Generazione, ottimizzazione e gestione dell'orario delle lezioni (GOGOL)

Francesca Rossi

frossi@math.unipd.it

Scopo

Cotruire un sistema software per la generazione e la gestione dell'orario delle lezioni di un corso di studi universitario, comprensivo di un ambiente configurabile basato su web e dei manuali d'uso del sistema

Utenti

- Presidente consiglio dei corsi di studio
- Personale segreteria didattica
- Singoli docenti

Azioni disponibili

- Inserimento dati del corso di laurea
- Inserimento vincoli docenti
- Inserimento vincoli struttura orario e ripartizione corsi su anni/periodi
- Generazione dell'orario di un trimestre
- Generazione suggerimenti per eventuali inconsistenze
- Gestione modifiche e generazione nuovo orario

Vincoli e preferenze

- Da considerare in modo diverso
  - I vincoli vanno soddisfatti comunque
  - Le preferenze vanno soddisfatte il piu' possibile
- Se si considerano tutti come vincoli, di solito nessuna soluzione

Inserimento dati dei corsi (segr. did.)

- Per ogni corso:
  - Anno
    - Vincolo: corsi obbligatori dello stesso anno non devono sovrapporsi
  - Periodo
    - Ogni periodo e' indipendente dall'altro
  - Nome docente/i
    - Vincolo: lo stesso docente non puo' avere lezioni diverse nello stesso orario
  - Stato (obbligatorio, opzionale)
    - Preferenza: gli opzionali possono sovrapporsi, anche se e' preferibile di no
  - Ore in aula o laboratorio
    - Sono tipologie di aule da gestire separatamente
  - Stima numero studenti
    - Vincolo: non si possono allocare corsi in aule con capienza minore della stima

Inserimento struttura orario (segr. did.)

- Anno accademico
  - Periodi di lezione, giorni di vacanza, ecc.
- Linee guida
  - Vincolo/preferenza: esempio: I anno mattina, II anno pomeriggio, III anno mattina (il piu' possibile)
  - Preferenza: buchi piu' piccoli possibili (orario il piu' compatto possibile)
- Ore da usare durante la giornata
  - Vincolo: 9:30-17:30 senza pause, ore di 60'
  - Preferenza: lezioni di 2 ore

Inserimento dati aule (segr. did.)

- Aule disponibili con
  - Capienza (vincolo)
  - orario/periodi di non disponibilita’ (vincolo)

Inserimento dati docenti (singoli docenti)

- Per ogni docente:
- Giorni/ore di indisponibilita’
  - Ad esempio per altri insegnamenti nello stesso trimestre
  - Vincolo: il docente non puo’ avere lezioni in quei giorni/ore
  - Motivazione
- Preferenze su orari e giorni
  - Esempio: Venerdi’ pomeriggio, …
  - Motivazione
- Abilitare accesso fino ad una certa data
  - Considerare vincoli e preferenze inseriti entro la data

Generazione orario di un periodo

- Per ogni periodo, generare l'orario con tutti i corsi del periodo, tale che
  - Tutti i vincoli siano soddisfatti
  - Le preferenze siano soddisfatte il piu' possibile
- Se non e' possibile soddisfare tutti i vincoli, il sistema deve
  - Suggerire modifiche o vincoli da rilassare
  - Proporre soluzioni subottime

Sistema Web per l'inserimento dati

- Pagine web con form per inserire i dati
- Menu a tendina per evitare di inserire dati errati
  - Basati sui dati della segreteria didattica
- Accesso agli utenti con diritti diversi
  - Sempre e dovunque alla segreteria didattica e al presidente CCS
  - Solo sui propri dati e in certi periodi ai singoli docenti
  - Gestione account utenti

Base di dati e modifiche incrementali

- Deve contenere tutti i dati
  - Corsi
  - Docenti
  - Corso di laurea
- Inserimento anche in piu' fasi
- Correzione dati errati
- Aggiunta di vincoli ad una soluzione proposta
  - Nuove indisponibilita', lezioni fissate in certe ore, …

Output del sistema

- Orario in formato pdf e/o html
- Formattazione leggibile facilmente

Come generare un orario?

- Soddisfare i vincoli sempre, le preferenze al meglio
- Problema di ottimizzazione di vincoli
  - Alcuni orari sono meglio di altri
  - Ottimizzazione multi-obbiettivo
- Due metodi principali di ricerca nello spazio delle soluzioni
  - Ricerca sistematica + euristiche
  - Ricerca locale

Branch and bound

- Costruisce una soluzione settando una variabile alla volta
- Tiene da parte soluzione ottima trovata finora, e suo valore
  - Esempi di valori: numero di preferenze soddisfatte, somma pesata sulle preferenze
- Ogni soluzione parziale viene confrontata con la ottima finora, e si va avanti solo se c'e' speranza di trovare una soluzione migliore
  - Sovrastima dei valori delle soluzioni complete a partire da una soluzione parziale
- Altrimenti ritratta l'ultima decisione, e prova un altro settaggio per l'ultima variabile
- Arriva sempre ad una soluzione ottima
- In collegamento con corso di Ricerca Operativa

Ricerca locale

- Costruisce una soluzione (orario) iniziando da una soluzione a caso
- Ad ogni passo, effettua una piccola modifica della soluzione corrente, in modo da ottenere una soluzione migliore
  - Esempio di “piccola modifica”: modificare il valore di una sola variabile
- Si ferma quando non c’e’ modo (tra quelli previsti) di migliorare
- Puo’ non arrivare ad una soluzione ottima

Info su Sistemi/software esistenti

- Sistema/librerie basate su ricerca locale
  - Easylocal (Andrea Schaerf, Univ. Udine)
  - http://www.diegm.uniud.it/satt/projects.php
- Un articolo su ricerca locale per problemi di orario
  - http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01696421
  - Prima trova una soluzione che soddisfa tutti I vincoli, poi la migliore con la ricerca locale cercando di soddisfare le preferenze
- Software, datasets, articoli, tesi, ecc. sul problema dell'orario
  - http://www.asap.cs.nott.ac.uk/ASAP/watt/resources/university.html