Seminario per il corso di Linguaggi di nuova concezione

UN CASO DI STUDIO PER CLP(FD): ROSTERING OSPEDALIERO

RAFFAELE CIPRIANO

22 Maggio 2007

Struttura del seminario

- *Il problema reale*
- *Modello decisionale in CLP(FD)*
- *Funzione obiettivo in CLP(FD)*
- *Miglioriamo ancora: La Ricerca Locale*
- *Alcuni accorgimenti pratici*
- *Sviluppi futuri*
- *Domande*

Il problema reale

- *Il rostering ospedaliero*

Il problema reale

Il rostering ospedaliero

- Il *self-scheduling* ...

Il problema reale

- ***Il rostering ospedaliero***
  - Il *self-scheduling* ...
- ***Requisiti del sistema***
  - **Input:** caratteristiche del mese
    - **struttura del mese**
    - **ferie**
    - **turni particolari**

Il problema reale

Il rostering ospedaliero

- Il *self-scheduling* ...

Requisiti del sistema

- **Input:** caratteristiche del mese
  - **struttura del mese**
  - **ferie**
  - **turni particolari**
- **Modello:** conoscenza dell'organizzazione del reparto
  - **suddivisione dei turni**
  - **esigenze di servizio**
  - **regole contrattuali**
  - **caratteristiche del personale**

Il problema reale

- **Output:** orario con le seguenti caratteristiche
  - copertura dei turni
  - competenze corrette
  - ferie soddisfatte
  - norme lavorative rispettate
  - lavoro bilanciato nel reparto
  - lavoro bilanciato per ciascun medico

Il problema reale

- **Output:** orario con le seguenti caratteristiche
  - **copertura dei turni**
  - **competenze corrette**
  - **ferie soddisfatte**
  - **norme lavorative rispettate**
  - **lavoro bilanciato nel reparto**
  - **lavoro bilanciato per ciascun medico**

Obiettivi

- trovare soluzione ammissibile (CSP)

Il problema reale

- **Output:** orario con le seguenti caratteristiche
  - **copertura dei turni**
  - **competenze corrette**
  - **ferie soddisfatte**
  - **norme lavorative rispettate**
  - **lavoro bilanciato nel reparto**
  - **lavoro bilanciato per ciascun medico**

Obiettivi

- trovare soluzione ammissibile (CSP)
- trovare soluzione ammissibile buona (COP)

Il problema reale

- **Output:** orario con le seguenti caratteristiche
  - **copertura dei turni**
  - **competenze corrette**
  - **ferie soddisfatte**
  - **norme lavorative rispettate**
  - **lavoro bilanciato nel reparto**
  - **lavoro bilanciato per ciascun medico**

Obiettivi

- trovare soluzione ammissibile (CSP)
- trovare soluzione ammissibile buona (COP)
- trovare soluzione ammissibile buona in poco tempo!

Modello decisionale CLP(FD)

- *Costruiamo un modello per il Constrain Satisfaction Problem (CSP) relativo alle esigenze del reparto. Useremo:*

Modello decisionale CLP(FD)

- ***Costruiamo un modello per il Constrain Satisfaction Problem (CSP) relativo alle esigenze del reparto. Useremo:***
  - SICStus Prolog
    - **in particolare la libreria CLP(FD)**

Modello decisionale CLP(FD)

- ***Costruiamo un modello per il Constrain Satisfaction Problem (CSP) relativo alle esigenze del reparto. Useremo:***
  - SICStus Prolog
    - **in particolare la libreria CLP(FD)**
  - La filosofia Constraint & Generate:
    - **Scelta delle variabili**
    - **Impostazione dei domini delle variabili**
    - **Impostazione dei vincoli sulle variabili**
    - **Assegnamento dei valori alle variabili**

Modello decisionale CLP(FD)

Scelta delle variabili

Supponiamo di avere...

- 7 giorni da coprire (1-7)
- 5 medici a disposizione (dr. Jekill, dr. Jones, dr. House, dr. Freud, dr. Watson)
- 3 turni possibili (U, R, E)

Idee?

Modello decisionale CLP(FD)

Scelta delle variabili

Supponiamo di avere...

- 7 giorni da coprire (1-7)
- 5 medici a disposizione (dr. Jekill, dr. Jones, dr. House, dr. Freud, dr. Watson)
- 3 turni possibili (U, R, E)

Idee?

DOM: Dr.	1	2	3	4	5	6	7
U							
R							
E							

Modello decisionale CLP(FD)

DOM: Turno	1	2	3	4	5	6	7
Dr. Jekill							
Dr. Jones							
Dr. House							
Dr. Freud							
Dr. Watson							

Modello decisionale CLP(FD)

DOM: Turno	1	2	3	4	5	6	7
Dr. Jekill							
Dr. Jones							
Dr. House							
Dr. Freud							
Dr. Watson							

	1			2			3			4			5			6			7	
DOM: bool	U	R	E	U	R	E	U	R	E	U	R	E	U	R	E	U	R	E	U	R
Dr. Jekill																				
Dr. Jones																				
Dr. House																				
Dr. Freud																				
Dr. Watson																				

Modello decisionale CLP(FD)

	1				2				...	31			
0/1	T_1	T_2	...	T_k	T_1	T_2	...	T_k	...	T_1	T_2	...	T_k
MedicoA									...				
MedicoB				$X_{B,k,1}$				$X_{B,k,2}$	...				
...	...	...	...	...	...	...	...	...	...	...	...	...	...
MedicoM									...				

Modello decisionale CLP(FD)

	1				2				...	31			
0/1	T_1	T_2	...	T_k	T_1	T_2	...	T_k	...	T_1	T_2	...	T_k
MedicoA									...				
MedicoB				$X_{B,k,1}$				$X_{B,k,2}$	...				
...	...	...	...	...	...	...	...	...	...	...	...	...	...
MedicoM									...				

