Tecniche di Ricerca Locale e loro Applicazioni

Dr. Ivan Serina

serina@ing.unibs.it

Università degli studi di Brescia

Outline

- Introduzione
- Ricerca locale in SAT
  - Gsat
  - Tabu Search
  - Wsat
- Applicazioni
  - Assegnamento di Frequenze
  - High School Timetabling
  - Vehicle routing
  - Portfolio …..
- Conclusioni

Tecniche di Ricerca

- *Metodi Costruttivi*
  - Basati su Backtracking (esaustivi)
    - Forward Checking, Backjumping…
    - Branch & Bound..
  - Backtracking-free
    - Greedy..
- *Methodi Selettivi*
  - Ricerca Locale
    - Hill-Climbing - Simulated Annealing - Tabu Search
  - Evolutionary Methods
    - Algoritmi genetici …
- *Programmazione Intera*
  - Relaxation + Rounding…

Caratteristiche

- *Metodi Costruttivi*
  - Migliore comprensione del processo di ricerca
  - Veloci per problemi semplici
  - Migliore controllo dei passi critici

- *Metodi Selettivi*
  - Si sono dimostrati efficaci in applicazioni reali
  - Forniscono soluzioni Approssimate
  - Possono modificare soluzioni precedenti

Ricerca Locale: Caratteristiche

- *Comune alle varie tecniche tecnica*
  - Spazio di ricerca
  - Vicinato
  - Funzione Costo
  - Selezione stato iniziale

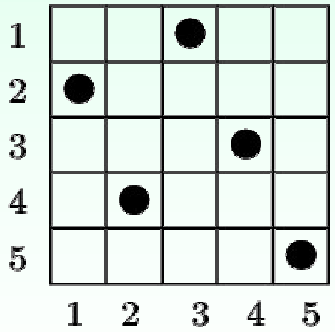
- *Funzione della tecnica utilizzata*
  - Scelta della mossa
  - Criterio di Stop (e Restart)

Spazio di Ricerca

- Deve contenere soluzioni ammissibili (ed anche le ottime)
- Il più piccolo possibile
- Consentire di esaminare facilmente i sui elementi

Esempio: problema delle *n* regine

  - Matrici [1..n]×[1..n]
  - Assegnamenti [1..n] →[1..n]
  - Permutazioni di [1..n]



Relazione Vicinato

- *Connette tutto lo spazio di ricerca*
- *Non troppo grande .....e*
- *Non troppo piccolo*
- *Facile da calcolare (e memorizzare)*
- *Facile da valutare le differenze di costo*
  - Efficace

Esempio: problema *n*-regine

- Vicinato di “Swap” (per le permutazioni)

  [4,**3**,6,5,2,**1**,8,7] →[4,**1**,6,5,2,**3**,8,7]
- Vicinato di Sostituzione (per gli assegnamenti)

  [3,4,1,7,8,**4**,2,1] →[3,4,1,7,8,**6**,2,1]

