

OTTIMIZZAZIONE COMBINATORIA

Implementazione Algoritmo per risoluzione del
problema dei tagli 2D a ghigliottina

Relazione e risultati

Studente
Elvis Ciotti
II anno spec.
Matr. 261459

Prof. Aristide Mingozi

Descrizione e istruzioni utilizzo del programma

Input

Lanciare a linea di comando con i parametri:

```
2d.exe --l <L> --h <H> --m <M> --p <p1> <p2> ... <pM> --pa --output <FILE.HTML>
```

Dove:

1. <L> = larghezza piano
2. <H> = altezza piano
3. <M> = numero pezzi
4. <p1>, <p2>, <pM> = elenco dei pezzi nel formato
 <pM> = <L> <H> <V> (L,H,V sono larghezza, altezza e profitto del pezzo)
5. --pa = setta il profitto dei pezzi proporzionale all'area (se è presente il parametro 3, i profitti vengono soprascritto con l'area).
6. <FILE.HTML> = nome file html di output, se non specificato viene generato nel formato
 out_<L>x<H>_<M>pezzi_<timestamp>.html

Tutti gli argomenti sono facoltativi. I primi 3 sono necessari, quelli mancanti tra questi verranno richiesti in input a runtime.

Esempi input:

piano di taglio 3x2 con un pezzo 2x2 (profitto 4) e un pezzo 1x1 (profitto 3)

```
ot.exe --l 3 --h 2 --m 2 --p 2 2 4 1 1 3
```

piano di taglio 3000x2000 con 20 pezzi casuali

```
ot.exe --l 3000 --h 2000 --m 20
```

Generazione pezzi casuali

Le dimensioni dei pezzi generati sono:

Se $M < 5$: pezzi con dimensione $l = [1/3 L, 3/2 L]$, $h = [1/3 H, 3/2 H]$

Se $5 < M < 10$: pezzi con dimensione $l = [1/3 H, 1/2 H]$, $h = [1/3 H, 1/2 H]$

Se $M > 20$: pezzi con dimensione $l = [1/4 H, 1/2 H]$, $h = [1/4 H, 1/2 H]$

M/4 pezzi vengono cambiati invertendo l e h (permette di avere pezzi in "verticale" rispetto al piano).

M/4 ha l e h dimezzate (permette di avere pezzi più piccoli e disomogenei) e profitto diviso per 4 e ulteriormente diminuito (analogamente al caso reale in cui i pezzi più grandi sono preferiti).

Profitto = area * numero casuale nel range [1, 2]

Output

L'output avviene su ipertesto (html / css) , visualizzabile tramite browser con supporto CSS.

Nel file di output sono rappresentati:

- Elenco dei pezzi (colorati e in scala), con relative dimensioni, area, profitto
- Dimensione del piano
- Risultato: profitto totale, area scarti, riga di comando per riottenere gli stessi pezzi utilizzati
- Elenco indentato delle operazioni eseguite: taglio verticale o orizzontale con relativa posizione, pezzo inserito oppure scarto
- rappresentazione grafica piano di taglio con i pezzi (colorati e numerati) inseriti nella posizione finale

Scheletro dell'algoritmo

1. Input: dati piano LxH, numero pezzi M, dati dei pezzi;
2. Generazione pezzi casuali, se i dati in input non sono specificati;
3. Aggiungi M pezzi (ai M pezzi già presenti) ottenuti invertendo le dimensioni di quelli presenti (rotazione);
4. Calcola le posizioni di taglio possibili (ottenute da tutte le combinazioni possibili di tagli effettuabili con i pezzi presenti);
5. Per ogni $x = 1 \dots X, y = 1 \dots H$ memorizza in $F0[x][y]$ il numero del pezzo (col profitto maggiore) con $l < x$ e $h < y$. Se nessun pezzo ci sta, inserisci -1;
6. Inizializza le matrici (vedi terza colonna tabella seguente);
7. Calcola $f(X,Y)$ (vedi sotto: *scheletro di f*);
8. Crea albero delle operazioni di taglio, leggendo i dati dalle matrici (vedi tabella);
9. Stampa pezzi, albero delle operazioni, posizionamento dei pezzi sul piano;
10. Libera memoria dalle strutture allocate;