- ***Impostazione dei domini delle variabili***
  - Variabili booleane

Modello decisionale CLP(FD)

	1				2				...	31			
0/1	T_1	T_2	...	T_k	T_1	T_2	...	T_k	...	T_1	T_2	...	T_k
MedicoA									...				
MedicoB				$X_{B,k,1}$				$X_{B,k,2}$	...				
...	...	...	...	...	...	...	...	...	...	...	...	...	...
MedicoM									...				

- ***Impostazione dei domini delle variabili***
  - Variabili booleane
  - Le variabili vengono impostate a 0 per medici non competenti sul turno o in ferie

Modello decisionale CLP(FD)

Impostazione dei vincoli sulle variabili

	1				2				...	31			
	T_1	T_2	...	T_k	T_1	T_2	...	T_k	...	T_1	T_2	...	T_k
MedicoA									...				
MedicoB				$X_{B,k,1}$				$X_{B,k,2}$	...				
...	...	...	...	...	...	...	...	...	...	...	...	...	...
MedicoM									...				

Modello decisionale CLP(FD)

Impostazione dei vincoli sulle variabili

	1				2				...	31			
	T_1	T_2	...	T_k	T_1	T_2	...	T_k	...	T_1	T_2	...	T_k
MedicoA									...				
MedicoB				$X_{B,k,1}$				$X_{B,k,2}$	...				
...	...	...	...	...	...	...	...	...	...	...	...	...	...
MedicoM									...				

- copertura dei turni - per colonna

Modello decisionale CLP(FD)

Impostazione dei vincoli sulle variabili

	1				2				...	31			
	T_1	T_2	...	T_k	T_1	T_2	...	T_k	...	T_1	T_2	...	T_k
MedicoA									...				
MedicoB				$X_{B,k,1}$				$X_{B,k,2}$	...				
...	...	...	...	...	...	...	...	...	...	...	...	...	...
MedicoM									...				

- copertura dei turni - per colonna
- norme lavorative rispettate - per porzioni di riga

Modello decisionale CLP(FD)

Impostazione dei vincoli sulle variabili

	1				2				...	31			
	T_1	T_2	...	T_k	T_1	T_2	...	T_k	...	T_1	T_2	...	T_k
MedicoA									...				
MedicoB				$X_{B,k,1}$				$X_{B,k,2}$	...				
...	...	...	...	...	...	...	...	...	...	...	...	...	...
MedicoM									...				

- copertura dei turni - per colonna
- norme lavorative rispettate - per porzioni di riga
- lavoro bilanciato per ciascun medico - per riga

Modello decisionale CLP(FD)

- *Assegnamento dei valori alle variabili*

Modello decisionale CLP(FD)

- ***Assegnamento dei valori alle variabili***
  - Ordine di assegnamento

Modello decisionale CLP(FD)

- ***Assegnamento dei valori alle variabili***
  - Ordine di assegnamento
    - **Concateniamo le liste . . . cosa succederà?**

Modello decisionale CLP(FD)

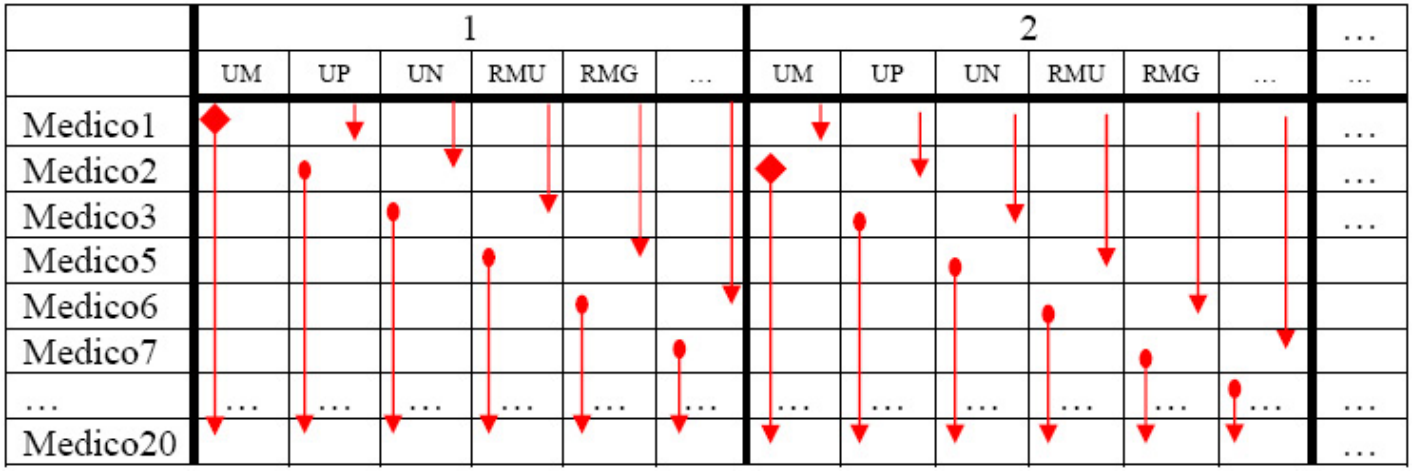
- ***Assegnamento dei valori alle variabili***
  - Ordine di assegnamento
    - **Concateniamo le liste . . . cosa succederà?**
    - **idee?**

Modello decisionale CLP(FD)