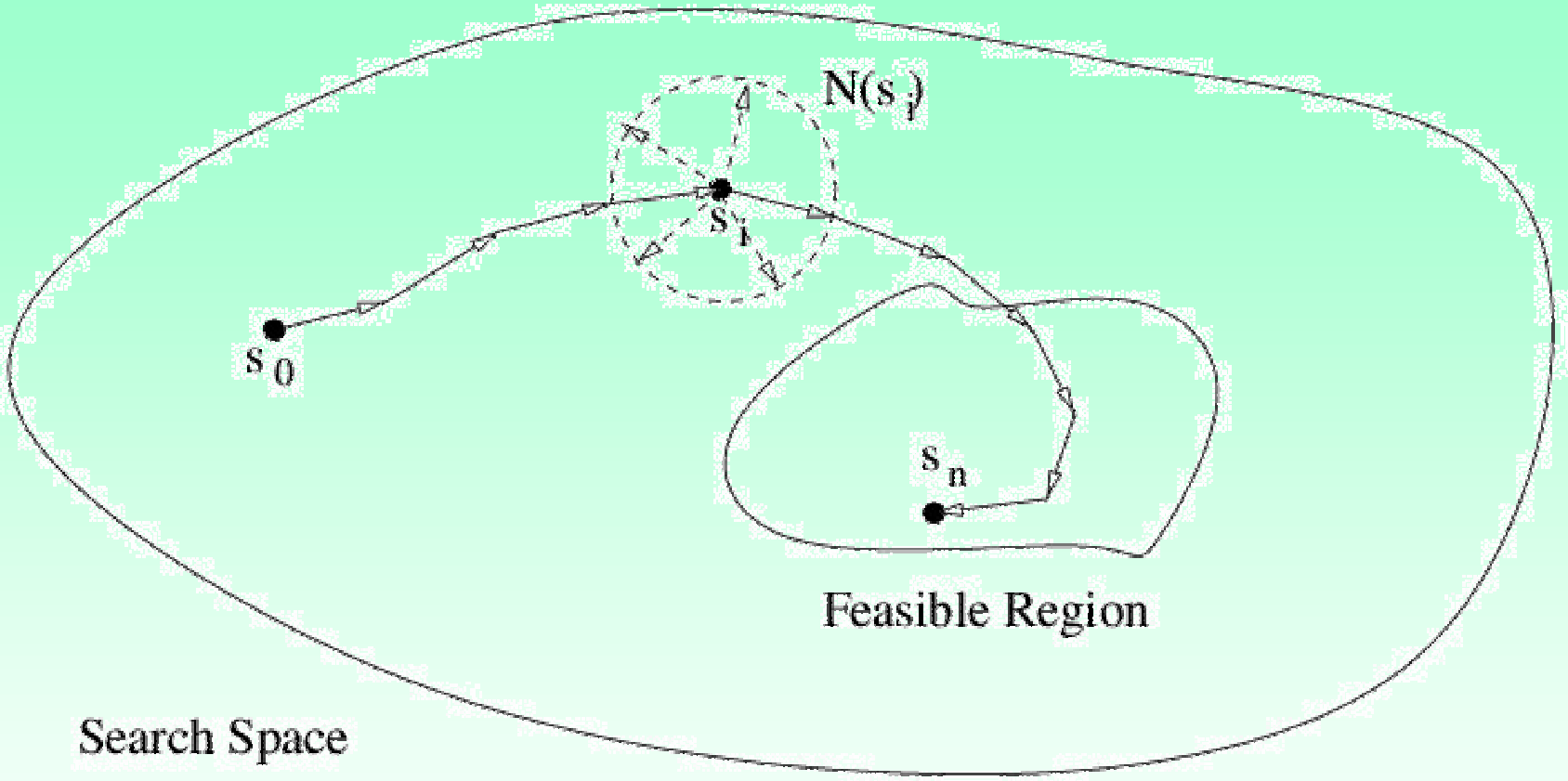
Funzione Costo

- *Esprime distanza da soluzione ammissibile*
- *Dipende da funzione obiettivo*
- Prende in considerazione miglioramenti invisibili
- Consente la diversificazione

Scelta Soluzione Iniziale

- *Random*
- *Ottenuta mediante tecniche greedy*
- *Soluzione di una istanza precedente*

Tecniche Ricerca Locale



f : Cost Function

Tecniche di base

- *Hill Climbing*
- *Tabu Search*
- *Ricerca Walk*
- *Simulated Annealing …….*

Hill Climbing

"applica solo mosse che migliorano (o al più non peggiorano)"

- Discesa casuale, il primo che migliora, il migliore
  - MCHC (Minton et al '90)
  - GSAT (Selman et al '92)
- Parametro Principale
  - IDLE_ITERATIONS

GSAT

Risolve la soddisfacibilità di formule proposizionali CNF

- Letterali: $\{p_1, p_2, \ldots, p_n\}$
- Assegnamenti: $A(p_i)= \{true, false\}$
- Clausole: $p_i \vee p_j \vee \neg p_k$

