ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

PLANNING OPTIMIZATION

Approximation algorithms

ONE LOVE. ONE FUTURE.

CONTENT

- Overview
- Knapsack problem
- TSP problem

Overview

- Algorithms with polynomial time complexity
- Find approximate solutions to optimization problems: the distances of the solutions found to optimal solutions can be probably guaranteed.
- Notation
  - Optimal objective value is f^*
  - Objective value of the solution found by the approximation algorithm is f
- Maximization problems
  - We can prove that: $f \geq \alpha f^*$ in which α is a positive constant less than 1
- Minimization problems
  - We can prove that: $f \leq \alpha f^*$ in which α is a positive constant greater than 1

KNAPSACK

- There are n items $A = \{1, 2, . . ., n\}$ in which item i has weight W_i and value C_i ($i = 1, 2, ..., n$). Find a subset of A such that the sum of weights of items is less than or equal to B and the sum of values is maximal.
- Let S be a solution to the problem (S is a subset of A). Let $f(S) = \sum_{i \in S} C_i$ be the value of the solution S
- Let S^* be an optimal solution and $f^* = f(S^*)$ be the optimal objective value
- Consider 2 algorithms running on 2 sorted list of items:
  - Sort in non-increasing order of values
  - Sort in non-increasing order of the fractions of values over the weights

KNAPSACK

```
GREEDY-1 () {
    ({C_1, W_1}, {C_2, W_2}, . . ., {C_n, W_n}) is the sorted list in which: C_1 ≥ C_2 ≥ . . . ≥ C_n;
    S = {};
    for i = 1 to n do {
        if W_i > B then break;
        S = S ∪ {i}; B = B – W_i;
    }
    return S;
}
```

```
GREEDY-2 () {
    ({C1, W1}, {C2, W2}, . . ., {Cn, Wn}) is the sorted list in which: C1/W1 ≥ C2/W2 ≥ . . . ≥ Cn/Wn;
    S = {};
    for i = 1 to n do {
        if Wi > B then break;
        S = S ∪ {i}; B = B – Wi;
    }
    return S;
}
```

GREEDY-2 () {

$(\{C_1, W_1\}, \{C_2, W_2\}, \ldots, \{C_n, W_n\})$ is the sorted list in which: $\frac{C_1}{W_1} \geq \frac{C_2}{W_2} \geq \ldots \geq \frac{C_n}{W_n}$;

$S = \{\}$;

for i = 1 to n do {

if $W_i > B$ then break;

$S = S \cup \{i\}$; $B = B - W_i$;

}

return S;

}

KNAPSACK

- There are n items $A = \{1, 2, . . ., n\}$ in which item i has weight W_i and value C_i (i = 1, 2, ..., n). Find a subset of A such that the sum of weights of items is less than or equal to B and the sum of values is maximal.
- **Approximation algorithms**
  - Let S_1 and S_2 be the solutions returned by GREEDY-1 and GREEDY-2
  - Notation: $f = \max\{f(S_1), f(S_2)\}$
  - We have $f \geq \frac{1}{2}f^*$

KNAPSACK

- Proof
- Consider the linear program below:

$$f(X) = C_1X_1 + C_2X_2 + \ldots + C_nX_n \rightarrow \max$$

$$\text{s.t.} \quad W_1X_1 + W_2X_2 + \ldots + W_nX_n \leq B$$

$$0 \leq X_1, X_2, \ldots, X_n \leq 1$$

- Simplex table

X_1	X_2	. . .	X_n	X_{n+1}	X_{n+2}	. . .	X_{2n+1}	Z	RHS
W_1	W_2	. . .	W_n	1	0	. . .	0	0	B
1	0		0	0	1		0	0	1
0	1	. . .	0	0	0	. . .	0	0	1
…	…	…	…	…	…	…	…	…	…
0	0		1	0	0		1	0	1
$-C_1$	$-C_2$	. . .	$-C_n$	0	0	0	0	1	0

KNAPSACK

	X_1	X_2	. . .	X_n	X_{n+1}	X_{n+2}	. . .	X_{2n+1}	Z	RHS	E
R_1	W_1	W_2	. . .	W_n	1	0	. . .	0	0	B	B/W_1
R_2	1	0		0	0	1		0	0	1	1
R_3	0	1	. . .	0	0	0	. . .	0	0	1	∞
	…	…	…	…	…	…	…	…	…	…	
R_{n+1}	0	0		1	0	0		1	0	1	∞
R_{n+2}	$-C_1$	$-C_2$	. . .	$-C_n$	0	0	0	0	1	0	

Select column 1, suppose $B \geq W_1$ → select row R_2: $R_1 = R_1 - W_1R_2$; $R_{n+2} = R_{n+2} + C_1R_2$