Assegnamento dei valori alle variabili

Ordine di assegnamento

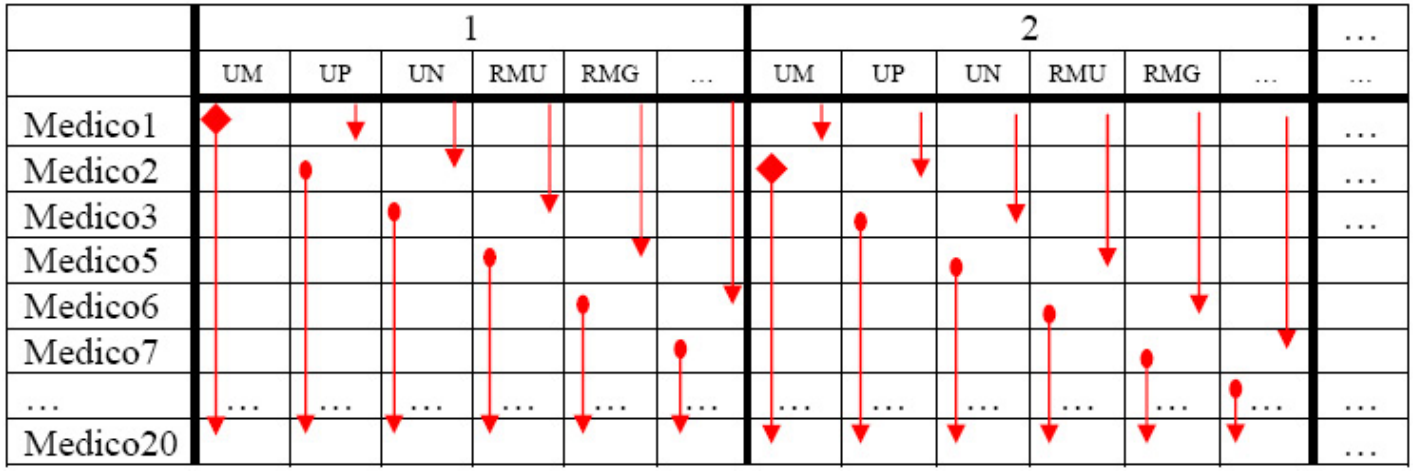
- **Concateniamo le liste . . . cosa succederà?**
- **idee?**
- **Andiamo in diagonale con i dovuti accorgimenti...**



Modello decisionale CLP(FD)

Assegnamento dei valori alle variabili

- Ordine di assegnamento
  - **Concateniamo le liste . . . cosa succederà?**
  - **idee?**
  - **Andiamo in diagonale con i dovuti accorgimenti...**

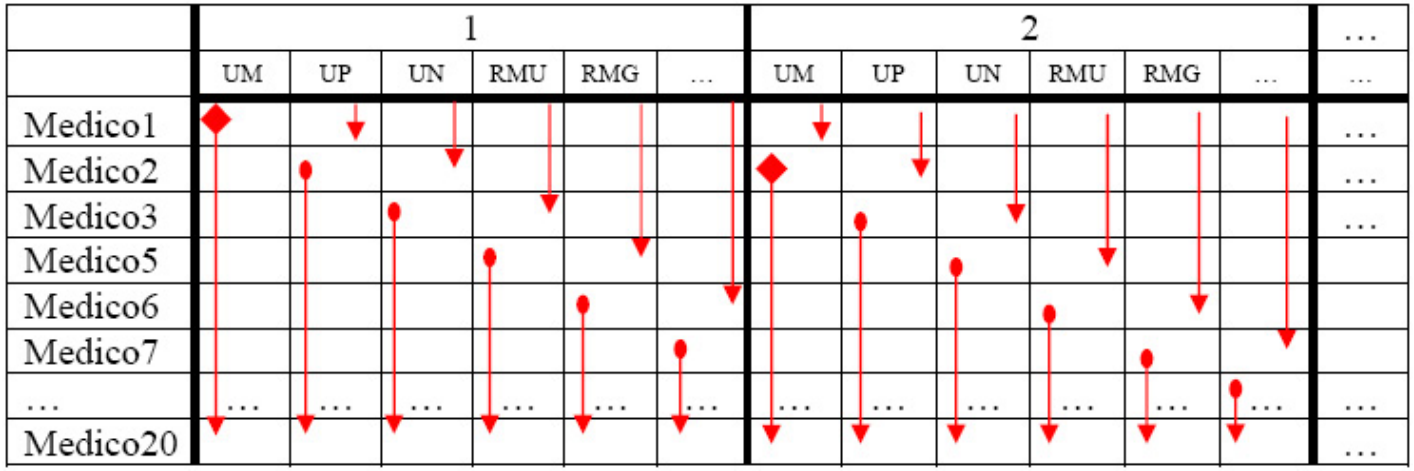


- Valore da assegnare (proviamo prima 0 o prima 1?)

Modello decisionale CLP(FD)

Assegnamento dei valori alle variabili

- Ordine di assegnamento
  - **Concateniamo le liste . . . cosa succederà?**
  - **idee?**
  - **Andiamo in diagonale con i dovuti accorgimenti...**

