Il Problema dei Tagli Planari

Poderico Luigi

Versione 1.0
27/12/95

Introduzione

Uno dei particolari compiti delle industrie che lavorano la carta è l'intaglio di etichette. Queste vengono ritagliate da dei fogli di carta rettangolari mediante **tagli planari**.

Un foglio (di carta) è **tagliato** quando, mediante una qualunque tecnologia, esso viene diviso in due parti. Chiamiamo **traiettoria del taglio** il luogo dei punti dove incide il taglio. Quando la traiettoria di un taglio è un segmento parallelo ad uno dei lati del foglio, il taglio è detto **planare**.

Le etichette da produrre, approssimate sempre mediante dei rettangoli, vengono ricavate da k fogli sovrapposti, mediante una successione di tagli planari. Il procedimento può essere formalizzato nel seguente modo:

```
procedure Intaglio;

begin
    Q:={R_Iniziale};
    Sol:=AlberoDiRadice(R_Iniziale);
    while Q≠∅ do
        begin
            select R∈Q;
            (R1, R2) = TaglioPlanare(R);
            Q:=Q\{R};
            if R2≠NULL
                then
                    begin
                        Q:=Q∪{R1, R2};
                        Sol.AggiungiFiglio(R, R1);
                        Sol.AggiungiFiglio(R, R2)
                    end
        end
end.
```

dove:

- Q è un insieme di rettangoli;
- Sol è un albero binario che rappresenta la successione di tagli planari;
- $R_{Iniziale}$ è il rettangolo dal quale devono essere ritagliate le etichette;
- la funzione TaglioPlanare prende in input un rettangolo R e restituisce:
  ⇒ una coppia di rettangoli (R_1, R_2) se R deve essere tagliato;
  ⇒ una coppia del tipo (R, NULL) se R non deve essere tagliato.

Nel seguito indicheremo con:

- n il numero di tipologie di etichette da intagliare;
- $R_1,...,R_n$ i rettangoli (etichette) da intagliare;
- $m_1,...,m_n$ il numero di rattangoli, rispettivamente relativi ai $R_1,...,R_n$, da intagliare;
- R il rettangolo che descrive il foglio dal quale intagliare le etichette;
- k^* il numero di fogli R necessari per intagliare m_1 etichette di tipo R_1, m_2 etichette di tipo R_2,..., m_n etichette di tipo R_n.

Il procedimento descritto ammette varie implementazioni; scopo del presente lavoro è quello di proporne una che:

- determini una successione di tagli planari che minimizzi lo spreco di carta (definito come la quantità di carta inutilizzata) e permetta di tagliare le etichette richieste;
- disponga fisicamente vicino le etichette uguali all'interno di *R*;
- disponga alcune tipologie di etichette ai bordi di *R*.

Disposizioni

Una successione di tagli planari partiziona *R* in un insieme di rettangoli classificabili come:

- **necessari** alla produzione delle etichette;
- **sfrido** cioè non necessari alla produzione .

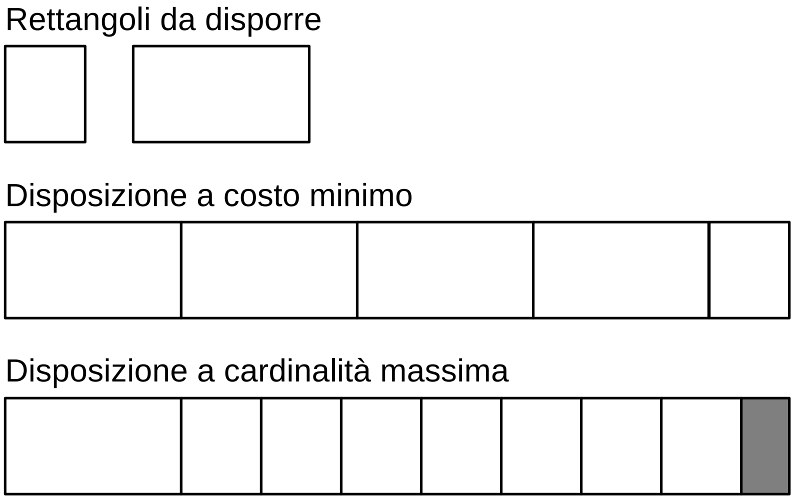
Se limitiamo l'attenzione ai soli rettangoli necessari, possiamo dire che una succesione di tagli planari rappresenta un modo per **disporre** sopra *R* gli elementi di una lista *D*, contenete ogni $R_1,\ldots,R_n$ almeno una volta. Viceversa, non sempre una succesione di tagli planari è generabile a partire da una disposizione di rettangoli. Definiamo **disposizioni ammissibili** tutte e solo le disposizioni generabili da una successione di tagli planari.