Matrici e significato

Nome	Dimensioni	inizializzazione	Significato
FX Y	(LxH)	-1	FX Y[x][y] contiene il risultato della chiamata $f(x,y)$
M1	(LxH)	S (scarto)	M1[x][y] memorizza l'operazione (a=taglio orizz b=taglio vertic p=pezzo s=scarto) che ha dato maggior profitto secondo la relativa $f(x,y)$
M2	(LxH)	-1	M2[x][y] memorizza dati aggiuntivi per l'operazione M1[x][y]. Contiene posizione di taglio (in caso di taglio) o numero del pezzo (in caso di posizionamento pezzo)

Scheletro di $f(x,y)$

{

Se $FX Y[x][y] > 0$ ritorna $FX Y[x][y]$ (esce dalla funzione);
Se $F0[x][y] = -1$ (nessun pezzo ci sta), ritorna 0 (esce dalla funzione);

Per $b=1 \dots y$ (b nell'insieme delle posizioni di taglio)
Calcola il valore massimo tra le chiamate $f(x,b) + f(x, y-b)$;

Per $a=1 \dots x$ (a nell'insieme delle posizioni di taglio)
Calcola il valore massimo tra le chiamate $f(a,y) + f(x-a, y)$;

Vedi il massimo tra le due operazioni precedenti e $F0[x][y]$ (profitto dal pezzo);
Scrivi in $M1[x][y]$ quale operazione ha dato il maggior profitto (taglio o pezzo);
Scrivi in $M2[x][y]$ la posizione del taglio o l'indice del pezzo;

}

Note

La memoria utilizzata è allocata inizialmente e non aumenta con le chiamate ricorsive della f .

Test

Di seguito alcuni test effettuati, scelti tra quelli che hanno dato un output graficamente più significativo. I risultati nelle tabelle. Dettagli (output) sui file html.

Ambiente di test:

A) Linux, AMD Athlon™ 64 Processor 3500+, 2211.340 Mhz, cache 512 kb, 4 Gb RAM

B) Windows XP, Pentium® D 3,4 Ghz, 4 Gb RAM

Problema	Piano	n. pezzi*	Pos.taglio risultanti**	Tempi calcolo (s)	
				ambiente A	ambiente B
1	1000x800	10	492	5,6	6,22
2	1000x800	20	631	7,7	8,19
3	1500x1000	10	760	13,9	16,2
4	1500x1000	20	1071	26,2	26,2
5	3000x2000	5	313	8	9,8
6	3000x2000	10	1360	106,12	109,3
7	3000x2000	20	1914	186,36	182,9

*I pezzi sono replicati invertendo le dimensioni (possono essere posizionati cioè in modo ruotato), quindi il vero numero di pezzi utilizzato nei calcoli è il doppio.

