Indice argomenti di Algoritmi avanzati

Vincenzo Manzoni

25 gennaio 2007

1. Grafi
   (a) $G = (N, A)$
   (b) Pesi associati agli archi
      i. c_{ij}: costo di attraversamento dell'arco (i, j)
      ii. l_{ij}: capacità inferiore dell'arco
      iii. u_{ij}: capacità superiore dell'arco
   (c) Pesi b_i associati a nodi: *deficit*
      i. $b_i > 0$: nodo di destinazione, di output, pozzo. b_i è detta *domanda* del nodo i.
      ii. $b_i < 0$: nodo di origine, di input, sorgente. b_i è detta *offerta* del nodo b_i.
      iii. $b_i = 0$: nodo di trasferimento
   (d) Insieme D dei nodi di domanda
   (e) Insieme O dei nodi di offerta
   (f) $\sum_{i \in D} b_i$: domanda globale
   (g) $-\sum_{i \in O} b_i$: offerta globale
   (h) Nelle reti di flusso, vale che $\sum_{i \in N} b_i = 0$
   (i) x_{ij}: flusso sull'arco (i, j) se soddisfa i vincoli di conservazione di flusso
      i. Per i nodi di input
      ii. Per i nodi di output
      iii. Per i nodi di trasferimento
   (j) Codifica di un grafo: la matrice di incidenza
2. Problema del *Flusso di Costo Minimo* (MCF)
   (a) Espressione come PL
   (b) (Esempi di problemi modellabili come MCF)
   (c) Forma standard del problema MCF (5 condizioni)
      i. Come ricondursi alla forma standard
3. Visita di un grafo
   (a) Condizioni per dire che un grafo è un albero (4)
   (b) Procedura di visita (algoritmo)
   (c) Implementazioni di Q :
      i. Fila: *breadth-first*; teorema.
      ii. Pila: *depth-first*
4. Cammini ottimi

(a) (Esempi di applicazione)
(b) Cammino minimo dal nodo r al nodo t
(c) Cammini minimi dal nodo r a tutti gli altri nodi $i \neq r$
(d) Formulazione come MCF
(e) Ipotesi di lavoro
    i. almeno un cammino dal nodo r ad ogni nodo $i \neq r$
    ii. il grafo non ha cicli negativi
(f) Albero di copertura

5. Problema dell'albero dei cammini minimi (SPT, *Shortest Path Tree*)
(a) Condizioni di ottimalità
(b) Condizioni di Bellman
(c) Algoritmo Naive-SPT
(d) Algoritmo SPT
(e) Teorema "delle etichette"
(f) Teorema del "numero finito di passi"
(g) Implementazione di Q come coda di priorità: estrazione in base al valore della chiave
    i. SPT.S (*Shortest-first*)
        A. Algoritmo di Dijkstra: Q come lista non ordinata. Complessità: $O(n^2)$
(h) Implementazione di Q come lista: estrazione in base alla posizione dell'elemento nella lista
    i. SPT.L (*List*)
        A. la lista Q è un fila: Algoritmo di Bellman. Complessità: $O(mn)$
        B. la lista Q è una pila
        C. (la lista Q è una lista a doppio ingresso)
(i) Teorema SPT.L, dove Q è fila, $n-1$.
    i. Verifica della presenza di cicli negativi
(j) Cammini minimi su grafi aciclici
    i. Definizione di grafo aciclico
    ii. Algoritmo SPT.Acyclic. Complessità: $O(m)$
(k) Cammini minimi con radici multiple

6. Albero di copertura di costo minimo (MST, *Minimal Spanning Tree*)
(a) Algoritmo di Prim
(b) Algorimo di Kruskal
(c) Algoritmo greedy per MST

7. Problema di flusso massimo (MF, *Max Flow*)
(a) Formulazione in PL
(b) Tagli, cammini aumentanti, condizioni di ottimo
(c) Algoritmo per cammini aumentanti. Complessità: $O(mnU)$
(d) Algoritmo basato sui preflussi. Complessità: $O(n^2m)$ oppure $O(n^3)$
(e) Flusso massimo con più sorgenti/pozzi

8. Problema di flusso di costo minimo (MCF, *Min Cost Flow*)

(a) Cammini, cicli aumentanti, condizioni di ottimo

(b) Algoritmo basato sui cammini minimi successivi. Complessità: $O(\bar{g}mn)$

(c) Algoritmo basato sulla cancellazione di cicli. Complessità: $O(nm^2\bar{u}\bar{c})$

9. Problemi di accoppiamento

   (a) Accoppiamento di massima cardinalità. Complessità: $O(mn)$

   (b) Accoppiamento di costo minimo. Complessità: $O(mn^2)$

   (c) Accoppiamento di massima cardinalità bottleneck

10. Il problema dello zaino (KP, *Knapsack Problem*)

    (a) Formulazione PLI

        i. In forma di minimo

        ii. In forma di massimo

    (b) Formulazione come problema di ottimizzazione combinatoria

11. Il problema del commesso viaggiatore (TSP, *Traveling Salesman Problem*)

    (a) Formulazione PLI

    (b) Formulazione come problema di ottimizzazione combinatoria

12. Il problema dell'ordinamento di lavori su macchine con minimizzazione del numero di macchine (MCMS, *Minimal Cardinality Machine Scheduling*)

    (a) Formulazione PLI

13. Il problema dell'ordinamento di lavori su macchine con minimizzazione del tempo di completamento (MMMS, *Minimal Makespan Machine Scheduling*)

    (a) Formulazione PLI

14. Il problema di colorazione del grafo (GC, *Graph Colouring*)

    (a) Formulazione PLI

    (b) Analogie con MCMS

15. Problemi di selezione da sottoinsiemi

    (a) Problemi di copertura

        i. Formulazione PLI

    (b) Problemi di partizione

        i. Formulazione PLI

    (c) Problemi di riempimento

        i. Formulazione PLI

16. Il problema *Weighted Vertex Cover* (WVC)

    (a) Formulazione PLI

17. Il problema di dislocazione ottima di impianti

    (a) Formulazione PLI

18. Algoritmi greedy

(a) Schema generale

(b) Algoritmo greedy per MST: (Kruskal)

(c) Algoritmo greedy per KP

   i. 3 criteri di ordinamento degli elementi

(d) Algoritmo greedy per assegnamento di costo minimo

(e) Algoritmo greedy per TSP

(f) Algoritmo greedy per MCMS

(g) Algoritmi greedy per MMMS (algoritmi *list scheduling*)

(h) Algoritmi greedy per PC

19. Algoritmi greedy e garanzia sulle prestazioni

    (a) Valutazione del'errore compiuto dall'algoritmo greedy per KP (valori unitari non crescenti)

    (b) Valutazione dell'errore compiuto dagli algoritmi di list scheduling (SPT e LPT) per MMMS

    (c) Valutazione dell'errore compiuto dall'algoritmo twice around MST per TSP

    (d) Costruzione di un algoritmo greedy per WVC che determini una soluzione ammissibile e ci permetta di valutarne la bontà

20. Algoritmi di ricerca locale

    (a) Schema dell'algoritmo

    (b) Funzioni intorno

    (c) Algoritmo di ricerca locale per MMMS

    (d) Algoritmo di ricerca locale per problemi di dislocazione ottima di impianti

    (e) Strategie metaeuristiche

        i. Multistart

        ii. Simulated annealling

            A. Schema dell'algoritmo

        iii. Ricerca taboo

        iv. Algoritmi genetici