Associamo ad ogni disposizione *D* un *costo c(D)* pari alla differenza tra l'area di *R* e la somma delle aree dei rettangoli in *D*:

$$c(D) = R.area - \sum_{r \in D} r.area \quad ;$$

dove *R.area* indica l'area del rettangolo R

E' facile verificare che le disposizioni a costo minimo sono, in generale, diverse da quelle di cardinalità massima.



Studiamo ora che tipo di relazione intercorre tra una disposizione ammissibile e lo spreco che essa comporta. Indicando con $\|D|R_i\|$ il numero di rettangoli del tipo R_i contenuti nella lista *D*, il numero di fogli necessari per intagliare le etichette richieste è:

$$k(D) = \max_{i=1,\ldots,n} \left\lceil \frac{m_i}{\|D|R_i\|} \right\rceil .$$

Da ciò si ricava che lo **spreco** indotto da una disposizione ammissibile *D* è:

$$\underbrace{k(D)\cdot c(D)}_{\text{Sfrido}} + \underbrace{\sum_{i=1}^{n}\left(k(D)\cdot\left\|D|R_i\right\| - m_i\right)\cdot R_i.area}_{\text{Etichette in sovraproduzione}} =$$

$$k(D)\cdot\left(R.area - \sum_{i=1}^{n}\left\|D|R_i\right\|\cdot R_i.area\right) + \sum_{i=1}^{n}\left(k(D)\cdot\left\|D|R_i\right\| - m_i\right)\cdot R_i.area =$$

$$k(D)\cdot R.area - k(D)\sum_{i=1}^{n}\left\|D|R_i\right\|\cdot R_i.area +$$

$$k(D)\sum_{i=1}^{n}\left\|D|R_i\right\|\cdot R_i.area - \sum_{i=1}^{n} m_i\cdot R_i.area =$$

$$k(D)\cdot R.area - \sum_{i=1}^{n} m_i\cdot R_i.area \; .$$

I conti appena fatti ci permettono di fare le seguenti considerazioni:

- il vero obiettivo da perseguire nella determinazione di una successione di tagli planari è la minimizzazione del numero di fogli necessari:

$$k^{*} = \min_{D\in\mathcal{D}} k(D) = \min_{D\in\mathcal{D}} \max_{i=1,\ldots,n} \left\lceil \frac{m_i}{\left\|D|R_i\right\|} \right\rceil$$

  dove $\mathcal{D}$ rappresenta l'insieme delle disposizioni ammissibili;

- una volta determinato k^{*}, possono essere calcolate diverse disposizioni tali da minimizzare lo sfrido aumentando il numero di etichette in sovraproduzione.

Un algoritmo di risoluzione

Per la determinazione di una disposizione ammissibile che minimizzi il numero di fogli necessari, proproniamo il seguente algoritmo:

```
procedure TagliPlanari;

for i:=1 to n do
        D[i]:=1;
h:=0;
while DisposizioneAmmissibile(D) do
        begin
                h:=argmax{m[i]/D[i] | i=1,...,n};
                D[h]:=D[h]+1
        end
if h≠0 then
        D[h]:=D[h]-1
else
        Error("Problema inammissibile");
```

dove la funzione DisposizioneAmmissibile(D) ritorna TRUE se e solo se *D* ammette una disposizione ammissibile su *R*.

L'algoritmo appena esposto è di tipo greedy; ad ogni iterazione cerca di aggiungere alla disposizione il rettangolo che fa diminuire il numero di fogli necessari. Nell'ipotesi in cui la funzione DisposizioneAmmissibile(D) risolva senza approssimazioni la ricerca di una disposizione ammissibile, la procedura TagliPlanari dovrebbe dare soluzioni di buona qualità.

Notiamo che l'ipotesi fatta sulla funzione DisposizioneAmmissibile(D) è molto forte, in quanto il problema della ricerca di una disposizione ammissibile è NP-Completo.

Generazione di disposizioni ammissibili

Una disposizione può essere rappresentata come una lista D contenente i rettangoli da disporre e soddisfacente la proprietà:

$$R_i \in D \qquad \forall i = 1, \dots, n \, .$$

Ricordiamo che le disposizioni D che vogliamo generare devono soddisfare i seguenti **vincoli di ammissibilità**:

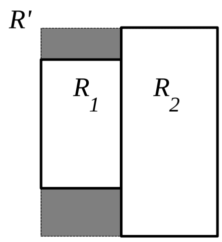
- D deve essere ammissibile, cioè partendo dal rettangolo iniziale R e applicando una successione di tagli planari, vengono costruiti gli elementi della lista D;
- disponga fisicamente vicino le etichette uguali all'interno di R;
- disponga alcune tipologie di etichette ai bordi di R.