** numero di posizioni possibili di taglio, diversa per ogni insieme casuale di pezzi.

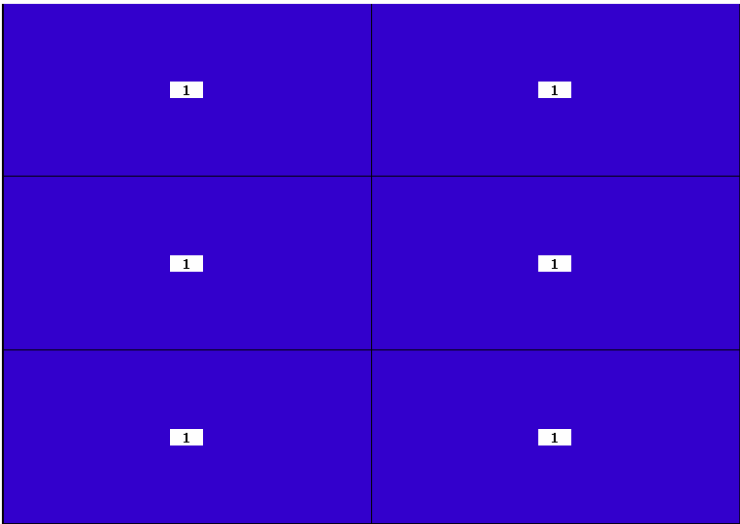
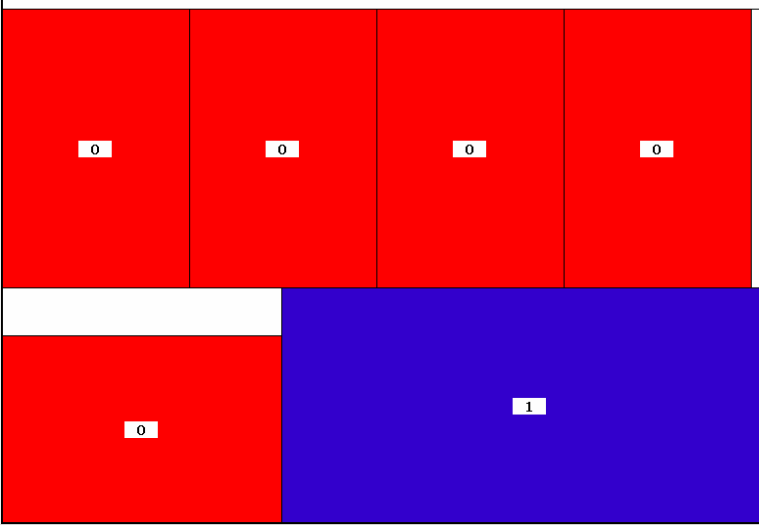
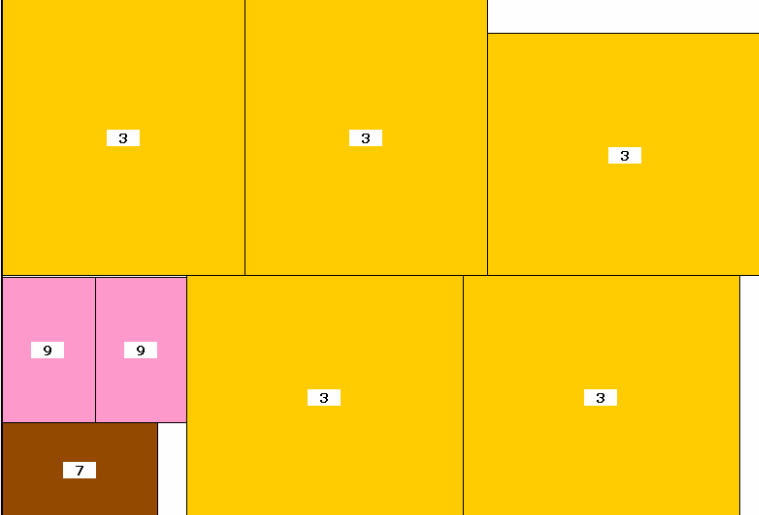
Dettaglio problemi

Problema	Linea di comando	File con input e risultato grafico*
1	--l 1000 --h 800 --m 10 --p 377 387 186870 467 397 227123 403 338 200008 466 281 185701 391 346 244767 274 356 138069 281 444 200548 231 142 46656 207 184 43441 194 187 56892	1.html
2	--l 1000 --h 800 --m 20 --p 385 391 231662 420 350 263018 282 345 110858 321 260 111628 302 323 183764 377 369 190797 355 319 197669 431 338 244585 284 312 153540 329 398 205214 213 462 180363 381 293 112576 220 421 104096 350 419 158170 389 323 144823 173 154 28682 241 175 46483 134 175 33221 248 166 49006 213 142 42939	2.html
3	--l 1500 --h 1000 --m 10 --p 726 487 599996 730 440 580728 549 444 449221 502 441 327028 749 346 296018 468 730 433975 376 612 344684 322 198 69417 362 225 106927 328 194 52423	3.html
4	--l 1500 --h 1000 --m 20 --p 405 450 357203 709 333 471679 375 258 172438 518 422 221680 396 280 179786 563 269 174928 655 286 278195 382 253 132580 618 456 322990 585 286 332552 418 617 462198 377 646 320869 253 407 151013 373 475 214056 270 603 295084 292 229 62684 216 232 40317 234 225 71693 218 147 33915 248 190 70692	4.html
5	--l 3000 --h 2000 --m 5 --p 1065 714 1501352 1894 898 3205004 1045 1332 1429592 803 660 691156 895 381 393290	5.html
6	--l 3000 --h 2000 --m 10 --p 1273 676 1608555 1293 986 1700260 1138 690 804525 1066 934 1947911 1232 898 1814609 908 1204 1176282 948 1143 1575048 597 363 311739 584 479 378931 557 355 270724	6.html
7	--l 3000 --h 2000 --m 20 --p 802 824 677315 1125 654 1340367 866 835 1174391 1414 530 969997 1146 833 1387874 1356 569 1453007 1389 689 1753750 1269 919 1646765 1219 596 1295356 1156 760 1450166 562 1142 915114 790 1245 1864907 656 987 1164640 922 1459 1418417 921 1017 967551 685 439 374764 544 310 142831 737 320 266783 670 304 275221 530 382 303263	7.html

*si riportano solo i files generati dalla configurazione A, i files generati dalla configurazione B sono identici tranne che per il tempo di calcolo.