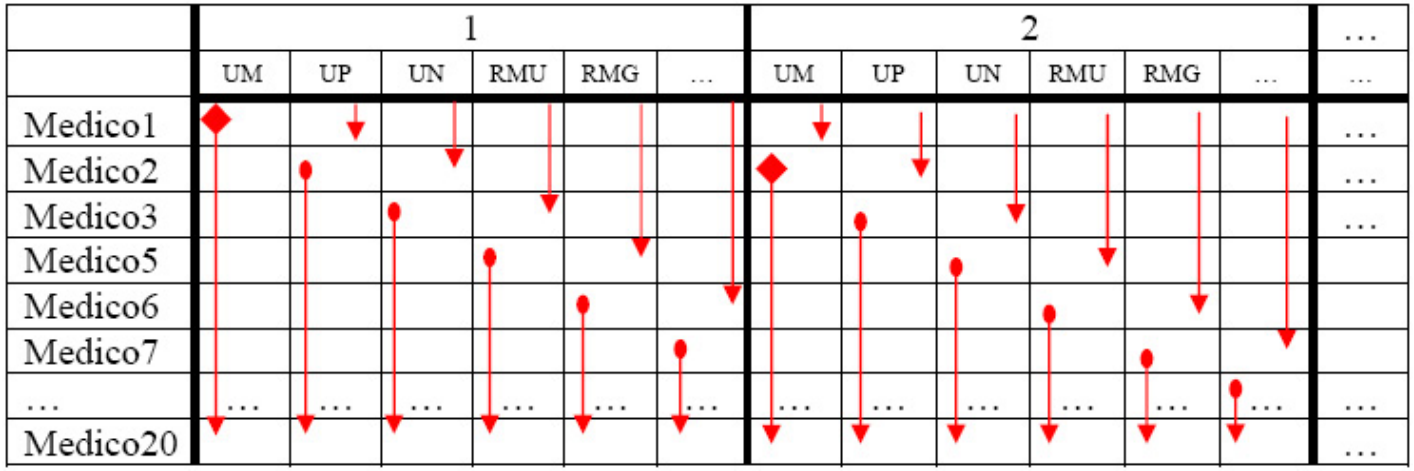


- Valore da assegnare (proviamo prima 0 o prima 1?)
  - **Se partiamo prima da 0 . . .**

Modello decisionale CLP(FD)

Assegnamento dei valori alle variabili

- Ordine di assegnamento
  - **Concateniamo le liste . . . cosa succederà?**
  - **idee?**
  - **Andiamo in diagonale con i dovuti accorgimenti...**



- Valore da assegnare (proviamo prima 0 o prima 1?)
  - **Se partiamo prima da 0 . . .**
  - **Se partiamo prima da 1 . . .**

Modello decisionale CLP(FD)

- *Abbiamo il nostro modello in CLP(FD) per il CSP, lo lanciamo...*

Modello decisionale CLP(FD)

- ***Abbiamo il nostro modello in CLP(FD) per il CSP, lo lanciamo...***
  - Il programma dà errore mentre pone dei vincoli ... vincoli troppo restrittivi!

Modello decisionale CLP(FD)

- ***Abbiamo il nostro modello in CLP(FD) per il CSP, lo lanciamo...***
  - Il programma dà errore mentre pone dei vincoli ... vincoli troppo restrittivi!
  - Il programma trova una soluzione in pochi secondi ... ok!

Modello decisionale CLP(FD)

- ***Abbiamo il nostro modello in CLP(FD) per il CSP, lo lanciamo...***
  - Il programma dà errore mentre pone dei vincoli ... vincoli troppo restrittivi!
  - Il programma trova una soluzione in pochi secondi ... ok!
  - Il labeling diverge...concludiamo che non esiste soluzione e il sistema non è in grado di accorgersene in tempi umani...continuerà ad esplorare lo spazio delle soluzioni!

Modello decisionale CLP(FD)

- ***Abbiamo il nostro modello in CLP(FD) per il CSP, lo lanciamo...***
  - Il programma dà errore mentre pone dei vincoli ... vincoli troppo restrittivi!
  - Il programma trova una soluzione in pochi secondi ... ok!
  - Il labeling diverge...concludiamo che non esiste soluzione e il sistema non è in grado di accorgersene in tempi umani...continuerà ad esplorare lo spazio delle soluzioni!
- ***Com'è la soluzione?***

Funzione obiettivo CLP(FD)

- ***Arricchiamo il modello con una FObj per i soft-constraint per:***
  - Equilibrare tra i medici i turni sgraditi
  - Evitare turni doppi (mattina e pomeriggio)
  - Evitare turni uguali consecutivi

Funzione obiettivo CLP(FD)

- ***Arricchiamo il modello con una FObj per i soft-constraint per:***
  - Equilibrare tra i medici i turni sgraditi
  - Evitare turni doppi (mattina e pomeriggio)
  - Evitare turni uguali consecutivi
- ***Cosa ne facciamo?***

Funzione obiettivo CLP(FD)

- ***Arricchiamo il modello con una FObj per i soft-constraint per:***
  - Equilibrare tra i medici i turni sgraditi
  - Evitare turni doppi (mattina e pomeriggio)
  - Evitare turni uguali consecutivi
- ***Cosa ne facciamo?***
  - `labeling( [down, minimize(FObj)], Vars ).`

Funzione obiettivo CLP(FD)

- ***Arricchiamo il modello con una FObj per i soft-constraint per:***
  - Equilibrare tra i medici i turni sgraditi
  - Evitare turni doppi (mattina e pomeriggio)
  - Evitare turni uguali consecutivi
- ***Cosa ne facciamo?***
  - `labeling( [down, minimize(FObj)], Vars ).`
    - **Inizia ad esplorare tutto lo spazio delle soluzioni...Diverge!**

Funzione obiettivo CLP(FD)

- Cerchiamo soluzioni progressivamente migliori:

```
while( not timeout ) {
    labeling( [ff,down], Lista ),
    save( Lista );
    f := valueofObjectiveFunction( Lista );
    addconstraint( FObj #< f );
    deleteValue( Lista )
} load( Lista )
```

Funzione obiettivo CLP(FD)

- Cerchiamo soluzioni progressivamente migliori:

```
while( not timeout ) {
    labeling( [ff,down], Lista ),
    save( Lista );
    f := valueofObjectiveFunction( Lista );
    addconstraint( FObj #< f );
    deleteValue( Lista )
} load( Lista )
```

- **Migliora di poco!**
- **Da gestire con assert + retract + fail !:(**

Funzione obiettivo CLP(FD)

- Cerchiamo soluzioni progressivamente migliori:

```
while( not timeout ) {
    labeling( [ff,down], Lista ),
    save( Lista );
    f := valueofObjectiveFunction( Lista );
    addconstraint( FObj #< f );
    deleteValue( Lista )
} load( Lista )
```

  - **Migliora di poco!**
  - **Da gestire con assert + retract + fail !:(**
- Per la FObj conviene **Local Search**

Local Search

- *Famiglia di* metodi incompleti*:*

- ***Famiglia di* metodi incompleti:**
  - si basano sull'uso di euristiche,

- ***Famiglia di* metodi incompleti:**
  - si basano sull'uso di euristiche,
  - esplorano solo alcune aree dello SDS

Local Search

- ***Famiglia di* metodi incompleti*:***
  - si basano sull'uso di euristiche,
  - esplorano solo alcune aree dello SDS
  - lo scopo è trovare una soluzione ammissibile/buona.