Algoritmo

- Scegli il letterale *p* che soddisfa il maggior numero di clausole (a parità viene scelto casualmente)
- Cambia (flip) il valore di verità di *p*

Cambia: $\{a_1, a_2, \ldots, a_i, \ldots\} \rightarrow \{a_1, a_2, \ldots, \neg a_i, \ldots\}$

Funzione Costo: "*Massimizza le clausole soddisfatte*"

GSAT

Procedure GSAT

- **Input**:a set of clauses α, MAX-FLIPS, and MAX-TRIES
- **Output**: a satisfying truth assignment of α, if found

```
for i := 1 to MAX-TRIES
    T := a randomly generated truth assignment;
    for j := 1 to MAX-FLIPS
        If T satisfies α then return T
        p := a propositional variable such that a change in its truth
          assignment gives the largest increase in the total number of
          clauses of α that are satisfied by T;
        T := T with the truth assignment of p reversed;
    end for
end for
Return "no satisfying assignment found“;
```

Tabu Search

"Fai la mossa migliore (anche se peggiora)"

- Non considera gli elementi di una speciale lista detta *tabu list*;
- Ad ogni passo inserisce la mossa effettuata in testa alla *tabu list* e toglie l'ultima della lista;
- Si possono considerare delle "*eccezioni*" alla tabu list: anche se una mossa e' nella tabu list ma da un grande miglioramento questa può essere applicata;

Tabu search

Procedure Tabu

- **Input**:a set of clauses α, MAX-FLIPS
- **Output**: a satisfying truth assignment of α, if found

T := a randomly generated truth assignment;
Initialize *tabu list*;
for *j* := 1 **to** MAX-FLIPS
    If T satisfies α **then return** T
    p := a propositional variable such that a change in its truth assignment gives the largest increase in the total number of clauses of α that are satisfied by T among the non-tabu moves;
    T := T with the truth assignment of p reversed;
    Insert p in the *tabu list;*
    Remove the oldest move from the tabu list;
Return "no satisfying assignment found";

Ricerca Walk

Procedure WSAT

- **Input**:a set of clauses α, MAX-FLIPS, MAX-TRIES, *noise*
- **Output**: a satisfying truth assignment of α, if found

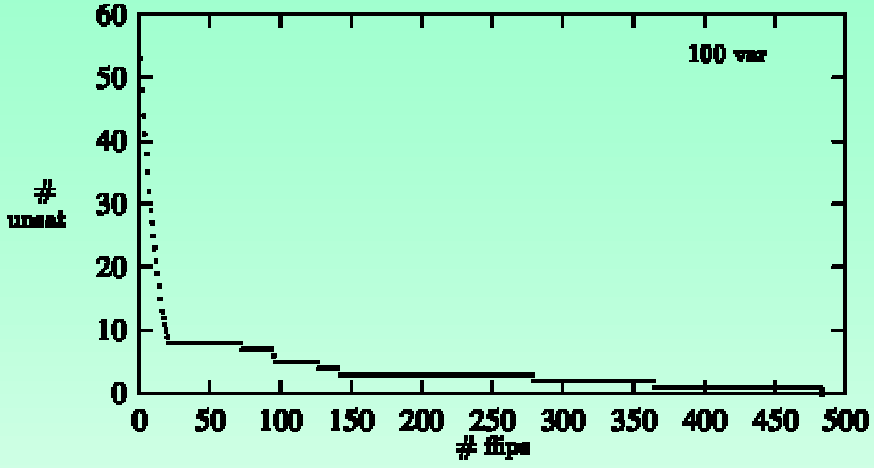
for *i* := 1 **to** MAX-TRIES
    T := a randomly generated truth assignment;
    for *j* := 1 **to** MAX-FLIPS
        If T satisfies α **then return** T
        With probability *noise*
            p:= a random variable occuring in some unsatisfied clausole;
        With probability *1-noise*
            p := a propositional variable such that a change in its truth assignment gives the largest increase in the total number of clauses of α that are satisfied by T;
        T := T with the truth assignment of p reversed;
Return "no satisfying assignment found“;

