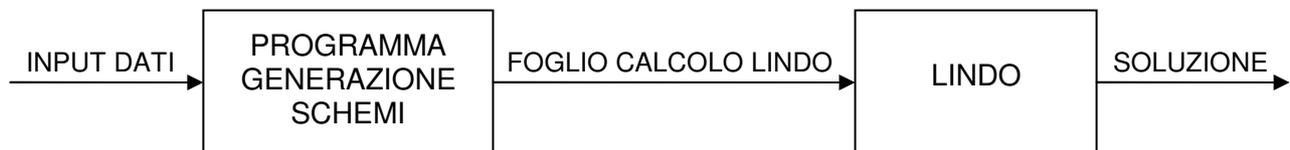


GENERATORE SCHEMI DI TAGLIO

Per la risoluzione del problema in esame si è ricorso all'utilizzo di due strumenti software:

- Software implementato in JAVA per la generazione automatica di schemi di taglio
- Software di risoluzione LINDO

L'applicativo di supporto implementato permette di generare automaticamente schemi di taglio per problemi di taglio ottimo monodimensionale e di fornire input per l'impostazione e la risoluzione del problema tramite il software LINDO.



3.1 Algoritmo generazione automatica schemi di taglio

Per realizzare una procedura che generi automaticamente i possibili schemi di taglio che caratterizzano un problema di taglio ottimo monodimensionale si è implementato un algoritmo che sviluppa tutte le possibili combinazioni tra le sezioni che devono essere estratte.

L'algoritmo si concentra fondamentalmente sulla concatenazione di cicli iterativi che definiscono tutti i possibili schemi di taglio del problema e successivamente valutano lo sforzo ottenuto per verificare che lo schema considerato sia ammissibile o meno.

Il linguaggio di programmazione di cui si è usufruito è il JAVA; di seguito verrà illustrata la procedura che rappresenta il nucleo dell'applicativo.

Per comprendere il meccanismo di funzionamento del software è conveniente considerare un semplice esempio di generazione di schemi di taglio, in questo modo sarà quindi più facile intuire l'automatismo di risoluzione sviluppato dal programma.

Consideriamo un semplicissimo problema di taglio ottimo di cui si devono individuare i possibili schemi di taglio, a tal proposito ipotizziamo i seguenti input:

Lunghezza barra su cui effettuare l'operazione di taglio = **10 cm**

Lunghezza 1° sezione di taglio (Tipo A) = **3 cm**

Lunghezza 2° sezione di taglio (Tipo B) = **2 cm**

si può facilmente dedurre da queste informazioni che i possibili schemi generabili sono 4 e cioè:

A = 3 | B = 0

A = 2 | B = 2

A = 1 | B = 3

A = 0 | B = 5

Per generare automaticamente tali schemi l'algoritmo implementato si articola nei seguenti passi:

- 1) Si individua tra le sezioni prese in esame quella di lunghezza minore (*minimotaglio*): nell'esempio considerato il *minimotaglio* è rappresentato dalla lunghezza della sezione B
- 2) Si calcolano i valori che definiscono il massimo numero delle volte che una sezione può essere contenuta nella barra su cui si effettuerà il taglio; queste informazioni verranno poi memorizzate in un vettore (*int[] max*).

In riferimento al semplice esempio considerato avremo:

- $max[0] = 3 \rightarrow$ la sezione di Tipo A potrà al massimo essere contenuta 3 volte all'interno della barra, essendo la barra lunga 10 cm con uno schema di taglio che prevede l'estrazione di 3 sezioni di Tipo A si produrrà uno sfrido di 1 cm ($10 \text{ cm} - 3 \times 3 \text{ cm} = 1 \text{ cm}$)
- $max[1] = 5 \rightarrow$ la sezione di Tipo B potrà al massimo essere contenuta 5 volte all'interno della barra, lo sfrido ottenuto sarà nullo ($10 \text{ cm} - 5 \times 2 \text{ cm} = 0 \text{ cm}$)

- 3) A questo punto è quindi possibile sviluppare cicli iterativi concatenati sulla base dei valori di massimo ricavati precedentemente; da tali iterazioni è possibile generare tutte le ottenibili combinazioni tra le sezioni. In riferimento al semplice esempio considerato dove le sezioni sono 2 (A e B) i cicli iterativi andrebbero implementati nel seguente modo:

```
FOR i = max[0],....., 0 //riferimento alla sezione di tipo A
  FOR j = max[1], .....0 //riferimento alla sezione di tipo B
    // Schema generato A = i | B = j
    // sfrido = Lunghezza Barra -
    //          ( i * (lunghezza sezione Tipo A) +
    //          j * (lunghezza sezione Tipo B) )
    // Memorizzazione dello schema in un vettore (schema)
    // schema[0] = i      schema[1] = j
  END
END
```

da queste iterazioni si otterrebbero quindi i seguenti schemi:

A = 3 B = 5	A = 2 B = 5	A = 1 B = 5	A = 0 B = 5
A = 3 B = 4	A = 2 B = 4	A = 1 B = 4	A = 0 B = 4
A = 3 B = 3	A = 2 B = 3	A = 1 B = 3	A = 0 B = 3
A = 3 B = 2	A = 2 B = 2	A = 1 B = 2	A = 0 B = 2
A = 3 B = 1	A = 2 B = 1	A = 1 B = 1	A = 0 B = 1
A = 3 B = 0	A = 2 B = 0	A = 1 B = 0	A = 0 B = 0

- 4) Come è facilmente intuibile non tutti gli schemi ottenuti dal passo precedente sono accettabili, gran parte degli schemi generati sono irrealizzabili ed è quindi necessario verificare la validità di tali configurazioni.

Considerato un qualunque schema S la validità di tale schema è data dalla verifica simultanea delle seguenti condizioni:

- Lo *sfrido* ottenuto applicando lo schema S è maggiore-uguale a 0
- Lo *sfrido* ottenuto applicando lo schema S è minore del *minimotaglio* (non è possibile estrarre alcuna altra sezione)

Dalla verifica di questi predicati si può vedere come nell'esempio considerato in cui gli schemi ottenuti al *passo 3* erano 24 soltanto 4 sono ammissibili.

Come visto, attraverso queste fasi, l'algoritmo è capace di individuare i possibili schemi di taglio; anche se l'esempio considerato risulta banale il principio di funzionamento può essere esteso anche a situazioni di complessità più elevata.

Sfruttando opportuni meccanismi di programmazione (ricorsione, vettori, ecc.) è possibile definire una funzione in grado di generare automaticamente schemi di taglio per problemi a n sezioni.

La funzione cardine del codice Java che esprime tale funzionalità è la seguente:

```
public void genera(int[] max,int z){
    for(int k=max[z];k>=0;k--){
        schema[z]=k; //memorizzazione indice k (numero applicazioni sezione z
                    //relative allo schema)
        if(z==max.length-1){
            if(sfrido(schema)<minimotaglio && sfrido(schema)>=0){
                //memorizza schema
                //memorizza corrispondente valore sfrido
            }
        }else genera(max,z+1); //chiamata ricorsiva
    }
}

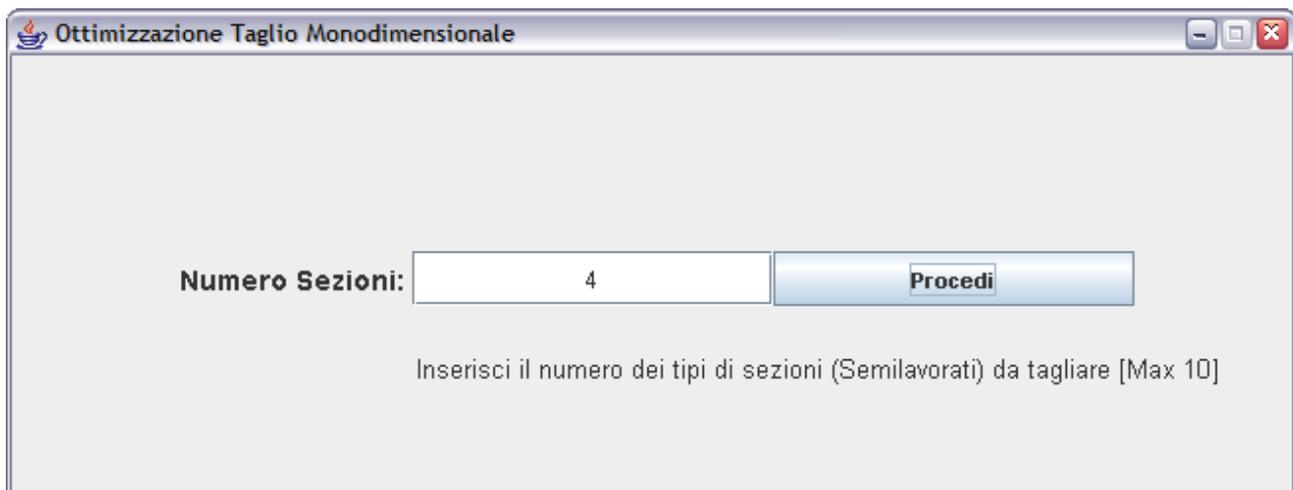
// Chiamata alla funzione: genera(max, 0);
```

3.2 Implementazione applicativo

L'algoritmo presentato è stato successivamente implementato come applicazione JAVA; per offrire una buona usabilità del software si è ricorso all'integrazione di una interfaccia grafica che consente di inserire dati di input e visualizzare risultati. L'interfaccia si suddivide principalmente in tre finestre attraverso le quali interagire con il programma.