Local Search

- ***Famiglia di* metodi incompleti*:***
  - si basano sull'uso di euristiche,
  - esplorano solo alcune aree dello SDS
  - lo scopo è trovare una soluzione ammissibile/buona.
- ***Concetti fondamentali***
  - Stato

Local Search

- ***Famiglia di* metodi incompleti:**
  - si basano sull'uso di euristiche,
  - esplorano solo alcune aree dello SDS
  - lo scopo è trovare una soluzione ammissibile/buona.
- ***Concetti fondamentali***
  - Stato
  - Funzione obiettivo

Local Search

- ***Famiglia di* metodi incompleti:**
  - si basano sull'uso di euristiche,
  - esplorano solo alcune aree dello SDS
  - lo scopo è trovare una soluzione ammissibile/buona.
- ***Concetti fondamentali***
  - Stato
  - Funzione obiettivo
  - Mossa

Local Search

- ***Famiglia di* metodi incompleti:**
  - si basano sull'uso di euristiche,
  - esplorano solo alcune aree dello SDS
  - lo scopo è trovare una soluzione ammissibile/buona.
- ***Concetti fondamentali***
  - Stato
  - Funzione obiettivo
  - Mossa
  - Vicinato

Local Search

- ***Famiglia di* metodi incompleti:**
  - si basano sull'uso di euristiche,
  - esplorano solo alcune aree dello SDS
  - lo scopo è trovare una soluzione ammissibile/buona.
- ***Concetti fondamentali***
  - Stato
  - Funzione obiettivo
  - Mossa
  - Vicinato
  - Esplorazione del vicinato

Local Search

- ***Famiglia di* metodi incompleti:**
  - si basano sull'uso di euristiche,
  - esplorano solo alcune aree dello SDS
  - lo scopo è trovare una soluzione ammissibile/buona.
- ***Concetti fondamentali***
  - Stato
  - Funzione obiettivo
  - Mossa
  - Vicinato
  - Esplorazione del vicinato
- ***Varie strategie...***

Approcci ibridi

Due tipi di approcci che fondono Constraint Programming e Local Search:

Approcci ibridi

Due tipi di approcci che fondono Constraint Programming e Local Search:

- Algoritmi di CP con all'interno procedure di LS che possono intervenire nell'albero di ricerca:
  - **migliorare la soluzione relativa a foglie o nodi;**
  - **restringere la lista dei figli di un nodo;**
  - **generare in modo greedy un cammino;**

Approcci ibridi

Due tipi di approcci che fondono Constraint Programming e Local Search:

- Algoritmi di CP con all'interno procedure di LS che possono intervenire nell'albero di ricerca:
  - **migliorare la soluzione relativa a foglie o nodi;**
  - **restringere la lista dei figli di un nodo;**
  - **generare in modo greedy un cammino;**
- Algoritmi LS con all'interno procedure di CP che possono
  - **scartare i vicini che violano vincoli del problema;**
  - **esplorare una porzione di vicinato;**
  - **definire la ricerca del miglior vicino come un problema di ottimizzazione con vincoli.**

Approcci ibridi

Due tipi di approcci che fondono Constraint Programming e Local Search:

- Algoritmi di CP con all'interno procedure di LS che possono intervenire nell'albero di ricerca:
  - **migliorare la soluzione relativa a foglie o nodi;**
  - **restringere la lista dei figli di un nodo;**
  - **generare in modo greedy un cammino;**
- Algoritmi LS con all'interno procedure di CP che possono
  - **scartare i vicini che violano vincoli del problema;**
  - **esplorare una porzione di vicinato;**
  - **definire la ricerca del miglior vicino come un problema di ottimizzazione con vincoli.**

La Ricerca Locale

- *Per implementare gli algoritmi di ricerca locale usiamo il framework*

  JEasyLocal [Schaerf, Di Gaspero]

La Ricerca Locale

- ***Per implementare gli algoritmi di ricerca locale usiamo il framework***

  JEasyLocal [Schaerf, Di Gaspero]

- ***Dobbiamo definire:***
  - Input
    - Mese e anno,
    - giorni festivi,
    - ferie richieste,
    - ...

- State
  - **Una soluzione, ovvero un orario:**

Dott	1	2	3	4	5	...	31
MedicoA Mat	UM				UN	...	
MedicoA Pom			RM			...	UN
MedicoB Mat		ED				...	
MedicoB Pom				SU		...	
...	...	...	...	...	...	...	...
MedicoB Mat		SU			ED	...	
MedicoB Pom	ED	UN				...	

La Ricerca Locale

- State
  - **Una soluzione, ovvero un orario:**

Dott	1	2	3	4	5	...	31
MedicoA Mat	UM				UN	...	
MedicoA Pom			RM			...	UN
MedicoB Mat		ED				...	
MedicoB Pom				SU		...	
...	...	...	...	...	...	...	...
MedicoB Mat		SU			ED	...	
MedicoB Pom	ED	UN				...	

- Una FObj associata allo stato
  - **Usiamo la stessa formula usata per CLP(FD)**

- Move
  - Mossa di *scambio*:

    "Scambiare i turni di due medici relativi allo stesso giorno e alla stessa fascia oraria"

Dott	1	2	...
Jones	...	DH	...
...	...	...	...
Freud		UM	...
...	...	...	...