Risultati SAT

formulas		GSAT			DP		
vars	clauses	M-FLIPS	tries	time	choices	depth	time
50	215	250	6.4	0.4s	77	11	1.4s
70	301	350	11.4	0.9s	42	15	15s
100	430	500	42.5	6s	84×10^3	19	2.8m
120	516	600	81.6	14s	0.5×10^6	22	18m
140	602	700	52.6	14s	2.2×10^6	27	4.7h
150	645	1500	100.5	45s	—	—	—
200	860	2000	248.5	2.8m	—	—	—
250	1062	2500	268.6	4.1m	—	—	—
300	1275	6000	231.8	12m	—	—	—
400	1700	8000	440.9	34m	—	—	—
500	2150	10000	995.8	1.6h	—	—	—

Table 1: Results for GSAT and DP on *hard* random 3CNF formulas.

Risultati SAT



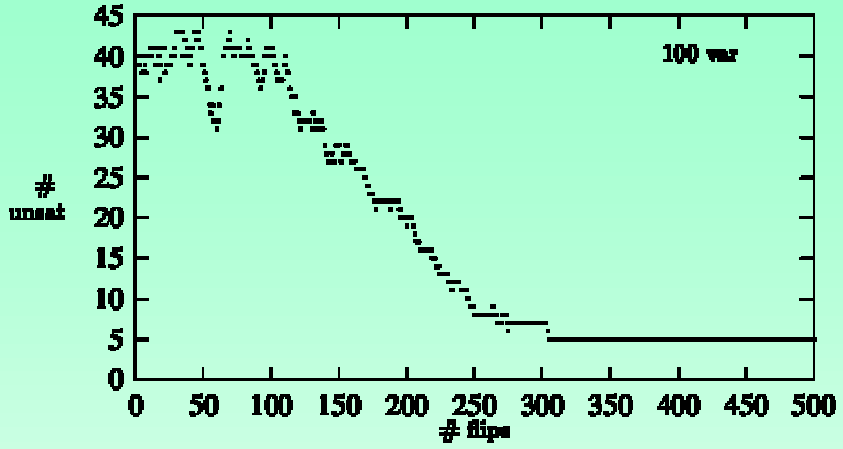


Figure 3: Simulated annealing's search space on a 100 variable formula.

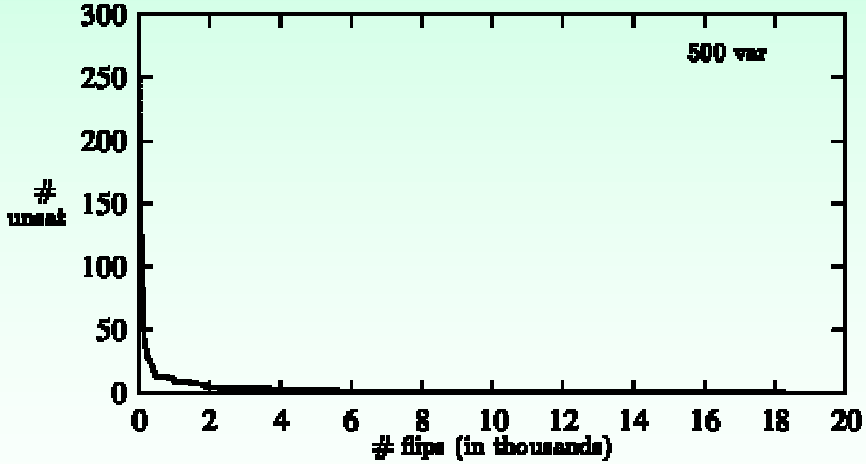


Figure 2: GSAT's search space on a 100 and 500 variables formulas.