	X_1	X_2	. . .	X_n	X_{n+1}	X_{n+2}	. . .	X_{2n+1}	Z	RHS	E
R_1	0	W_2	. . .	W_n	1	$-W_1$	. . .	0	0	$B-W_1$	
R_2	1	0		0	0	1		0	0	1	
R_3	0	1	. . .	0	0	0	. . .	0	0	1	
	…	…	…	…	…	…	…	…	…	…	
R_{n+1}	0	0		1	0	0		1	0	1	
R_{n+2}	0	$-C_2$	. . .	$-C_n$	0	C_1	0	0	1	C_1	

KNAPSACK

	X_1	X_2	. . .	X_n	X_{n+1}	X_{n+2}	. . .	X_{2n+1}	Z	RHS	E
R_1	0	W_2	. . .	W_n	1	$-W_1$	. . .	0	0	$B-W_1$	$(B-W_1)/W_2$
R_2	1	0		0	0	1		0	0	1	∞
R_3	0	1	. . .	0	0	0	. . .	0	0	1	1
	…	…	…	…	…	…	…	…	…	…	. . .
R_{n+1}	0	0		1	0	0		1	0	1	∞
R_{n+2}	0	$-C_2$	. . .	$-C_n$	0	0	0	0	1	C_1	

Select column 2, Suppose $(B–W_1) \geq W_2$ → select row R_3: $R_1 = R_1 – W_2R_3$; $R_{n+2} = R_{n+2} + C_2R_3$

	X_1	X_2	. . .	X_n	X_{n+1}	X_{n+2}	. . .	X_{2n+1}	Z	RHS	E
R_1	0	0	. . .	W_n	1	$-W_1$	. . .	0	0	$B - W_1 - W_2$	
R_2	1	0		0	0	1		0	0	1	
R_3	0	1	. . .	0	0	0	. . .	0	0	1	
	…	…	…	…	…	…	…	…	…	…	
R_{n+1}	0	0		1	0	0		1	0	1	
R_{n+2}	0	0	. . .	$-C_n$	0	C_1	C_2	0	1	$C_1 + C_2$	

KNAPSACK

	X_1	X_2	X_3	. . .	X_n	X_{n+1}	X_{n+2}	. . .	X_{2n+1}	Z	RHS	E
R_1	0	0	W3	. . .	W_n	1	$-W_1$	. . .	0	0	$B - W_1 - W_2$	$(B - W_1 - W_2) / W_3$
R_2	1	0	0	. . .	0	0	1		0	0	1	∞
R_3	0	1	0	. . .	0	0	0	. . .	0	0	1	∞
R_4	0	0	1	. . .	0	0	0	…	…	…	1	1
R_{n+1}	0	0	0	. . .	1	0	0		1	…	…	∞
R_{n+2}	0	0	$-C_3$	. . .	$-C_n$	0	C_1	C_2	0	1	$C_1 + C_2$	

Select column 3, suppose $(B - W_1 - W_2) < W_3$ → select row R_1: $R_1 = R_1/W_3$; $R_{n+2} = R_{n+2} + C_3R_1/W_3$

	X_1	X_2	X_3	X4	…	X_n	X_{n+1}	X_{n+2}	. . .	X_{2n+1}	Z	RHS
R_1	0	0	1	W_4/W_3	…	W_n/W_3	1	$-W_1$	. . .	0	0	$(B-W_1 - W_2)/W_3$
R_2	1	0	0	0	…	0	0	1		0	0	1
R_3	0	1	0	0	…	0	0	0	. . .	0	0	1
R_4	0	0	0	$-W_4/W_3$	…	$-W_n/W_3$	0	0	…	…	…	…
R_{n+1}	..	..	..	. . .	…	1	0	0		1	0	1
R_{n+2}	0	0	0	$-C_4+W_4*C_3/W_3$	…	$-C_n+W_n*C_3/W_3$	0	C_1	C_2	0	1	$C_1 + C_2$

KNAPSACK

	X_1	X_2	X_3	X4	…	X_n	X_{n+1}	X_{n+2}	. . .	X_{2n+1}	Z	RHS
R_1	0	0	1	W_4/W_3	…	W_n/W_3	1	$-W_1$	. . .	0	0	$(B-W_1-W_2)/W_3$
R_2	1	0	0	0	…	0	0	1		0	0	1
R_3	0	1	0	0	…	0	0	0	. . .	0	0	1
R_4	0	0	0	$-W_4/W_3$	…	$-W_n/W_3$	0	0	…	…	…	…
R_{n+1}	..	..	..	. . .	…	1	0	0		1	0	1
R_{n+2}	0	0	0	$-C_4+W_4*C_3/W_3$	…	$-C_n+W_n*C_3/W_3$	0	C_1	C_2	0	1	$C_1 + C_2$

