An Introduction to Approximation Algorithm

Agenda

1. Brief Introduction
2. Set Cover Problem
3. Travel Salesman

Presenter : Ming-Yu Liu

Date : 2015 / 7 / 20

Approximation

Intro

[Why We Need It]

For NP Hard Problems

$$C^{\{\ \text{Fast},\ \text{Cheap},\ \text{Reliable}\ \}}_{2}$$

Solutions

α – approximation algorithm

$\alpha > 1$ for min problem, < 1 for max

Linear Programming

Problem

[Weighted Set Cover]

Universe	$U = \{ u_1 , u_2 \dots u_m \}$
Subsets / Costs	$S_1 , S_2 \dots S_n \subseteq U$; $c_1 , c_2 \dots c_n$
Objective	**Min** $\sum_{j \in I} c_j$; $I \subseteq \{1 , 2 \dots n\}$ **S.T.** $\bigcup_{j \in I} S_j = U$

Linear Programming

Formula

[Integer Programming]

Min $\sum_{j=1}^{m} c_j x_j$

S.T. $\sum_{j\,:\,u_i \in S_j} x_j \geq 1 \quad , \quad i = 1, \dots, n$

$x_j \in \{0, 1\} \quad , \quad j = 1, \dots, m$

Where x_j = Subset S_j is chosen or not

n = Number of Constraints

m = Number of Subsets

Linear Programming

Example

[Universe] = { 1, 2, 3, 4, 5 }

S_1= { 1, 5 } S_3= { 2, 3, 4 }

S_2= { 1, 3 } S_4= { 2, 4, 5 }

x_j for S_j $= \begin{cases} 1 \text{ if select } S_j \\ 0 \text{ if not} \end{cases}$

S . T .

$$\begin{aligned} x_1 &+ x_2 \geq 1 \\ x_3 &+ x_4 \geq 1 \\ x_2 &+ x_3 \geq 1 \\ x_1 &+ x_4 \geq 1 \end{aligned}$$

$x_j \in \{0, 1\}$， $j = 1, \dots, m$

Linear Programming

Relaxation

[Integer Programming]

Min $\sum_{j=1}^{m} c_j x_j$

S.T. $\sum_{j\,:\,u_i \in S_j} x_j \geq 1 \quad , \quad i = 1, \dots, n$

$\boxed{x_j \in \{0, 1\}} \quad , \quad j = 1, \dots, m$

Where x_j = Subset S_j is chosen or not

n = Number of Constraints

m = Number of Subsets

Linear Programming

Relaxation

[Linear Programming]

Min $\sum_{j=1}^{m} c_j x_j$

S.T. $\sum_{j\,:\,u_i \in S_j} x_j \geq 1 \quad , \quad i = 1, \dots, n$

$x_j \geq 0 \quad , \quad j = 1, \dots, m$

Where x_j = Subset S_j is chosen or not

n = Number of Constraints

m = Number of Subsets

Deterministic Rounding

Definition

[LP Solution] = x_j^*

IP Solution x_j^* for $S_j = \begin{cases} 1 \text{ if } x_j^* \geq 1 / f \\ 0 \text{ if not} \end{cases}$

Where $f = max_{i=1,\dots,n} f_i$

$f_i = \left| \{ j : u_i \in S_j \} \right|$

Deterministic Rounding

Theorem

[Optimal Solution Upper Bound]

Claim It's a f – approximation for set cover

Where $Z^*_{LP} \leq Z^*_{IP} = \sum_{j=1}^{m} c_j x_j$

Z^* = optimal solution for the linear and integer problem respectively

Proof It's a f – approximation for set cover

$$Z_{LP}^{*} \le Z_{IP}^{*} = \sum_{j=1}^{m} c_j x_j \;\; ; \;\; x_j^{*} \text{ for } S_j = \begin{cases} 1 \text{ if } x_j^{*} \ge 1 / f \\ 0 \text{ if not} \end{cases}$$

Step 1 : Transposition

$$f \,.\, x_j^{*} \ge 1$$

Step 2 : Plug in answer to the set cover problem

$$\sum_{j=1}^{m} c_j \,.\, (f \,.\, x_j^{*}) = f \sum_{j=1}^{m} c_j x_j^{*}$$

$$f \,.\, Z_{LP}^{*} \le f \,.\, Z_{IP}^{*} = f \sum_{j=1}^{m} c_j x_j$$

Greedy Algorithm

Definition

[Plain English]

“ Making a series of decisions to get to a solution, while the decisions at each step are made by choosing the one that leads to the most obvious benefits ”

Advantage	Disadvantage
● Easy to implement	● No solution promise

Greedy Algorithm

Problem

[Unweighted Set Cover]

Universe	$U = \{ u_1 , u_2 \ldots u_m \}$
Subsets	$S_1 , S_2 \ldots S_n \subseteq U$
Objective	Min $\lvert I \rvert$; $I \subseteq \{1 , 2 \ldots n\}$ S . T . $\bigcup_{j \in I} S_j = U$

Greedy Algorithm

Demo

[Set Cover Example]

