CSC2421 Lecture 1: Greedy Algorithms 7 December 2005
Mutaamba Maasha

---

1 Course

Instructor:	Allan Borodin
Office:	SF 2303b
Telephone:	416.978.6416
Email :	bor@cs.utoronto.edu
Lectures :	Thursday 1-3 KP 213
Website :	http://www.cs.toronto.edu/ bor/2421s05/

2 Overview

The focus of this course is approximation algorithms. Approximation algorithms are usually employed to obtain better complexity bounds for given problems. Algorithms come in different sizes, shapes and textures but many strategies employed by algorithms can be classified in following categories : greedy, local search, divide and conquer, dynamic programming, backtracking (branch and bound), LP relaxation and rounding, LP primal-dual, local ratio, and network flows. We will consider approximation algorithms that use these various techniques.

Although approximations can be made for problems with known polynomial bounded solutions, NP-complete problems are the most attractive candidates for approximation due to their intractability. For much of the course approximation algorithms for various representatives of NP optimization problems will be examined. We will look at approximation algorithms and consider the approximations worst case bounds. There will be more emphasis placed on deterministic algorithms than randomized algorithms.

3 Greedy Algorithms

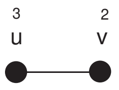
In this lecture we will focus on greedy approximations to NP-complete problems.

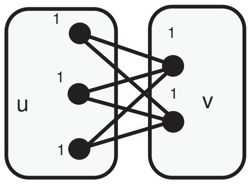
3.1 Vertex Cover

Given an undirected graph $G = (V, E)$, a weight function $w : V \to \mathbb{Q}^+$. The objective function is defined for $V' \subseteq V$ as $w(V') = \sum_{v \in V'} w(v)$. A set $V' \subseteq V$ is a vertex cover if and only if $\forall (u, v) \in E : v \in V' \vee u \in V'$. The *vertex cover problem* is to find a vertex cover V' that minimizes $w(V')$. The unweighted analog is the case where $w : V \to \{1\}$ and is is known as the *cardinality vertex cover problem.*

3.1.1 Generalizations

A weighted is set cover problem can be expressed as an unweighted set cover problem in polynomial time. First scale the weights to get integer weights. For each node $u \in V$ create $w(u)$ nodes in V. Then for each edge $(u, v) \in E$ create a complete bipartite graph using the set of $w(u)$ and $w(v)$ nodes. The case for a sample edge (u, v) is illustrated below.





3.1.2 Approximation : Greedy via Maximal Matching

There is a simple greedy algorithm for approximating a unweighted vertex cover case using maximal matching to find a vertex cover. Given a graph $G = (V, E)$, a subset of edges $M \subseteq E$ is a matching if no two edges of M share an endpoint.

```
U = ∅
M = ∅
while E ≠ ∅
   take a (u, v) ∈ E
   U ← U ∪ {u, v}
   M ← M ∪ (u, v)
   remove edges from E that are adjacent to u, v
end
```

A matching M gives a 2 approximation for the vertex cover problem as as a vertex cover. U is clearly a vertex cover as OPT has at least one vertex in every edge in M. As both vertices where placed in U : $|U| = 2 \cdot |M| \leq 2 * OPT$ Thus this is a 2 approximation.

3.1.3 Hardness

There are no known polynomial time algorithms which achieves a $(2 - \epsilon)$ approximation for constant $\epsilon > 0$. The best known approximation is $2 - \frac{loglogn}{2logn}$ approximation by Monien, Speckenmeyer 1985 [11]. This problem has an NP hardness of approximation lower bound of $10\sqrt{5} - 21 \approx 1.361$ assuming NP $\neq$ P due to Dinur, Safra 2004 [2].

3.2 Set Cover

Given a finite set U and a sequence of sets $\mathcal{S} = \{\mathrm{S}_i | \mathrm{S}_i \subset \mathrm{U}\}$ where $\bigcup_{\mathrm{S}_i \in \mathcal{S}} \mathrm{S}_i = \mathrm{U}$ and a weight function $\omega : \mathcal{S} \to \mathbb{Q}^+$. A set $\mathcal{C} \subseteq \mathcal{S}$ is said to cover U when $\bigcup_{C_i \in \mathcal{C}} C_i = U$. The objective function is defined as $COST(\mathcal{C}) = \sum_{\mathrm{C}_i \in \mathcal{C}} \omega(\mathrm{C}_i)$. The *set cover problem* is to find a set cover $\mathcal{C}$ which minimizes $COST(\mathcal{C})$.