Dettaglio problemi

P	Pezzi				Risultato		
1	PEZZO	L	H	V	<div>6</div> <div>5</div>	<div>4</div> <div>4</div>	<div>4</div> <div>4</div>
	0	377	387	186870			
	1	467	397	227123			
	2	403	338	200008			
	3	466	281	185701			
	4	391	346	244767			
	5	274	356	138069			
	6	281	444	200548			
	7	231	142	46656			
	8	207	184	43441			
	9	194	187	56892			
2	PEZZO	L	H	V	<div>4</div> <div>8</div> <div>10</div>	<div>4</div> <div>8</div> <div>10</div>	<div>4</div> <div>8</div> <div>10</div>
	0	385	391	231662			
	1	420	350	263018			
	2	282	345	110858			
	3	321	260	111628			
	4	302	323	183764			
	5	377	369	190797			
	6	355	319	197669			
	7	431	338	244585			
	8	284	312	153540			
	9	329	398	205214			
	10	213	462	180363			
	11	381	293	112576			
	12	220	421	104096			
	13	350	419	158170			
	14	389	323	144823			
	15	173	154	28682			
	16	241	175	46483			
	17	134	175	33221			
	18	248	166	49006			
	19	213	142	42939			
3	PEZZO	L	H	V	<div>2</div> <div>1</div>	<div>2</div> <div>1</div>	<div>2</div> <div>1</div>
	0	726	487	599996			
	1	730	440	580728			
	2	549	444	449221			
	3	502	441	327028			
	4	749	346	296018			
	5	468	730	433975			
	6	376	612	344684			
	7	322	198	69417			
	8	362	225	106927			
	9	328	194	52423			

4	PEZZO	L	H	V		
	0	405	450	357203		
	1	709	333	471679		
	2	375	258	172438		
	3	518	422	221680		
	4	396	280	179786		
	5	563	269	174928		
	6	655	286	278195		
	7	382	253	132580		
	8	618	456	322990		
	9	585	286	332552		
	10	418	617	462198		
	11	377	646	320869		
	12	253	407	151013		
	13	373	475	214056		
	14	270	603	295084		
	15	292	229	62684		
	16	216	232	40317		
	17	234	225	71693		
	18	218	147	33915		
	19	248	190	70692		
5	PEZZO	L	H	V		
	0	1065	714	1501352		
	1	1894	898	3205004		
	2	1045	1332	1429592		
	3	803	660	691156		
	4	895	381	393290		
6	PEZZO	L	H	V		
	0	1273	676	1608555		
	1	1293	986	1700260		
	2	1138	690	804525		
	3	1066	934	1947911		
	4	1232	898	1814609		
	5	908	1204	1176282		
	6	948	1143	1575048		
	7	597	363	311739		
	8	584	479	378931		
	9	557	355	270724		

7	PEZZO	L	H	V	
	0	802	824	677315	
	1	1125	654	1340367	
	2	866	835	1174391	
	3	1414	530	969997	
	4	1146	833	1387874	
	5	1356	569	1453007	
	6	1389	689	1753750	
	7	1269	919	1646765	
	8	1219	596	1295356	
	9	1156	760	1450166	
	10	562	1142	915114	
	11	790	1245	1864907	
	12	656	987	1164640	
	13	922	1459	1418417	
	14	921	1017	967551	
	15	685	439	374764	
	16	544	310	142831	
	17	737	320	266783	
	18	670	304	275221	
	19	530	382	303263	

Test a ritroso

Si sono effettuati test coprendo interamente (nessuno scarto) un piano con pezzi derivanti da tagli casuali.