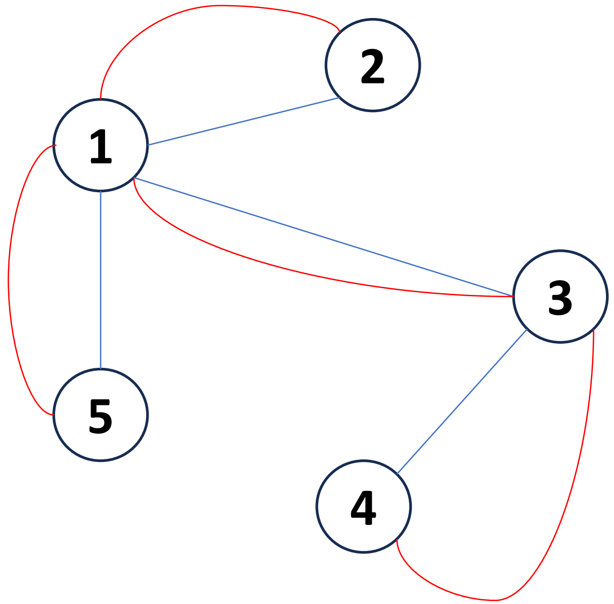
- Values of the last row are non-negative, the table corresponds to an optimal solution: $X_1 = 1$, $X_2 = 1$, $X_3 = (B-W_1-W_2)/W_3$, $X_4 = \ldots = X_n = 0$.
- Generality: select column from left to right (from columns 1, 2, 3, . . .). Suppose k is the first index (iteration) where $B-W_1-W_2-\ldots-W_{k-1} < W_k$. An optimal solution to the linear problem is: $X_1 = X_2 = \ldots = X_{k-1} = 1$., $X_k = (B-W_1-W_2-\ldots-W_{k-1})/W_k$, $X_{k+1} = \ldots = X_n = 0$ optimal objective value is $C_1 + C_2 + \ldots + \alpha C_k$, với $\alpha = (B-W_1-W_2-\ldots-W_{k-1})/W_k$, ($\alpha < 1$)

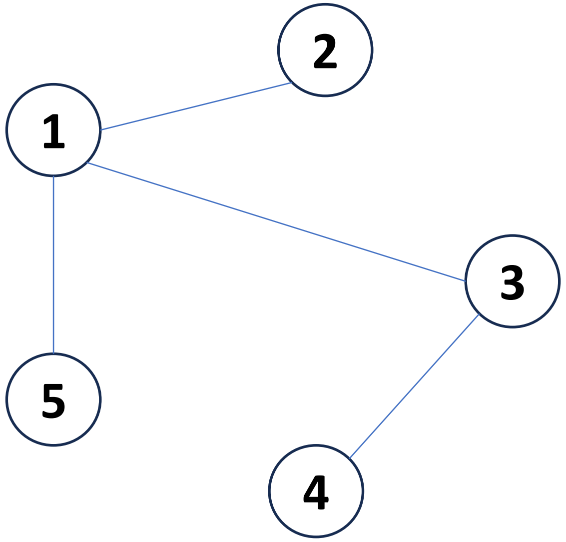
BÀI TOÁN KNAPSACK

- The given linear program is a relaxation of the original Knapsack problem: optimal objective value of the Knapsack problem is $f^* \leq C_1 + C_2 + \ldots + \alpha C_k$. Objective value of the solution returned by the approximation is $f(S_1) = C_1 + C_2 + \ldots + C_{k-1}$.
- We have $f^* \leq C_1 + C_2 + \ldots + \alpha C_k < C_1 + C_2 + \ldots + C_{k-1} + C_k = f(S_1) + C_k$. Therefore, $f(S_1) > f^*/2$ or $f(S_2) \geq C_k > f^*/2$.
- Hence $f = \max\{f(S_1), f(S_2)\} > f^*/2$.

TRAVELLING SALESMAN PROBLEM - TSP

- Let $G = (V, E)$ be a complete graph in which $V = \{1, 2, \ldots, n\}$ is the set of nodes. Edge (u,v) has weight $c(u,v)$. Find the Hamilton cycle G such that the total weights is minimal.
- Approximation algorithm:
  - Let $T(G)$ be a minimum spanning tree of G.
  - Duplicate the edges of T → we obtain an Euler graph with the Euler cycle $E(T)$.
  - Let S^0 be the sequence (pairwise distinct) of nodes visited when travelling along E(T). The addition of the first node of S^0 to the end of S^0 yields a Hamilton cycle S which is the solution of the approximation algorithm.
  - Example
    - $E(T)$ = 1, 2, 1, 5, 1, 3, 4, 3, 1
    - S^0 = 1, 2, 5, 3, 4
    - S = 1, 2, 5, 3, 4, 1





HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !