	11	12	terranova	progettazione_grafica_e_design_di_interfacce
venerdi	12	13	terranova	progettazione_grafica_e_design_di_interfacce
venerdi	14	15	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
venerdi	15	16	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
venerdi	16	17	terranova	progettazione_grafica_e_design_di_interfacce
venerdi	17	18	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
sabato	9	10	zanchetta	tecniche_e_strumenti_di_marketing_digitale
sabato	14	15	zanchetta	tecniche_e_strumenti_di_marketing_digitale
sabato	15	16	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
sabato	16	17	zanchetta	tecniche_e_strumenti_di_marketing_digitale
sabato	17	18	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica

Figura 27: Tabella di una possibile soluzione nella diciannovesima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	suppini	introduzione_al_social_media_management
venerdi	10	11	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
venerdi	11	12	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
venerdi	12	13	suppini	introduzione_al_social_media_management
venerdi	14	15	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
venerdi	15	16	pozzato	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i
venerdi	16	17	pozzato	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i
venerdi	17	18	pozzato	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i
sabato	9	10	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica
sabato	14	15	taddeo	crossmedia_articolazione_delle_scritture_multimediali
sabato	15	16	taddeo	crossmedia_articolazione_delle_scritture_multimediali
sabato	16	17	taddeo	crossmedia_articolazione_delle_scritture_multimediali
sabato	17	18	gabardi	comunicazione_pubblicitaria_e_comunicazione_pubblica

Figura 28: Tabella di una possibile soluzione nella ventesima settimana.



Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	lombardo	la_gestione_delle_risorse_umane
venerdi	10	11	damiano	tecnologie_server_side_per_il_web
venerdi	11	12	lombardo	la_gestione_delle_risorse_umane
venerdi	12	13	taddeo	crossmedia_articolazione_delle_scritture_multimediali
venerdi	14	15	damiano	tecnologie_server_side_per_il_web
venerdi	15	16	taddeo	crossmedia_articolazione_delle_scritture_multimediali
venerdi	16	17	taddeo	crossmedia_articolazione_delle_scritture_multimediali
venerdi	17	18	taddeo	crossmedia_articolazione_delle_scritture_multimediali
sabato	9	10	pozzato	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i
sabato	14	15	pozzato	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i
sabato	15	16	taddeo	crossmedia_articolazione_delle_scritture_multimediali
sabato	16	17	taddeo	crossmedia_articolazione_delle_scritture_multimediali
sabato	17	18	taddeo	crossmedia_articolazione_delle_scritture_multimediali

Figura 29: Tabella di una possibile soluzione nella ventunesima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	taddeo	crossmedia_articolazione_delle_scritture_multimediali
venerdi	10	11	taddeo	crossmedia_articolazione_delle_scritture_multimediali
venerdi	11	12	damiano	tecnologie_server_side_per_il_web
venerdi	12	13	damiano	tecnologie_server_side_per_il_web
venerdi	14	15	pozzato	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i
venerdi	15	16	pozzato	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i
venerdi	16	17	damiano	tecnologie_server_side_per_il_web
venerdi	17	18	pozzato	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i
sabato	9	10	pozzato	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i
sabato	15	16	taddeo	crossmedia_articolazione_delle_scritture_multimediali
sabato	16	17	pozzato	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_i
sabato	17	18	taddeo	crossmedia_articolazione_delle_scritture_multimediali

Figura 30: Tabella di una possibile soluzione nella ventiduesima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	damiano	tecnologie_server_side_per_il_web
venerdi	10	11	schifanella	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii
venerdi	11	12	damiano	tecnologie_server_side_per_il_web
venerdi	12	13	damiano	tecnologie_server_side_per_il_web
venerdi	14	15	schifanella	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii
venerdi	15	16	lombardo	la_gestione_delle_risorse_umane
venerdi	16	17	schifanella	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii
venerdi	17	18	lombardo	la_gestione_delle_risorse_umane
sabato	9	10	schifanella	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii
sabato	14	15	damiano	tecnologie_server_side_per_il_web
sabato	15	16	damiano	tecnologie_server_side_per_il_web
sabato	16	17	schifanella	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii
sabato	17	18	schifanella	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii

Figura 31: Tabella di una possibile soluzione nella ventitresima settimana.

