

# Laboratorio di Algoritmi

## Progetto “Lamiere” (gennaio 2024)

**Nota:** La scadenza del progetto è fissata per venerdì 12 gennaio **compreso**.

**Nota:** Si consiglia di controllare gli eventuali aggiornamenti di questo documento, che potrebbero riportare le risposte a dubbi degli studenti e la correzione di eventuali errori, e di consultare il documento con le indicazioni tassative e le informazioni utili per svolgere e consegnare correttamente il progetto.

**Il problema** Un impianto per la produzione di infissi opera trasformando lamiera di vari spessori e materiali metallici in pezzi molto variegati, spesso unici. Il progetto si propone di contribuire alla fase iniziale della produzione, cioè il taglio delle lamiere in sagome, che poi verranno piegate e assemblate per ottenere i pezzi finali. Poiché le fasi successive non interessano, nel seguito intenderemo con “produzione” solo la fase di taglio e con “pezzi” le sagome bidimensionali ritagliate dalle lamiere.

La produzione deve svolgersi in un dato orizzonte temporale, costituito da un insieme di giorni, rappresentati dai numeri interi compresi fra 1 e  $T$ . L'impianto contiene un dato insieme di macchine, le cui caratteristiche sono note a priori. In particolare, ogni macchina ha una *velocità di taglio* (in millimetri al secondo), che dipende dallo spessore della lamiera, ma non dal materiale (per semplicità), in base a una formula descritta in dettaglio nel seguito. Inoltre, in ciascun giorno dell'orizzonte è noto il *tempo totale di attività* di ciascuna macchina, espresso in secondi. Se una macchina non è attiva in un dato giorno, tale tempo è nullo. Si suppone che sia sufficiente conoscere il tempo totale, mentre gli orari precisi non sono necessari. L'impianto ha un elenco di pezzi da produrre, che specifica per ognuno:

- il materiale di cui è composto;
- lo spessore della lamiera con cui va prodotto, espresso in millimetri;
- il perimetro di taglio, che comprende anche eventuali buchi interni, espresso in millimetri;
- la *release date*, cioè l'indice intero del giorno prima del quale non può essere tagliato;
- la *deadline*, cioè l'indice intero del giorno dopo il quale non può essere tagliato;
- il sottoinsieme delle macchine che lo possono produrre.

Il progetto richiede per prima cosa di raccogliere alcune informazioni generali sul problema, in particolare sulla relazione fra domanda e offerta, fra la richiesta di pezzi e la disponibilità delle macchine in grado di produrli. Queste informazioni sono descritte in maggior dettaglio nel seguito.

Il progetto dovrà quindi definire un piano di produzione per l'impianto, indicando per ogni pezzo dell'elenco dato se viene prodotto o no (la richiesta potrebbe eccedere la capacità dell'impianto); in caso positivo, bisogna indicare in quale giorno (l'orario non importa) e su quale macchina. Non ci sono garanzie che le macchine compatibili con un pezzo siano disponibili nell'intervallo di tempo in cui esso è producibile, né che la capacità produttiva delle macchine sia sufficiente per tutti i pezzi. Infatti, l'obiettivo generale del piano è minimizzare il numero di pezzi non prodotti. In pratica, tali pezzi verranno prodotti in ritardo, danneggiando il livello di servizio, oppure verranno assegnati a un produttore esterno, con un aumento dei

costi. Siccome questo problema di ottimizzazione è computazionalmente difficile, si ricorrerà ad algoritmi euristici, descritti nel seguito, che non offrono garanzia di ottimalità.

**Il progetto** I dati del progetto sono contenuti in un un unico file di testo. La prima riga del file fornisce il numero di giorni  $T$  dell'orizzonte temporale, seguito dalla parola chiave **giorni**. Ad esempio:

```
3 giorni
```

indica che l'orizzonte temporale si compone dei giorni 1, 2 e 3.

Quindi, il file riporta il numero di macchine disponibili, seguito dalla parola chiave **macchine**. Ciascuna delle righe seguenti fornisce le informazioni associate ad una macchina:

1. l'indice numerico intero  $m$ , crescente da 1 al numero delle macchine;
2. un coefficiente reale  $r_m$  (espresso in  $\text{mm}^2/\text{sec}$ ) che consente di calcolare i tempi di taglio sulla macchina  $m$  nel modo descritto nel seguito;
3. il tempo di attività  $d_{mg}$  della macchina  $m$  in ciascun giorno  $g$  dell'orizzonte temporale, espresso in secondi come numero intero.

Ad esempio:

```
2 macchine
1 53.4 180 0 500
2 36.6 400 200 0
```

indica che l'impianto contiene due macchine. La macchina 1 ha coefficiente di taglio pari a  $53.4 \text{ mm}^2/\text{sec}$ , è attiva per 180 secondi nel giorno 1, non è attiva nel giorno 2 e infine è attiva per 500 secondi nel giorno 3. Analogamente, la macchina 2 ha coefficiente pari a  $36.6/s \text{ mm}^2/\text{sec}$ , è attiva per 400 secondi nel giorno 1, per 200 secondi nel giorno 2 e non è attiva nel giorno 3.