Il nostro intento è quello di fornire un modo algoritmico ed efficiente per enumerare gli elementi di $\mathcal{D}$, costruendo una particolare funzione intorno. Inoltre verranno definite particolare funzioni su $\mathcal{D}$ a volori reali, che permetteranno di ricercare le disposizioni che soddisfano i vincoli di ammissibilità.

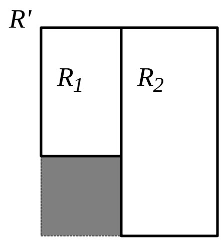
Gli operatori di giustapposizione

Descriviamo di seguito tre operatori che risulteranno indispensabili per la definizione di un intorno su $\mathcal{D}$: giustapposizione orizzontale, giustapposizione verticale e ruotazione.

L'operatore di **giustapposizione orizzontale** (denotato anche con il simbolo +), prende due rettangoli, R_1 ed R_2, e restituisce il più piccolo rettangolo R' che contiene R_1 ed R_2 giustapposti orizzontalmente.

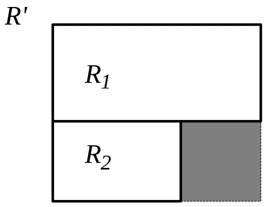


La disposizione verticale di R_1 ed R_2 per la costruzione di R_1+R_2 non è influente al fine di minimizzare lo sfrido; mentre è influente per minimizzazione del numero di tagli necessari. Infatti, nell'esempio, i tagli necessari sono 3, mentre ne basterebbero 2.



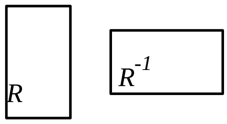
In base a questa considerazione, conveniamo che l’operatore di giustapposizione orizzontale allinea verso l’alto i suoi due operandi.

L’operatore di **giustapposizione verticale** (denotato anche con il simbolo *), prende due rettangoli, R_1 ed R_2, e restituisce il più piccolo rettangolo R' che contiene R_1 ed R_2 giustapposti verticalmente.



Per le stesse considerazioni fatte per l’operatore di giustapposizione verticale, assumiamo per convenzione che l’allineaminto degli operandi avvenga verso sinistra.

L’ultimo operatore che presentiamo è quello di **rotazione** (indicato con il simbolo $^{-1}$), che prende un rettangolo e lo ruota di 90°.



E’di facile verifica la seguente:

Proposizione Se R_1 ed R_2 sono due disposizioni ammissibili, allora:

- R_1+R_2 è ammissibile;
- R_1*R_2 è ammissibile;
- R_1^{-1} è ammissibile.

■

Descrizione di disposizioni ammissibili

Studiamo la seguente **grammatica di generazione**, che usa gli operatori di giustapposizione nella forma polacca inversa:

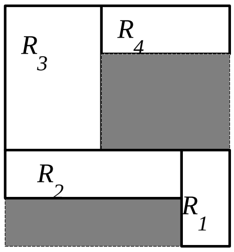
disposizione ::= disposizione disposizione +
| disposizione disposizione *
| rettangolo

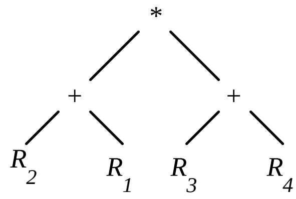
rettangolo ::= rett | rett^{-1}

rett ::= R_1 | R_2 | ... | R_n

E’ facilmete provabile per induzione che la grammatica di generazione è *corretta*, cioè descrive delle disposizione ammissibili.

Cerchiamo di verificare se è anche *completa*, cioè se genera tutte le disposizioni ammissibili. Consideriamo la procedura Intaglio presentata nell’introduzione; questa, per ogni possibile successione di tagli planari, genera un albero Sol che la descrive. Visitando Sol in ordine posticipato, associando ai tagli orizzontali l’operatore * ed ai tagli verticali l’operatore +, descriviamo un espressione generabile dalla grammatica di generazione.





$$R_2\, R_1 + R_3\, R_4 + *$$

Nella verifica appena mostrata, è implicita l'esistenza di corrispondaneza biunivoca tra le espressioni generabile mediante la grammatica di generazione e gli **alberi di generazione** definiti come alberi binari con:

- i nodi appartenenti all'insieme {+, *};
- le foglie appartenenti all'insieme $\{R_1, R_2, ..., R_n, R_1^{-1}, R_2^{-1}, ..., R_n^{-1}\}$.

Una funzione intorno

Consideriamo le seguenti regole di trasformazione di alberi di generazione:

1) scambio di due sottoalberi;
2) un nodo + diventa nodo * e viceversa;
3) una foglia R_i diventa R_i^{-1} e viceversa.