$\Longrightarrow$

Dott	1	2	...
Jones	...	UM	...
...	...	...	...
Freud		DH	...
...	...	...	...

La Ricerca Locale

- Move
  - **Mossa di *scambio*:**

    "Scambiare i turni di due medici relativi allo stesso giorno e alla stessa fascia oraria"

Dott	1	2	...
Jones	...	DH	...
...	...	...	...
Freud		UM	...
...	...	...	...

$\Longrightarrow$

Dott	1	2	...
Jones	...	UM	...
...	...	...	...
Freud		DH	...
...	...	...	...

- Stato iniziale
  - **Quello restituito da CLP(FD)**

La Ricerca Locale

- Move
  - **Mossa di *scambio*:**

    "Scambiare i turni di due medici relativi allo stesso giorno e alla stessa fascia oraria"

Dott	1	2	...
Jones	...	DH	...
...	...	...	...
Freud		UM	...
...	...	...	...

$\Longrightarrow$

Dott	1	2	...
Jones	...	UM	...
...	...	...	...
Freud		DH	...
...	...	...	...

- Stato iniziale
  - **Quello restituito da CLP(FD)**
- Strategia di esplorazione del vicinato
  - **Random - Hill Climbing**
  - **Esaustiva - Steepest Descent**
  - **Esaustiva con TabuSearch**

Eplorazione ibrida

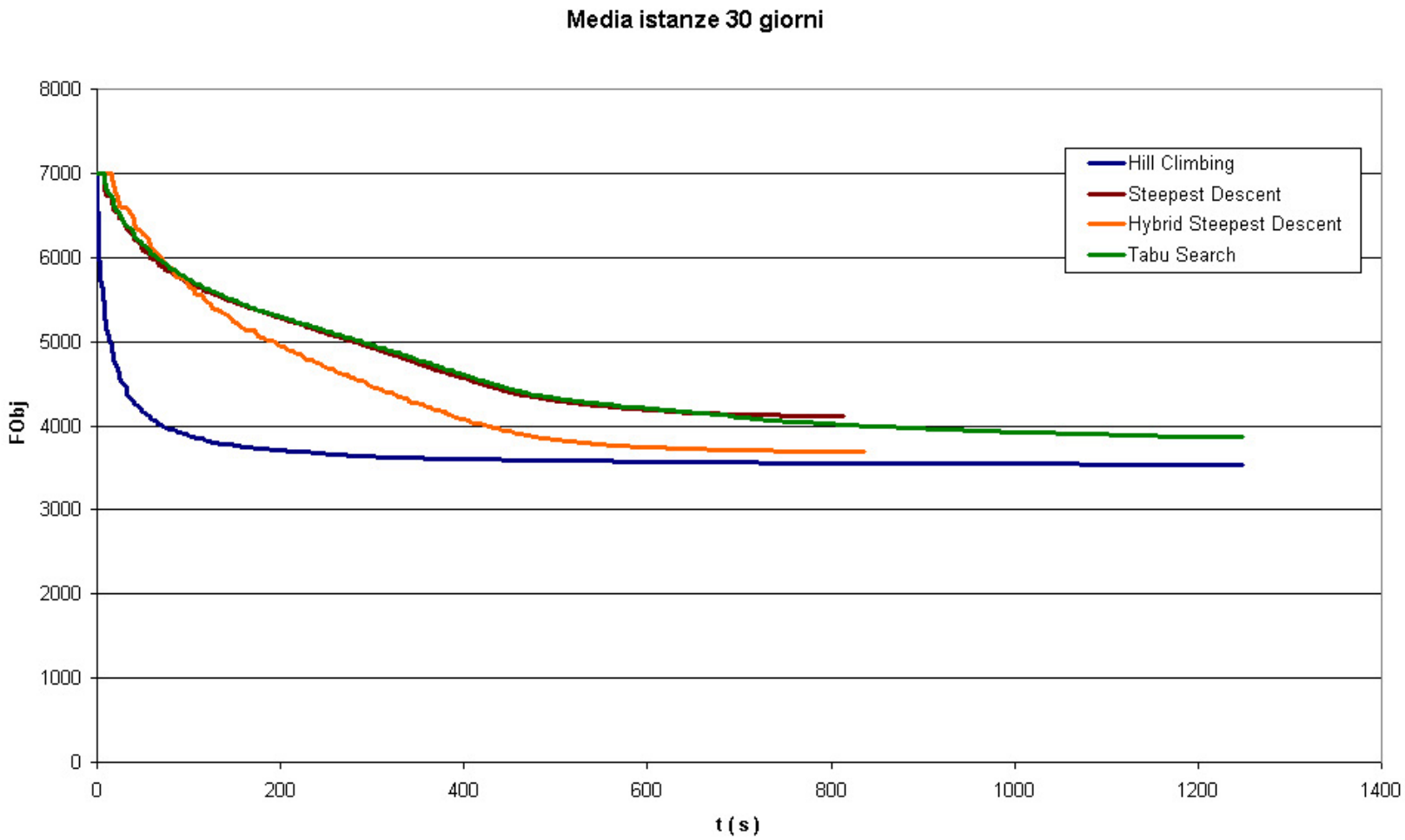
Due tipi di approcci che fondono Constraint Programming e Local Search:

- Algoritmi di CP con all'interno procedure di LS che possono intervenire nell'albero di ricerca:
  - **migliorare la soluzione relativa a foglie o nodi;**
  - **restringere la lista dei figli di un nodo;**
  - **generare in modo greedy un cammino;**
- Algoritmi LS con all'interno procedure di CP che possono
  - **scartare i vicini che violano vincoli del problema;**
  - **esplorare una porzione di vicinato;**
  - **definire la ricerca del miglior vicino come un problema di ottimizzazione con vincoli.**

- *Lanciamo Ricerca Locale*

Lanciamo Ricerca Locale

- La FObj viene notevolmente migliorata
- Hill climbing più veloce e più efficace



Alcuni accorgimenti pratici

- *JEasyLocal in Java, CLP(FD) in SICStus Prolog!*
- *Come fare?*

Alcuni accorgimenti pratici

- ***JEasyLocal in Java, CLP(FD) in SICStus Prolog!***
- ***Come fare?***
  - Master - slave?
  - Master Java o SICStus?