formula		GSAT									Simul. Ann.		
		basic			walk			noise					
vars	clauses	time	flips	R	time	flips	R	time	flips	R	time	flips	R
100	430	.4	7554	8.3	.2	2385	1.0	.6	9975	4.0	.6	4749	1.1
200	860	22	284693	143	4	27654	1.0	47	396534	6.7	21	106643	1.2
400	1700	122	2.6×10^6	67	7	59744	1.1	95	892049	6.3	75	552433	1.1
600	2550	1471	30×10^6	500	35	241651	1.0	929	7.8×10^6	20	427	2.7×10^6	3.3
800	3400	*	*	*	286	1.8×10^6	1.1	*	*	*	*	*	*
1000	4250	*	*	*	1095	5.2×10^6	1.2	*	*	*	*	*	*
2000	8490	*	*	*	3255	23×10^6	1.1	*	*	*	*	*	*

Table 1: Comparing noise strategies on hard random 3CNF instances.

formula			GSAT									Simul. Ann.		
			basic			walk			noise					
id	vars	clauses	time	flips	R	time	flips	R	time	flips	R	time	flips	R
med.	273	2311	7.5	70652	125	0.4	3464	1.0	4.5	41325	1.1	4.5	12147	1.0
rev.	201	1332	3.7	41693	100	0.3	3026	1.0	2.7	29007	1.1	2.7	9759	1.0
hanoi	417	2559	*	*	*	39	334096	2.6	20017	16×10^6	100	3250	4.1×10^6	25
huge	937	14519	*	*	*	39	143956	1.1	9649	37×10^6	200	9302	4.4×10^6	13

Table 2: Comparing noise strategies on SAT encodings of planning problems.

formula		Int.P.	GSAT									Simul. Ann.		
			basic			walk			noise					
id	vars	time	time	flips	$R.$	time	flips	$R.$	time	flips	$R.$	time	flips	$R.$
f16a1	1650	2039	58	709895	5	2	3371	1.1	375	1025454	6.7	12	98105	1.3
f16b1	1728	78	269	2870019	167	12	25529	1.0	1335	2872226	167	11	96612	1.4
f16c1	1580	758	2	12178	1.0	1	1545	1.0	5	14614	1.0	5	21222	1.0
f16d1	1230	1547	87	872219	7.1	3	5582	1.0	185	387491	1.0	4	25027	1.0
f16e1	1245	2156	1	2090	1.0	1	1469	1.0	1	3130	1.0	3	5867	1.0

Table 3: Comparing noise strategies on the circuit synthesis problem instances as studied in Kamath *et al.* (1991).

Applicazioni

Le tecniche ricerca locale hanno trovato valida applicazione in diversi settori applicativi:

- Problemi di ottimizzazione combinatoria, solitamente NP-completi
- Legati alla finanza
- Problemi Scheduling
- Problemi di planning
  - LPG

Applicazioni

- Assegnamento frequenze
- High School Timetabling
- Vehicle routing
- Portfolio …..

Assegnamento delle Frequenze

- Date:
  - n celle $c_1, c_2, \ldots, c_n$
  - m frequenze o canali $(f_1, \ldots f_m)$
  - vettore delle richieste R_n
  - matrice di separazione $S_{n \times n}$
  - matrice dei preassegnamenti $P_{n \times m}$ {-1,0,1}
- Trovare per ogni cella c_i