E' abbastanza evidente come queste tre regole definiscono un intorno sull'insieme degli alberi di generazione. Infatti basta considerare il fatto che ogni albero di generazione ha lo stesso numero *n* di foglie.

Poiché esiste una corrispondenza biunivoca tra gli alberi di generazione e le espressioni generate dalla grammatica di generazione, l'intorno su gli alberi di generazione induce un intorno sulle espressioni della grammatica di generazione.

Funzioni indice

Come abbiamo già detto, tra le disposizioni ammissibili $\mathcal{D}$ dobbiamo cercare quelle che verificano determinati vincoli. Al fine di guidare la ricerca verso queste disposizioni, presentiamo due funzioni indice.

Le convenzioni che useremo sono:

- operatore di proiezione cartesiano:

$$(x_1, \ldots, x_n) \downarrow i = x_i$$

- una funzione d'appoggio *bh*:

$$bh: \mathbf{R}^2 \to \mathbf{R}$$

$$bh(x, y) = x \cdot y$$

- metodi sui rettangoli:

$$R_i.base = \text{valore della base di } R_i;$$

$$R_i.altezza = \text{valore della base di } R_i;$$

$$R_i.area = R_i.base \cdot R_i.altezza;$$

$R_i.baricentro.x$ = ascissa del baricentro di R_i;

$R_i.baricentro.y$ = ordinata del baricentro di R_i.

- metodi sulle disposozioni - indicato con *R’* il più piccolo rettangolo che inscrive una disposizione *D*, abbiamo:

D.base = *R’.base*;

D.altezza = *R’.altezza*;

D.area = *R’.area*;

D.baricentro.x = *R’.baricentro.x*;

D.baricentro.y = *R’.baricentro.y*.

Indicatore 1: le disposizioni generate sono contenute in *R*

L’indicatore che andiamo a descrivere è una funzione f_R che data una disposizione *D*, calcola il più piccolo rettangolo che la contiene. Per descrivere la f_R useremo una definizione ricorsiva guidata dalla sintassi:

$$f_R\left[D_1\ D_2\ +\right] = \left(f_R\left[D_1\right]\downarrow 1 + f_R\left[D_2\right]\downarrow 1, \max\left(f_R\left[D_1\right]\downarrow 2, f_R\left[D_2\right]\downarrow 2\right)\right)$$

$$f_R\left[D_1\ D_2\ *\right] = \left(\max\left(f_R\left[D_1\right]\downarrow 1, f_R\left[D_2\right]\downarrow 1\right), f_R\left[D_1\right]\downarrow 2 + f_R\left[D_2\right]\downarrow 2\right)$$

$$f_R\left[R_i^{-1}\right] = \left(f_R\left[R_i\right]\downarrow 2, f_R\left[R_i\right]\downarrow 1\right)$$

$$f_R\left[R_i\right] = \left(R_i.base, R_i.altezza\right)$$

Quindi delle disposizioni $D \in \mathcal{D}$, scartiamo quelle tali che:

$$f_R[D]\downarrow 1 > R.base \vee f_R[D]\downarrow 2 > R.altezza \ .$$

Indicatore 2: le disposizioni generate hanno determinati rettangoli sui bordi

Sia *m*, con $m \in [0, n]$, l’ultimo indice dei rettangoli che vanno disposti più vicino possibile al bordo di *R*:

- $i \in [0, m] \Rightarrow R_i$ va disposto più vicino possibile al bordo di *R*;
- $i \in [m+1, n] \Rightarrow R_i$ non ha vincoli di disposizione.

Il secondo indicatore è defintito come:

$$f_b[D] = \sum_{\substack{R \in D \\ R = R_i\ i \in [0,m]}} \sqrt{\left(R.baricentro.x - D.baricentro.x\right)^2 + \left(R.baricentro.y - D.baricentro.y\right)^2} \ .$$

Delle disposizioni $D \in \mathcal{D}$ vanno considerate quelle che massimizzano la $f_b[D]$.

Indicatore 3: le disposizioni generate hanno i rettangoli uguali vicini

Il terzo ed ultimo indicatore è definito come:

$$f_s[D] = \sum_{\substack{R,S\in D\\ R\neq S\\ R=R_i=S}} \sqrt{\left(R.baricentro.x - S.baricentro.x\right)^2 + \left(R.baricentro.y - S.baricentro.y\right)^2}.$$

Un modello per il problema dei tagli planari

Tutte le informazioni finora date possono essere condensate nel seguente modello matematico per il problema dei tagli planari:

$$\begin{array}{ll} \min & f_s[D] - f_v[D] \\ t.c. & \\ & f_R[D]\downarrow 1 \leq R.base \\ & f_R[D]\downarrow 2 \leq R.altezza \\ & D \in \mathcal{D} \end{array}.$$