Giorno	Inizio	Fine	Docente	Corso
venerdi	9	10	lombardo	la_gestione_delle_risorse_umane
venerdi	10	11	lombardo	la_gestione_delle_risorse_umane
venerdi	11	12	schifanella	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii
venerdi	12	13	damiano	tecnologie_server_side_per_il_web
venerdi	14	15	schifanella	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii
venerdi	15	16	schifanella	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii
venerdi	16	17	damiano	tecnologie_server_side_per_il_web
venerdi	17	18	schifanella	progettazione_e_sviluppo_di_applicazioni_web_su_dispositivi_mobile_ii
sabato	14	15	damiano	tecnologie_server_side_per_il_web
sabato	15	16	damiano	tecnologie_server_side_per_il_web
sabato	16	17	damiano	tecnologie_server_side_per_il_web
sabato	17	18	damiano	tecnologie_server_side_per_il_web

Figura 32: Tabella di una possibile soluzione nella ventiquatresima settimana.

## Tempi

Time	2450.030
CPU Time	2442.362

Tabella 3: Prestazioni (in secondi)

La versione finale richiede un tempo discreto per restituire una soluzione, come è possibile visionare direttamente dalla tabella sopra indicata. Ciò che è stato importante notare durante la codifica di tale CSP, è come cambiando alcuni vincoli o alcune scelte implementative, la soluzione, anche se restituita in ogni caso, richiedeva un tempo di gran lunga superiore a quello ottenuto con la descrizione implementativa su presentata. Ad esempio in una versione precedente si era pensato di creare un predicato che contenesse tutti i possibili slot assegnabili e successivamente si chiedeva di assegnare 1 e 1 solo corso per ogni possibile slot assegnabile, in questo modo non si necessitava del vincolo di più lezioni nella stessa ora, ma in alcuni casi non assegnava alcuna ora per uno o più insegnamenti. Questo era risolto da un ulteriore vincolo che imponesse di eliminare tutti quei modelli in cui il conteggio dei corsi assegnati era inferiore al numero effettivo di corsi da assegnare. Formalmete:

```
8 { slot(Giorno, Inizio, Fine, Settimana) : orario(Inizio, Fine)} 8 :-  
intero(Giorno), settimana(Settimana).
```

Assegna per ogni giorno intero 8 ore, per tutte le settimane.

```
4 { slot(Giorno, Inizio, Fine, Settimana) : orario(Inizio, Fine)} 5 :-  
ridotto(Giorno), settimana(Settimana)
```

Assegna per ogni giorno ridotto (sabato) da 4 a 5 ore, per tutte le settimane.

```
assegnabili(Giorno, Inizio, Settimana):- slot(Giorno, Inizio, _, Settimana),  
not full(Settimana), not week(Giorno), not riservata(Giorno, Inizio, _, Settimana).
```

Elimina da `slot` tutte le occorrenze di `Giorno` che non fanno parte del predicato `week(Giorno)` (ovvero venerdì e sabato) per le settimane non piene e che non sono riservati (per presentazione o blocco da lasciare libero).

```
assegnabili(Giorno, Inizio, Settimana):- slot(Giorno, Inizio, _, Settimana),  
full(Settimana), giorno(Giorno), not riservata(Giorno, Inizio, _, Settimana).
```

Elimina da `slot` tutti i giorni delle settimane piene ovvero la 7 e la 16 che sono riservati (per presentazione o blocco da lasciare libero).

```
1 { lezione(Giorno, Inizio, Fine, Corso, Settimana, Docente) :  
insegnamento(Corso, Docente, _) } 1 :-  
assegnabili(Giorno,Inizio,Settimana), orario(Inizio,Fine).
```

in cui viene effettivamente viene richiesta l'assegnazione di un `Corso` per ogni occorrenza del predicato `assegnabili(Giorno,Inizio,Settimana)`.

Tale scelta implementativa, unita ai vincoli presentati nella sezione precedente, anche se restituiva una soluzione e rispettava tutti i vincoli richiesti, richiedeva un tempo di elaborazione superiore alle 7 ore. In virtù di ciò, si è ri-progettata la soluzione in modo tale che potesse soddisfare gli stessi vincoli, con una complessità temporale minore.