Universe	$U = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$
Subsets	$S_1 = \{ 0, 1, 2, 7, 8, 9 \}$ $S_2 = \{ 0, 1, 2, 3, 4 \}$ $S_3 = \{ 3, 4, 6 \}$ $S_4 = \{ 5, 6, 7, 8, 9 \}$ $S_5 = \{ 4, 5, 6 \}$
Solution	$S_2 \cup S_4 = U$; $\lvert I \rvert = 2$

Greedy Algorithm

Theorem

[Optimal Solution Upper Bound]

Claim Set Cover with at most $m\log_e n$ sets

Where m = Optimal solution's size

n = # of universe's points

Proof Set Cover with at most $m\log_e n$ sets

n = # of universe's points ; m = Optimal solution's size

Iteration 1 : $n_1 =$ elements left

$$n_1 \le n - n / m = n(1 - 1 / m)$$

Iteration 2 :

$$n_2 \le n_1(\ 1 - 1/(m-1)\ )\ \le n_1(\ 1 - 1/m\ )$$
$$\le n\ (\ 1 - 1/m\ )^2$$

Iteration k :

$$n_k \le n\ (\ 1 - 1/m\ )^k < 1$$

Proof Set Cover with at most $m log_e n$ sets

n = # of universe's points ; m = Optimal solution's size

Step 1 : Transposition

$$n\ (1-1/m)^{k} < 1 \Rightarrow (1-1/m)^{m\frac{k}{m}} < 1/n$$

Step 2 : Given $(1-x)^{1/x} \approx 1/e$

$$e^{-\frac{k}{m}} < 1/n$$

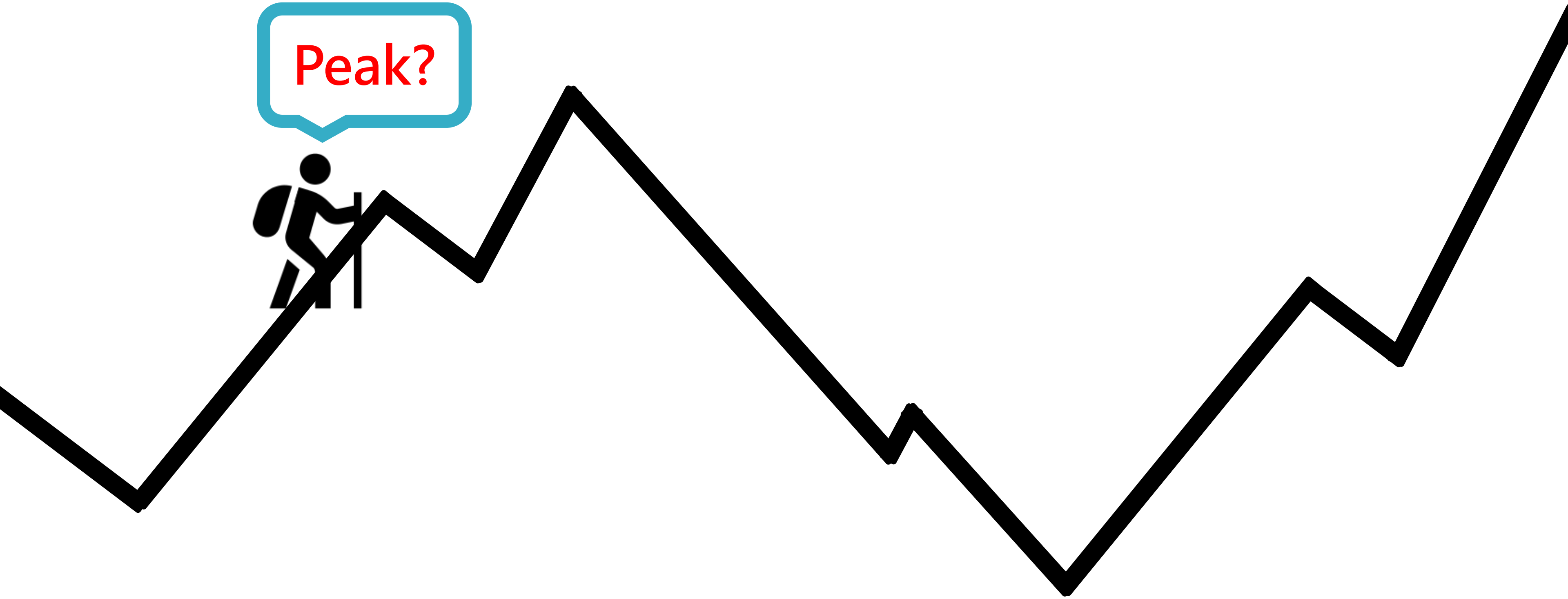
Step 3 : Take log for both side ;
Transposition

$$k/_m > log_e n \Rightarrow k > m log_e n$$

Local Search

Definition

[Algorithm Metaphor]



Local Search

Algorithm

[Random Hill Climbing]

Initial Search	Random
Select Move	Random
Acceptable Move	Not Worsening
Stop Search	Idle Iteration

Local Search

Problem

[Travel Salesman]

City Order	$\boldsymbol{k}(1), \boldsymbol{k}(2), \dots, \boldsymbol{k}(\boldsymbol{n})$
Cost Matrix	Symmetric $\boldsymbol{C} = (\boldsymbol{c}_{ij})$
Objective	Min $\sum_{i=1}^{n-1} c_{k(i)k(i+1)} + c_{k(n)k(1)}$