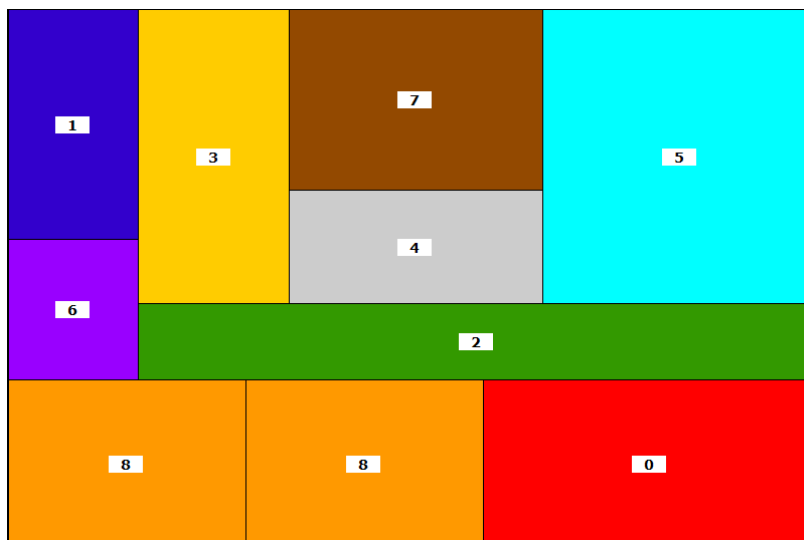
Si sono inseriti i pezzi manualmente, con profitto uguale all'area e verificato che l'algoritmo restituisse una soluzione equivalente (cioè nessuno scarto e con lo stesso profitto) . In tutti i casi si ha avuto esito positivo.

Per ogni caso si riporta il piano con i tagli casuali (pezzi non in scala, le dimensioni sono indicate in tabella), i dati dei pezzi e il risultato (in scala) del programma.

Caso 1)

	2		
1	3	4	5
6		7	
8	8	0	

PEZZO	L	H	A	V
0	62	31	1922	1922
1	24	43	1032	1032
2	126	14	1764	1764
3	28	55	1540	1540
4	47	21	987	987
5	51	55	2805	2805
6	24	26	624	624
7	47	34	1598	1598
8	44	31	1364	1364



vedi dimensioni pezzi e risultato elaborazione programma anche nel file caso1.html

caso 2)

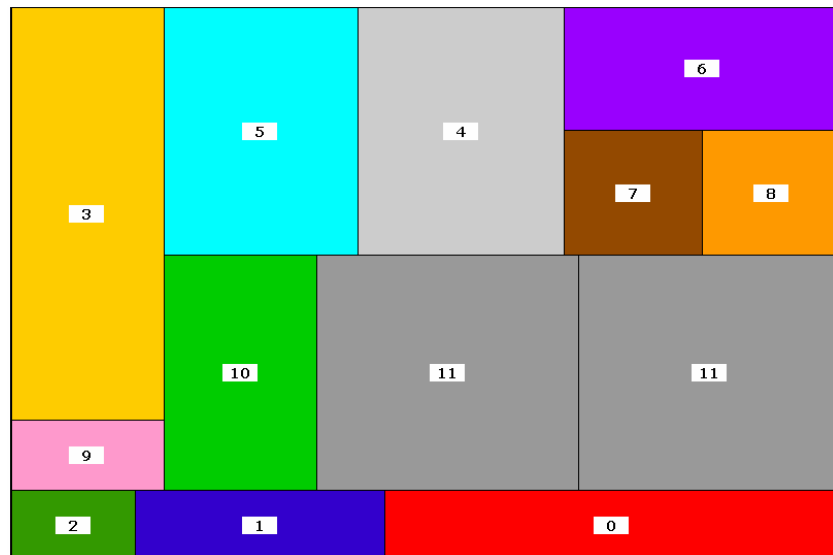
input come sopra (con dimensioni decuplicate)

vedi dimensioni pezzi e risultato elaborazione programma nel file caso2.html

caso 3)

