Algoritmi e Strutture di Dati (3^a Ed.)
Matroidi

Alan Bertossi, Alberto Montresor

Moltissimi problemi (ma non tutti) per i quali un algoritmo *greedy* permette di trovare la soluzione ottima possono essere formulati come problemi su una struttura matematica chiamata *matroide*. La teoria dei matroidi è stata introdotta da Whitney (1935) per studiare le proprietà algebriche delle dipendenze lineari, e viene richiamata molto brevemente in questo paragrafo. Una trattazione completa si può trovare nel testo di Lawler (1976).

Un matroide è una coppia $M = (S, F)$, dove S è un insieme finito di elementi ed F è una famiglia di sottoinsiemi di S, chiamati *insiemi indipendenti* di S, che soddisfa le proprietà seguenti:

(1) *ereditarietà*: l'insieme vuoto appartiene ad F e tutti i sottoinsiemi propri di un insieme di F appartengono anch'essi ad F, ovvero: $\emptyset \in F$, e se $I \in F$ e $J \subseteq I$, allora $J \in F$;

(2) *scambio*: se due insiemi di F non hanno la stessa cardinalità, allora un elemento che sta soltanto nel sottoinsieme più grande può essere aggiunto all'insieme più piccolo, e l'insieme risultante appartiene ancora ad F, ovvero: se $I \in F$, $J \in F$, e $|I| < |J|$, allora esiste $x \in J - I$ tale che $I \cup \{x\} \in F$.

M è un matroide di una matrice A se S corrisponde all'insieme delle colonne di A, ed un insieme I di colonne appartiene ad F se le colonne in I sono linearmente indipendenti. M è un matroide di un grafo non orientato $G = (V, E)$ se S corrisponde all'insieme E degli archi, mentre un insieme I di archi sta in F se e solo se gli archi in I non formano un circuito. In altri termini, ciascun insieme indipendente di un matroide di un grafo forma una foresta. Il matroide di un grafo G è equivalente al matroide della matrice A di incidenza nodi-archi di G, dove gli elementi 0 e 1 della matrice A sono considerati nel campo degli interi modulo 2.

Sui matroidi sono state dimostrate moltissime proprietà, delle quali ne enunciamo nel seguito solo alcune e senza dimostrazione.

Dati un matroide M ed un insieme $I \in F$, un elemento $x \notin I$ è un'estensione di I se può essere aggiunto ad I mantenendone l'indipendenza, cioè se $I \cup \{x\} \in F$. Per esempio, se M è un matroide di un grafo, allora un'estensione è un arco che può essere aggiunto ad un insieme di archi senza introdurre un circuito. Un insieme I per il quale non esiste alcuna estensione è detto *massimale*. Una proprietà molto utile dei matroidi, che si può facilmente dimostrare per assurdo, è che tutti i suoi insiemi indipendenti massimali hanno la stessa cardinalità.

Un matroide è pesato se c'è un peso positivo associato ad ogni elemento $x \in S$. Il peso di un insieme $I \in F$ è dato dalla somma dei pesi degli elementi che appartengono ad I. Molti problemi per i quali la soluzione ottima può essere trovata con un algoritmo *greedy* possono essere formulati come problemi di trovare un insieme indipendente massimale di peso massimo in un matroide pesato. Per esempio, il problema di trovare il minimo albero di copertura di un grafo G è equivalente al problema di trovare un insieme indipendente massimale di peso massimo nel matroide M del grafo G. Infatti, se c_a è il peso dell'arco a nel grafo G, allora il peso p_a dello stesso arco nel matroide M è definito come $p_a = C - c_a$, dove C è un numero più grande del massimo peso degli archi di G. In questo modo, i pesi di M sono positivi e trovare l'insieme indipendente massimale di peso massimo in M corrisponde a trovare il minimo albero di copertura di G. Infatti, $p(I) = \sum_{a \in I} p_a$ è il peso di un insieme indipendente I di M, mentre $c(I) = \sum_{a \in I} c_a$ è quello di un albero di copertura di G formato dagli stessi archi. Per costruzione, $p(I) = (n-1)C - c(I)$, dove n è il numero di nodi di G, e quindi massimizzare $p(I)$ è equivalente a minimizzare $c(I)$.

La caratteristica più importante dal punto di vista algoritmico dei matroidi pesati è che il metodo *greedy* verifica le proprietà della scelta *greedy* e della sottostruttura ottima e che pertanto trova sempre l'insieme indipendente massimale di peso massimo! Riformulato su un matroide pesato $M = (S, F)$ con funzione peso p (positiva), l'algoritmo *greedy* è espresso dalla procedura greedy-matroid().

```
SET greedy-matroid(matroide M = (S, F) con funzione peso p)
    SET I ← ∅
    { ordina S per peso decrescente }
    foreach a ∈ S do                         % Secondo l'ordinamento appena calcolato
        if I ∪ {a} ∈ F then
            I ← I ∪ {a}
    return I
```

La complessità dell'algoritmo dipende essenzialmente dal tempo richiesto per verificare che $I \cup \{x\}$ sia indipendente. Se tale tempo è $O(f(n))$, dove n è la cardinalità di S, allora la complessità risultante è $O(n \log n + nf(n))$.