Alcuni accorgimenti pratici

- ***JEasyLocal in Java, CLP(FD) in SICStus Prolog!***
- ***Come fare?***
  - Master - slave?
  - Master Java o SICStus?
- ***Come far comunicare i linguaggi per lo scambio di informazioni?***

Alcuni accorgimenti pratici

- ***JEasyLocal in Java, CLP(FD) in SICStus Prolog!***
- ***Come fare?***
  - Master - slave?
  - Master Java o SICStus?
- ***Come far comunicare i linguaggi per lo scambio di informazioni?***
  - Jasper?...

Alcuni accorgimenti pratici

- ***JEasyLocal in Java, CLP(FD) in SICStus Prolog!***
- ***Come fare?***
  - Master - slave?
  - Master Java o SICStus?
- ***Come far comunicare i linguaggi per lo scambio di informazioni?***
  - Jasper?...
  - File di testo?

Alcuni accorgimenti pratici

- ***JEasyLocal in Java, CLP(FD) in SICStus Prolog!***
- ***Come fare?***
  - Master - slave?
  - Master Java o SICStus?
- ***Come far comunicare i linguaggi per lo scambio di informazioni?***
  - Jasper?...
  - File di testo?
  - File XML?

Alcuni accorgimenti pratici

- *L'orario generato come lo restituiamo?*

Alcuni accorgimenti pratici

- ***L'orario generato come lo restituiamo?***
  - Deve poter essere visualizzabile - stampabile...

Alcuni accorgimenti pratici

- ***L'orario generato come lo restituiamo?***
  - Deve poter essere visualizzabile - stampabile...
  - In ospedale usano un foglio excel che compilano a mano...

Alcuni accorgimenti pratici

L'orario generato come lo restituiamo?

- Deve poter essere visualizzabile - stampabile...
- In ospedale usano un foglio excel che compilano a mano...
- Salviamo l'orario in un formato ricosciuto da excel per poterlo caricare comodamente!

Alcuni accorgimenti pratici

- ***L'orario generato come lo restituiamo?***
  - Deve poter essere visualizzabile - stampabile...
  - In ospedale usano un foglio excel che compilano a mano...
  - Salviamo l'orario in un formato ricosciuto da excel per poterlo caricare comodamente!
- ***Dobbiamo dare il tool in mano a qualcuno che non sa niente di ricerca locale o programmazione logica... GUI!***

Sviluppi futuri

Non si finisce mai! :)

Sviluppi futuri

Non si finisce mai! :)

- *Migliorare la fusione di CLP(FD)-LS*

Sviluppi futuri

Non si finisce mai! :)

- ***Migliorare la fusione di CLP(FD)-LS***
  - Uso di GECODE per CLP(FD), scritto in C++

Sviluppi futuri

Non si finisce mai! :)

- ***Migliorare la fusione di CLP(FD)-LS***
  - Uso di GECODE per CLP(FD), scritto in C++
  - Uso di EasyLocal++, versione C++ di JEasyLocal

Sviluppi futuri

Non si finisce mai! :)

- ***Migliorare la fusione di CLP(FD)-LS***
  - Uso di GECODE per CLP(FD), scritto in C++
  - Uso di EasyLocal++, versione C++ di JEasyLocal
  - Fusione dei due framework in un unico ambiente di sviluppo

Sviluppi futuri

Non si finisce mai! :)

- ***Migliorare la fusione di CLP(FD)-LS***
  - Uso di GECODE per CLP(FD), scritto in C++
  - Uso di EasyLocal++, versione C++ di JEasyLocal
  - Fusione dei due framework in un unico ambiente di sviluppo
- ***Vantaggi possibili***

Sviluppi futuri

Non si finisce mai! :)

- ***Migliorare la fusione di CLP(FD)-LS***
  - Uso di GECODE per CLP(FD), scritto in C++
  - Uso di EasyLocal++, versione C++ di JEasyLocal
  - Fusione dei due framework in un unico ambiente di sviluppo
- ***Vantaggi possibili***
  - Modellare problemi ad alto livello: GECODE è pacchetto CLP(FD) completo;

Sviluppi futuri

Non si finisce mai! :)

- ***Migliorare la fusione di CLP(FD)-LS***
  - Uso di GECODE per CLP(FD), scritto in C++
  - Uso di EasyLocal++, versione C++ di JEasyLocal
  - Fusione dei due framework in un unico ambiente di sviluppo
- ***Vantaggi possibili***
  - Modellare problemi ad alto livello: GECODE è pacchetto CLP(FD) completo;
  - Algoritmi di ricerca locale già pronti

Sviluppi futuri

Non si finisce mai! :)

- ***Migliorare la fusione di CLP(FD)-LS***
  - Uso di GECODE per CLP(FD), scritto in C++
  - Uso di EasyLocal++, versione C++ di JEasyLocal
  - Fusione dei due framework in un unico ambiente di sviluppo
- ***Vantaggi possibili***
  - Modellare problemi ad alto livello: GECODE è pacchetto CLP(FD) completo;
  - Algoritmi di ricerca locale già pronti
  - Manipolare strutture dati e flusso di controllo: C++

Sviluppi futuri

Non si finisce mai! :)