0		1	2
3	4	5	6
			7 8
9	10	11	11

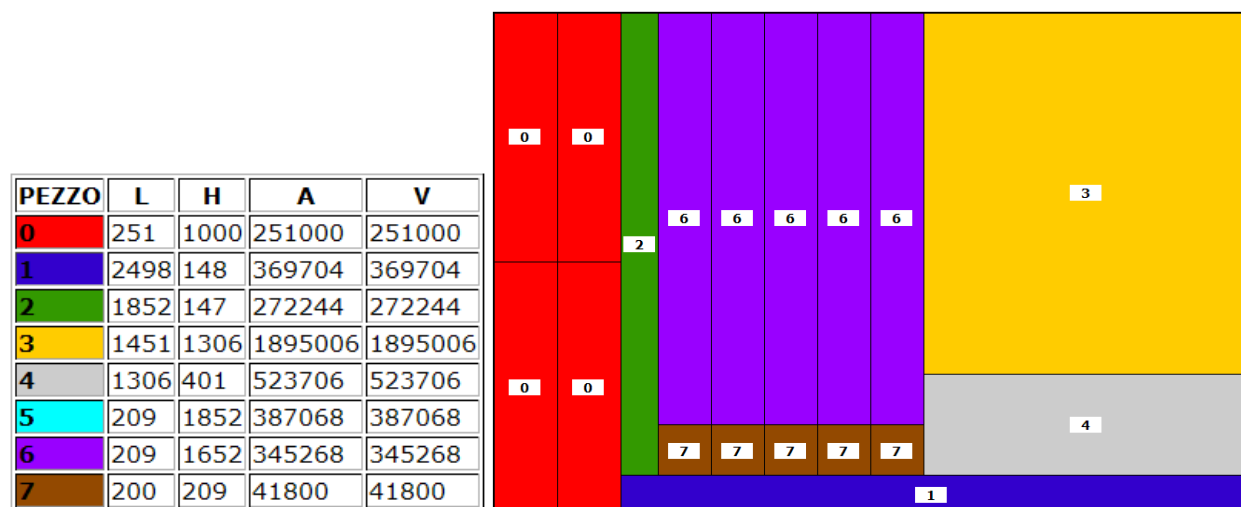
PEZZO	L	H	A	V
0	249	1648	410352	410352
1	905	249	225345	225345
2	447	249	111303	111303
3	550	1501	825550	825550
4	747	901	673047	673047
5	700	901	630700	630700
6	1003	451	452353	452353
7	500	450	225000	225000
8	503	450	226350	226350
9	550	250	137500	137500
10	552	850	469200	469200
11	850	949	806650	806650



vedi dimensioni pezzi e risultato elaborazione programma anche nel file caso3.html

caso 4)

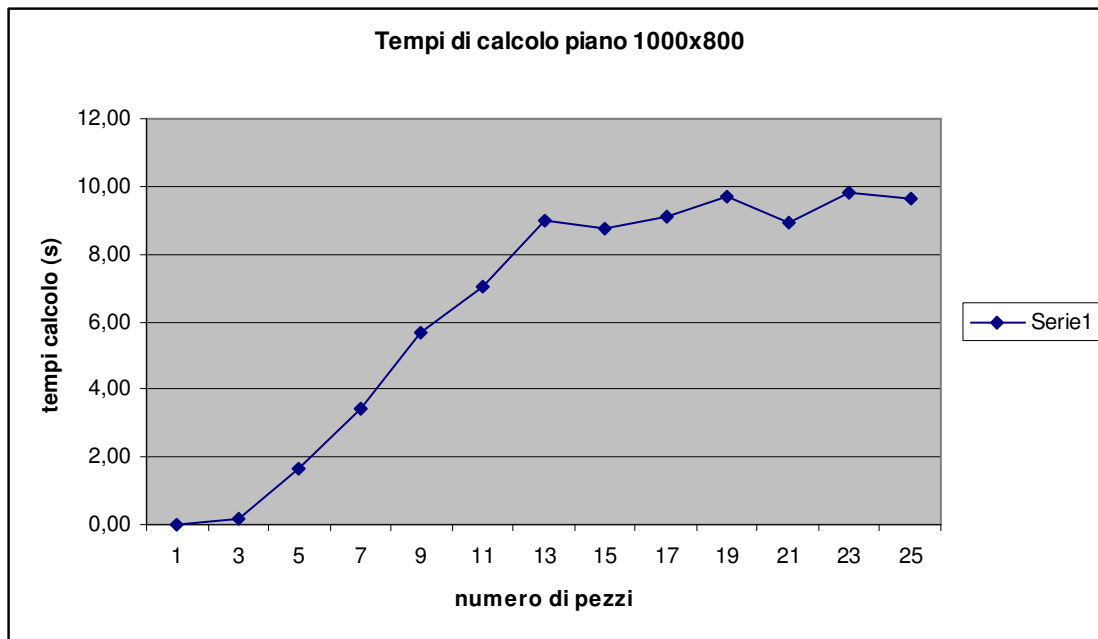
		1					
0	0	3					
0	0						
				5	5	5	5
							6
							7



PEZZO	L	H	A	V
0	251	1000	251000	251000
1	2498	148	369704	369704
2	1852	147	272244	272244
3	1451	1306	1895006	1895006
4	1306	401	523706	523706
5	209	1852	387068	387068
6	209	1652	345268	345268
7	200	209	41800	41800

vedi dimensioni pezzi e risultato elaborazione programma anche nel file caso4.html

Test con dimensione piano fissata e numero di pezzi crescente



Test con numero di pezzi fissato e dimensione piano crescente