  un insieme di frequenze $F_i = \{f_{i1}, \ldots, f_{ir_i}\}$

  tale che

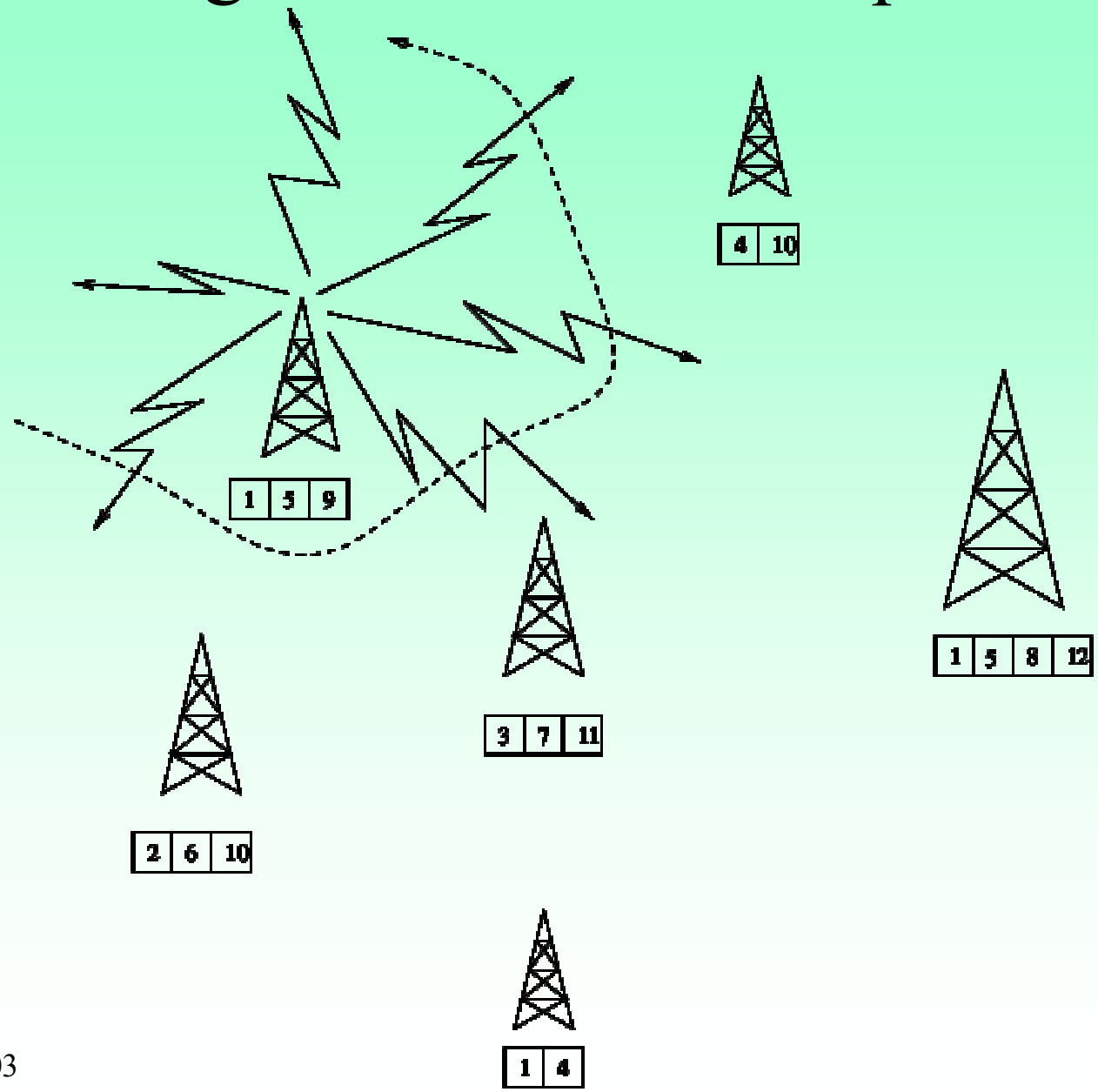
  - $\forall\, i, j, h, k : 1 \le i \le j \le n,$
    $1 \le h \le r_i,\ 1 \le k \le r_j$

    $$|f_{ih} - f_{jk}| \le S_{ij}$$
  - $\forall\, i, j :\ 1 \le i \le n, 1 \le j \le m$

    $$\text{if} \quad P_{ij} = -1 \ \text{ then } \ j \notin F_i$$

    $$\text{if} \quad P_{ij} = 1 \quad \text{then } \ j \in F_i$$

Assegnamento delle Frequenze



Assegnamento delle Frequenze

- *Spazio di ricerca*

  Matrice degli assegnamenti $X_{n \times n}$:

$$x_{i \times j}: \quad \begin{cases} 0 & \text{if } j \notin F_i \\ 1 & \text{if } j \in F_i \end{cases}$$