Si vedrà a questo punto come usufruire dell'applicativo implementato per studiare un problema di taglio ottimo monodimensionale.

1 – FINESTRA INPUT NUMERO TIPO SEZIONI



questa è la prima finestra che si incontra durante l'uso del software: si deve inserire il numero dei tipi di sezioni (A, B, C, D,) attraverso le quali generare gli schemi di taglio.

Per il caso di studio *SuperChef* abbiamo 4 tipologie di prodotti da estrarre e quindi come input inseriremo il valore numerico 4.

2 – FINESTRA INPUT DATI COMMESSA

The screenshot shows a software window titled "Ottimizzazione Taglio Monodimensionale - Input Dati". It contains a form with the following data:

Materia Prima	Lunghezza (cm)	
Barra	304.8	

Pezzo	Lunghezza (cm)	Quantità
A	91.44	30
B	85.344	45
C	137.16	27
D	67.056	60

At the bottom of the window is a button labeled "Genera Schemi Di Taglio".

Attraverso questa finestra di input è possibile inserire i dati relativi all'ordine ricevuto, una volta inseriti i dati è poi possibile avviare la procedura di generazione automatica degli schemi di taglio.

NOTA: Se le tipologie di sezioni fossero state 10 allora la finestra di input si sarebbe presentata nel seguente modo:

The screenshot shows the same software window, but with an empty input form for 10 piece types (A through L). The data is as follows:

Materia Prima	Lunghezza (cm)	
Barra		

Pezzo	Lunghezza (cm)	Quantità
A		
B		
C		
D		
E		
F		
G		
H		
I		
L		

At the bottom of the window is a button labeled "Genera Schemi Di Taglio".

3 – VISUALIZZAZIONE RISULTATI ED ESPORTAZIONE CODICE LINDO

Ottimizzazione Taglio Monodimensionale - Generazione Schemi

Generazione Schemi Di Taglio:

Lunghezza Barra: 304.8 cm

A=91.44 cm B=85.344 cm C=137.16 cm D=67.056 cm

Numero schemi generati: 14

SCHEMI DI TAGLIO

	A	B	C	D	Sfrido
X1	3	0	0	0	30.48
X2	2	1	0	0	36.576
X3	2	0	0	1	54.864
X4	1	2	0	0	42.672
X5	1	1	0	1	60.96
X6	1	0	1	1	9.144
X7	1	0	0	3	12.192
X8	0	3	0	0	48.768
X9	0	2	0	2	0.0
X10	0	1	1	1	15.24
X11	0	1	0	3	18.288
X12	0	0	2	0	30.48
X13	0	0	1	2	33.528
X14	0	0	0	4	36.576

Esportazione Foglio Di Calcolo LINDO

FUNZIONE OBIETTIVO

Minimizzazione Sfrido Minimizzazione Semilavorati

IMPOSTAZIONE VINCOLI

Uguaglianza Sovrapproduzione %

Esporta Codice Lindo

attraverso questa finestra è possibile visualizzare i risultati ottenuti dalla procedura di generazione, inoltre è possibile esportare il foglio di calcolo già impostato per poi avviare la risoluzione tramite il software LINDO.

La fase di esportazione consente di scegliere a quale funzione obiettivo far riferimento e come impostare i vincoli del problema.

Se ad esempio nel problema si prevede una sovrapproduzione del 20% è possibile impostare i vincoli in modo che il foglio di calcolo esportato consideri questa opportunità:

Esportazione Foglio Di Calcolo LINDO

FUNZIONE OBIETTIVO

Minimizzazione Sfrido Minimizzazione Semilavorati

IMPOSTAZIONE VINCOLI

Uguaglianza Sovrapproduzione %

Esporta Codice Lindo

L'esportazione consente quindi di eseguire direttamente il risoluzione del problema tramite LINDO senza dover impostare manualmente i vincoli e la funzione obiettivo.