- ***Migliorare la fusione di CLP(FD)-LS***
  - Uso di GECODE per CLP(FD), scritto in C++
  - Uso di EasyLocal++, versione C++ di JEasyLocal
  - Fusione dei due framework in un unico ambiente di sviluppo
- ***Vantaggi possibili***
  - Modellare problemi ad alto livello: GECODE è pacchetto CLP(FD) completo;
  - Algoritmi di ricerca locale già pronti
  - Manipolare strutture dati e flusso di controllo: C++
  - Ditribuire le elaborazioni su più processori

Sviluppi futuri

- *Difficoltà*

Sviluppi futuri

- ***Difficoltà***
  - Entrare nei dettagli implementativi di software scritto da altri

Sviluppi futuri

- ***Difficoltà***
  - Entrare nei dettagli implementativi di software scritto da altri
  - Trovare strutture dati agili per far comunicare i due framework con facilità

Sviluppi futuri

- ***Difficoltà***
  - Entrare nei dettagli implementativi di software scritto da altri
  - Trovare strutture dati agili per far comunicare i due framework con facilità
  - Trade-off: efficienza computazionale VS facilità di modellazione

Parentesi: il GECODE

- *GEneric COnstraint Development Environment*

Parentesi: il GECODE

- *GEneric COnstraint Development Environment*
- *Ambiente free in C++ per la programmazione con vincoli*

Parentesi: il GECODE

- ***GEneric COnstraint Development Environment***
- ***Ambiente free in C++ per la programmazione con vincoli***
  - Stabile

Parentesi: il GECODE

- ***GEneric COnstraint Development Environment***
- ***Ambiente free in C++ per la programmazione con vincoli***
  - Stabile
  - Completo

Parentesi: il GECODE

- ***GEneric COnstraint Development Environment***
- ***Ambiente free in C++ per la programmazione con vincoli***
  - Stabile
  - Completo
  - Efficiente (più di SICStus)

Parentesi: il GECODE

- ***GEneric COnstraint Development Environment***
- ***Ambiente free in C++ per la programmazione con vincoli***
  - Stabile
  - Completo
  - Efficiente (più di SICStus)
  - Codice sorgente a disposizione

Parentesi: il GECODE

- ***GEneric COnstraint Development Environment***
- ***Ambiente free in C++ per la programmazione con vincoli***
  - Stabile
  - Completo
  - Efficiente (più di SICStus)
  - Codice sorgente a disposizione
  - No vincoli di allocazione della memoria

Parentesi: il GECODE

- ***GEneric COnstraint Development Environment***
- ***Ambiente free in C++ per la programmazione con vincoli***
  - Stabile
  - Completo
  - Efficiente (più di SICStus)
  - Codice sorgente a disposizione
  - No vincoli di allocazione della memoria
  - Sta diventando uno standard...

Parentesi: il GECODE

- ***GEneric COnstraint Development Environment***
- ***Ambiente free in C++ per la programmazione con vincoli***
  - Stabile
  - Completo
  - Efficiente (più di SICStus)
  - Codice sorgente a disposizione
  - No vincoli di allocazione della memoria
  - Sta diventando uno standard...
  - Documentato ...male!!!

Parentesi: il GECODE

- ***GEneric COnstraint Development Environment***
- ***Ambiente free in C++ per la programmazione con vincoli***
  - Stabile
  - Completo
  - Efficiente (più di SICStus)
  - Codice sorgente a disposizione
  - No vincoli di allocazione della memoria
  - Sta diventando uno standard...
  - Documentato ...male!!!
- ***www.gecode.org***

Domande?

?

Bibliografia

- E. K. Burke, P. De Causmaecker , G. Vanden Berghe, H. Van Landeghem *The State of the Art of Nurse Rostering*, Journal of Scheduling, **7**, pp. 441-499, 2004.
- L. Di Gaspero, A. Schaerf, *EasyLocal++: an object-oriented framework for the flexible design of local-search algorithms*, Software-Practice and Experience, 33:733-765, 2003.
- A. Dovier, *Linguaggi e Tecniche Speciali di Programmazione*, http://www.dovier.dimi.it/DID/lnc.pdf, 2004.
- F. Focacci, F. Laburthe, A. Lodi, *Local search and constraint programming*, Handbook of Metaheuristic, 2002.
- N. Jussien, O. Lhomme, *Local search with constraint propagation and conflict-based heuristic*, Artificial Intelligence, **139**(1), pp. 21-45, 2002.
- E. Monfroy, F. Saubion, T. Lambert *On hybridization of local search and constraint propagation*, In proceedings of ICLP'2004, volume 3132 of LNCS, pp. 299-313, Springer Verlag, 2004.

Bibliografia

- R. Cipriano, A. Dovier, L. Di Gaspero *Un tool integrato per il rostering ospedaliero*, Università degli Studi di Udine, Tesi di Laurea Specialistica in Informatica, A.A. 2004-2005.
- R. Cipriano , L. Di Gaspero , A. Dovier *Hybrid Approaches for Rostering: A Case Study in the Integration of Constraint Programming and Local Search*, Hybrid Metaheuristics, Third International Workshop, HM 2006, Gran Canaria, Spain, October 13-15, 2006, Proceedings.

raffaele.cipriano@dimi.uniud.com

Bibliografia

- R. Cipriano, A. Dovier, L. Di Gaspero *Un tool integrato per il rostering ospedaliero*, Università degli Studi di Udine, Tesi di Laurea Specialistica in Informatica, A.A. 2004-2005.
- R. Cipriano , L. Di Gaspero , A. Dovier *Hybrid Approaches for Rostering: A Case Study in the Integration of Constraint Programming and Local Search*, Hybrid Metaheuristics, Third International Workshop, HM 2006, Gran Canaria, Spain, October 13-15, 2006, Proceedings.

raffaele.cipriano@dimi.uniud.com

Grazie per l'attenzione