- Tale che
  - le “richieste” sono soddisfatte: $\Sigma_{j=1}^{m} x_{ij} = r_i$
  - i preassegnamenti sono soddisfatti:

    if P_{ij}=-1 then x_{ij}=0

    if P_{ij}=1 then x_{ij}=1

Assegnamento delle Frequenze

- *Vicinato:*
  - Defined by a triple: $\langle i, j, k \rangle$
  - $j \in F_i$ and $k \notin F_i$
  - $P_{ij} \neq 1$ and $P_{ik} \neq 0$

Example:

	j			k		
	1	0	0	0	1	0
	0	0	0	1	1	0	0
i	0	1	1	0	0	0	1
	1	1	0	1	0	0	0

Assegnamento delle Frequenze

- *Funzione Costo*
  - Somma delle violazioni Intra-celle
  - Somma delle violazioni Inter-cella

- Esempio

X	S
1 0 0 0 0 1 0	2 1 0 1
0 0 0 1 1 0 0	1 2 0 1
0 1 1 0 0 0 1	0 0 3 1
1 1 0 1 0 0 0	1 1 1 2

Assegnamento delle Frequenze

- Simulated Annealing [Duque-Antòn 93]
  - Mixed Cooling Scheme
  - Extended Neighborhood (chains of moves)
- Tabu Search [Castelino et al. 96]
  - Frequency-based Tabu
- Comparison [Costa 93]: Tabu Search Wins!

High School Timetabling

Trovare una schedulazione settimanale delle lezioni di una scuola superiore, dato

- La matrice “richieste” docenti
  - Docente *Mazzone* insegna 4 ore nella 3A
- Indisponibilità Insegnanti e Classi
  - Docente *Trapattoni* non e’ disponibile il lunedì
- Altri vincoli
  - Classi 1A e 1B fanno ginnastica assieme

NP Completo (Even et al. ‘76)

High School Timetabling (Esempio)

Requirements	A1	A2
smith	3	2
jones	0	5
brown	6	1

Availability	Mo	Tu	We	Th	Fr
smith	— —	— —	— —	X X	— —
jones	— —	X X	— —	— —	— —
brown	— —	X —	X —	— —	— —
A1	— —	— —	— —	— —	— X
A2	X —	— —	— —	— —	— —

Timetable	Mo	Tu	We	Th	Fr
smith	A1 A2	A1 A2	A1 —	— —	— —
jones	— —	— —	A2 A2	A2 A2	A2 —
brown	— A1	— A1	— A1	A1 A1	A1 —

High School Timetabling

$$
\begin{aligned}
& find && x_{ijk} && (i = 1..m; j = 1..n; k = 1..p) \\
s.t. \quad & \sum_{k=1}^{p} x_{ijk} = r_{ij} && && (i = 1..m; j = 1..n) \\
& \sum_{i=1}^{m} x_{ijk} \leq t_{jk} && && (j = 1..n; k = 1..p) \\
& \sum_{j=1}^{n} x_{ijk} \leq c_{ik} && && (i = 1..m; k = 1..p) \\
& x_{ijk} = 0 \text{ or } 1 && && (i = 1..m; j = 1..n; k = 1..p)
\end{aligned}
$$

- $R_{m \times n}$: matrice delle richieste
- $T_{n \times p}$, $C_{n \times p}$: matrici delle disponibilità
- $X_{m \times n \times p}$: timetable $\rightarrow x_{ijk}=1$ se insegnate *j* insegna nella classe *i* al periodo *k*

High School Timetabling

Altri "*Hard Constraints*"

- "Copertura" delle classi
- Lezioni simultanee (es ginnastica)
- Classi bilingue
- Altri impegni dei docenti (es. incontri con i genitori, ore a disposizione, …)

High School Timetabling

Funzione Obiettivo (*Soft Contraints*):