Seguono le informazioni sui pezzi. La prima riga fornisce il numero dei pezzi da produrre, seguito dalla parola chiave **pezzi**. Ciascuna delle righe seguenti fornisce le informazioni associate ad un pezzo:

1. l'indice numerico intero  $p$ , crescente da 1 al numero dei pezzi;
2. il materiale, descritto da un'unica parola, che può essere solo **bronzo**, **ferro** o **alluminio**;
3. lo spessore  $s_p$ , espresso in millimetri come numero intero;
4. il perimetro complessivo  $p_p$ , espresso in millimetri come numero intero;
5. gli indici interi dei giorni che fanno da *release date* e da *deadline*;
6. l'elenco degli indici interi delle macchine che possono produrre il pezzo, ordinato per valori crescenti<sup>1</sup>.

---

<sup>1</sup>Consiglio di semplificare la lettura caricando la riga in una stringa da 256 caratteri, ipotizzata sufficientemente lunga, e applicando iterativamente l'istruzione `sscanf(s, "%d%[\n]", &m, s)`; per caricare un numero intero nella variabile `m` e il resto della stringa ancora in `s`, in modo da cancellare l'intero e preparare la lettura dell'intero seguente. In alternativa, ci si può addentrare nelle finezze dell'istruzione `strtok`.

Ad esempio:

10 pezzi

1 bronzo 6 669 1 2 1

2 bronzo 10 397 2 3 2

3 ferro 6 2427 1 2 1 2

4 alluminio 2 2228 2 3 2

5 bronzo 8 2024 2 3 1 2

6 ferro 5 2372 2 3 1 2

7 bronzo 8 1433 1 2 1 2

8 ferro 6 2236 1 2 1

9 alluminio 9 2352 2 3 2

10 ferro 3 1633 2 3 1

indica che il pezzo 1 è fatto di bronzo, ha uno spessore di 6 mm, un perimetro di 669 mm, deve essere prodotto non prima del giorno 1 e non più tardi del giorno 2 e può essere prodotto solo sulla macchina 1. Per fare un altro esempio, il pezzo 6 è fatto di ferro, ha uno spessore di 5 mm, un perimetro di 2372 mm, deve essere prodotto non prima del giorno 2 e non più tardi del giorno 3 e può essere prodotto su entrambe le macchine disponibili.

Il fattore principale che influisce sul piano di produzione è il tempo impiegato a tagliare i pezzi. Le altre operazioni, come il carico e lo scarico delle lamiere, i movimenti a vuoto della testina di taglio, l'ingresso della testina nella lamiera al principio e l'uscita alla fine del taglio di ogni singolo pezzo, vengono per semplicità ignorate. Il tempo di taglio  $t_{pm}$  del pezzo  $p$  sulla macchina  $m$  (espresso in secondi) è invece descritto dalla formula

$$t_{pm} = \left\lceil \frac{p_p s_p}{r_m} \right\rceil$$

dove  $p_p$  e  $s_p$  sono il perimetro e lo spessore del pezzo  $p$  (espressi in millimetri) e  $r_m$  è il coefficiente della macchina  $m$  (espresso in  $\text{mm}^2/\text{sec}$ ). Per sicurezza e per semplicità, il risultato viene arrotondato all'intero superiore. Per esempio, una macchina con coefficiente pari a  $53.4/s \text{ mm}^2/\text{sec}$  impiega 10 secondi a tagliare una lunghezza di 534 mm in una lamiera dello spessore di 1 mm, mentre impiega 20 secondi a tagliare una lunghezza di 267 mm in una lamiera dello spessore di 4 mm, e così via.

Caricati i dati, il progetto richiede per prima cosa di costruire la relazione di compatibilità fra domanda e offerta. Questa relazione lega due insiemi di entità:

1. i pezzi  $p$  da produrre;
2. gli *slot* disponibili, ovvero le coppie  $(m, g)$  tali che la macchina  $m$  sia attiva nel giorno  $g$ .

Uno *slot*  $(m, g)$  è compatibile con un pezzo  $p$  quando:

- la macchina  $m$  può produrre il pezzo  $p$ ;
- il giorno  $g$  appartiene all'intervallo di produzione del pezzo  $p$ .