Alcune componenti

Comp(weight)	Teacher's Daily Assignment				
Holes(1)	1B	—	2B	2B	2A
Too few(3)	—	—	2B	—	—
Splits(6)	2B	2A	1B	2B	—
Clusters(6)	—	2B	2B	2B	2B

High School Timetabling (rappresentazione del problema)

teacher	Monday						Tuesday						..
parenti			3A	3A	<>	2A					<>	2A	..
scocchera	1A	1A	4A	5A	5A	XX	1A	4A	4A	<>		XX	..
mariani	2B	2B	2B	<>	1B		2B	3B					..
marcattili	1B	1B	4B	4B			<>	1B	**	4B	5B		..
giacchetti	1C	<>	3C	2C					<>	3C	3C	2C	..
mori	5C	4C	4C	<>	1C					5C	1C	1C	..
saconis	1D	<>	<>	2D	2D						2D	2D	..
gabriele	XX	XX	XX	XX	XX	XX	2E	2E	3D	3D	5D		..
giagnoro	3A	2A						5A	3A	2A	1A	4A	..
greco	XX	XX	XX	XX	XX	XX		2C	3C	2B	1B	3B	..
ranieri		2d	2D	1D	3D	2e				1D	3D		..
spadoni	XX	XX	XX	XX	XX	XX					3A	4a	..
di_pofi				2B	4B					1B	3B	5B	..
meloni	<>	<>	5C	1C	4C		XX	XX	XX	XX	XX	XX	..
valle	XX	2D	2d	3D	3D	2E	XX	XX	XX	XX	XX	XX	..

High School Timetabling

Algoritmo

- Alternare:
  - Tabu search (mosse atomiche)
  - Hill Climbing(doppie mosse)

High School Timetabling

Caratteristiche tecniche di ricerca

- Hill Climbing (veloce)
  - Doppie mosse casuali
    (le mosse “lunghe” possono trovare dei miglioramenti che mosse semplici potrebbero non considerare)
  - Accetta mosse che non migliorano
    (utile per spostarsi nei plateaux)
- Tabu Search (lenta)
  - *Adaptive relaxation*
    (il peso degli hard constraints varia in funzione del numero di soluzioni ammissibili incontrate)
  - Tabu list di lunghezza variabile e doppio stato tabu
    (lunghezza della tabu list cambiata dinamicamente in modo casuale)
    (mosse tabu e posizioni tabu)

High School Timetabling

Vantaggi dell'alternanza

- Hill Climbing trova la prima soluzione
- Hill Climbing "cambia" la posizione nello spazio di ricerca quando la tabu search fallisce
- La ricerca tabu "lavora" solo quando necessario

High School Timetabling

Analisi dei risultati sperimentali

- Le soluzioni automatiche si sono dimostrate superiori rispetto a quelle manuali
- Modifiche manuali sono a volte necessarie
- L'interazione sistema/utente e assolutamente necessaria
  - La ricerca Tabu e particolarmente utile per modifiche interattive (ad esempio impostando alcuni cambiamenti come necessari)

Vehicle routing

"Determinare percorso o collezione di percorsi ottimo in un insieme di città considerando opportuni vincoli"

- Definito come grafo diretto G=(V,A)
  - $V=(v_1,\ldots v_n)$ insieme di vertici (città)
  - $A=\{(v_i,v_j): v_i,v_j \in V, i \neq j\}$ insieme di connessioni
  - Vertice v_0 rappresenta la locazione in cui sono dislocati *m* veicoli identici
  - $D=(v_{ij})$ matrice non negativa delle distanze o dei costi o del tempo di percorrenza su A

Conclusioni

- Effettiva applicabilità delle tecniche di ricerca locale in problemi di interesse applicativo
- Numerosi algoritmi disponibili
- Non esiste una tecnica “migliore” delle altre, dipende dalla particolare applicazione
- Fondamentale importanza della definizione dello spazio di ricerca, vicinato e possibili miglioramenti delle singole tecniche