Il programma deve stampare il numero di *slot* totale, seguito dalla parola chiave `slot`. Quindi, deve stampare l'elenco degli *slot*, ordinati per tempo di attività  $d_{mg}$  decrescente. La logica dell'ordinamento è di considerare per primi gli `slot` probabilmente più utili, in quanto più lunghi. In caso di parità, si ordinano gli `slot` lessicograficamente, cioè prima per indice di macchina crescente, poi per indice di

giorno crescente. Ogni *slot* occupa una riga e bisogna stampare l'indice di macchina  $m$ , quello del giorno  $g$  e il tempo di attività  $d_{mg}$ , separati da spazi. Per esempio:

```
4 slot
1 3 500
2 1 400
2 2 200
1 1 180
```

Quindi, il programma deve stampare il numero dei pezzi, seguito dalla parola chiave `pezzi`. Segue l'elenco dei pezzi ordinati per numero crescente di *slot* compatibili. In caso di parità, si sommano i tempi di taglio  $t_{pm}$  associati agli *slot* compatibili, ottenendo il tempo totale  $\tau_p = \sum_{(m,g)} t_{pm}$  e si ordinano i pezzi per tempo totale decrescente. In caso di ulteriore parità, si considerano i pezzi per indice crescente. Ogni pezzo occupa una riga, che riporta l'indice del pezzo, il numero di *slot* compatibili e il tempo totale separati da uno spazio. La logica dell'ordinamento è di considerare per primi i pezzi più critici da produrre, cioè compatibili con pochi *slot* e lunghi da tagliare. Per esempio:

```
10 pezzi
9 1 579
8 1 252
4 1 122
2 1 109
10 1 92
1 1 76
5 2 747
6 2 548
3 3 1069
7 3 843
```

Quindi, il programma stampa il numero di terne  $(p, m, g)$  compatibili, seguito dalla parola chiave `compatibilita'` e il loro elenco, ordinato per tempo di taglio  $t_{pm}$  crescente. Siccome tutte le terne con gli stessi  $p$  e  $m$  hanno lo stesso tempo per definizione, e altre terne possono avere tempi identici per caso, in caso di parità le terne vengono ordinate lessicograficamente, cioè per indici di pezzo, macchina e giorno crescenti. Ogni terna occupa una riga, che riporta gli indici interi  $p$ ,  $m$  e  $g$  e il tempo  $t_{pm}$  separati da spazi. Per esempio:

```
16 compatibilita'
1 1 1 76
10 1 3 92
2 2 2 109
4 2 2 122
7 1 1 215
6 1 3 223
8 1 1 252
3 1 1 273
5 1 3 304
7 2 1 314
7 2 2 314
6 2 2 325
3 2 1 398
3 2 2 398
5 2 2 443
9 2 2 579
```

A questo punto, il programma costruisce un piano di produzione assegnando i pezzi agli slot con la seguente procedura euristica. Ad ogni passo considera il pezzo più critico in base alla definizione discussa sopra e lo assegna allo *slot* compatibile che ha il massimo tempo di attività residuo, purché sufficiente a tagliare il pezzo. Il tempo residuo è ovviamente il tempo di attività non ancora impiegato per tagliare pezzi già assegnati: ogni volta che un pezzo  $p$  viene assegnato a uno *slot*  $(m, g)$ , il tempo residuo di quest'ultimo cala del tempo di taglio  $t_{pm}$ . Se nessuno *slot* ha tempo residuo sufficiente a tagliare il pezzo, questo viene scartato.

Al termine, il programma stampa la soluzione ottenuta, nel seguente formato:

- la prima riga riporta il numero dei pezzi scartati, seguito dalla parola chiave **pezzi scartati**;
- ciascuna riga seguente si riferisce a uno *slot* in ordine lessicografico (prima per indice di macchina crescente, poi per indice di giorno crescente) e riporta l'indice della macchina, l'indice del giorno, il tempo residuo espresso in secondi e l'elenco dei pezzi assegnati allo slot, ordinati per indici crescenti; tutti questi dati sono separati da spazi.

Per esempio:

```
5 pezzi scartati
1 1 104 1
1 3 104 5 10
2 1 2 3
2 2 78 4
```

Infine, il programma genera un piano di produzione alternativo come segue. Ad ogni passo considera lo *slot* con il massimo tempo di attività  $d_{mg}$  e costruisce un problema ausiliario di zaino, nel quale gli oggetti corrispondono ai pezzi compatibili con lo slot, i valori sono tutti unitari, i volumi pari ai tempi di taglio  $t_{pm}$  sulla macchina associata allo *slot* e la capacità dello zaino è il tempo di attività dello *slot* stesso. Risolto il problema all'ottimo, gli oggetti inseriti nello zaino individuano i pezzi da assegnare allo *slot*<sup>2</sup>. L'algoritmo procede quindi con lo *slot* successivo e i pezzi ancora non assegnati.

Al termine, il programma stampa la soluzione ottenuta, nel formato sopra descritto. Per esempio:

```
5 pezzi scartati
1 1 104 1
1 3 185 6 10
2 1 86 7
2 2 78 4
```

## Chiarimenti

In questa sezione saranno riportate le risposte a domande e dubbi.

---

<sup>2</sup>Come ogni problema di ottimizzazione, il problema dello zaino può ammettere molteplici soluzioni equivalenti. Le soluzioni di esempio sono state ottenute risolvendo il problema come in laboratorio. Soluzioni alternative sono accettabili purché corrette, cioè ottime per il problema dello zaino.