3.2.1 Generalizations

Set cover is a generalization of vertex cover. Given a vertex cover problem $G = (V, E)$ we have the corresponding set cover problem where $U = E$ and $S_i = \{adj(v_i)\}$ where $adj(v_i)$ is the the set of edges adjacent to v_i.

3.2.2 Approximation : Greedy

The greedy algorithm picks sets which admits uncovered elements at the lowest average cost.

$\mathcal{C} = \emptyset$
while $\mathrm{U} \neq \emptyset$
  pick S_i in $\mathcal{S}$ such that $\mathrm{S}_i \cap \mathrm{U} \neq \emptyset$ and $\frac{COST(\mathrm{S}_i)}{|\mathrm{U} \cap \mathrm{S}_i|}$ is minimized
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{S_i\}$
  $\mathrm{U} \leftarrow \mathrm{U} - S_i$
end
return $\mathcal{C}$

The harmonic number is defined (and bounded) as $ln\ n \leq H_n = \sum_{i=1}^{n} \frac{1}{i} \leq 1 + ln\ n$ and $H_n \approx ln\ n$. This algorithm achieves a ratio of $H_\Delta \leq H_n$ where $\Delta = \max |S_i|$.

3.2.3 Hardness

Even for unweighted set cover this is essentially the best approximation. For $\epsilon > 0$ there is no $(1-\epsilon)H_n$ (respectively $\Omega(\log\ n)$) approximation unless NP has $n^{O(loglogn)}$-time (respectively NP=P) deterministic algorithms by Feige 1998[4].

3.3 Metric Uncapacitated facility

Given a graph $G = (V, E)$ and sets we shall consider the sets $\mathcal{F}$, $\mathcal{C}$ such that $V = \mathcal{F} \cup \mathcal{C}$ and form a bipartite partition of G. The set $++ \not\mathcal{F}$ represents facilities and the set $\mathcal{C}$ represents cites. The cost of opening a facility is given by $f : \mathcal{F} \to \mathbb{Q}^+$. The cost of connecting a city to a facility is given by $d : \mathcal{C} \times \mathcal{F} \to \mathbb{Q}^+$. The objective function is defined for $X \subseteq \mathcal{F}$ as

$$COST(X) = \sum_{x_i \in X} f(x_i) + \sum_{c \in \mathcal{C}} min_{f \in X} d(c, f)$$

The *metric facility problem* is given $\mathcal{F}, \mathcal{C}$ and a metric cost d find the set $X \subseteq \mathcal{F}$ which minimizes $COST$.

3.3.1 Generalization

Non metric uncapacitated facilities is a generalization of set cover. Where the cities are given by $C = U$, the facilities are given by $F = S$ and

$$d(e_i, S_i) = \begin{cases} \infty \leftarrow e_i \in S_i \\ 0 \leftarrow e_i \notin S_i \end{cases}$$

.

3.3.2 Hardness : Metric facility location

The best known approximation algorithm is a 1.52 greedy approximation by M. Mahdian, Y. Ye, and J. Zhang 2002 [10]. There is a 1.463 hardness result unless NP $\subset$ TIME($n^{O(loglogn)}$) which was obtained by "reduction" of set cover to facility location by Guha and Khuller 1999 [8].

3.4 Steiner Tree

Given a graph $G = (V, E)$ and a weight function $\omega : E \to \mathbb{Q}^+$ and $R \subseteq V$. For some $E' \subseteq E$ we say $T = (V, E')$ is feasible if R is connected. The objective function is given by $COST(E') = \sum_{e \in E'} \omega(e)$. The *Steiner tree problem* is to find a feasible T which minimizes $COST$.

3.4.1 Approximation : Greedy via MST

There is a reduction of the Steiner tree to metric steiner.

$\boxed{\text{Steiner Tree}}$	$\boxed{\text{Metric Steiner Tree}}$
$I = (G = (V, E), R, \omega)$	$I' = (G' = (V, V \times V), R, \omega')$
weight : ω	weight : $\omega'(u, v) = \begin{cases} \text{if } (u,v) \in E \text{ then } \omega(u,v) \\ \text{if } (u,v) \notin E \text{ then weight of the shortest } (u,v) \text{ path in G} \end{cases}$
Solution : $T \subset E$	solution : $T' \subset V \times V$

Given T' there is T. As ω' perseveres the cost of $(u, v) \in E$ the optimal solution in I' does not exceed that in I. Given T' construct T as follows:

```
for each (u, v) ∈ E
  if (u, v) ∈ E  then  T = T ∪ (u, v)
  if (u, v) ∉ E  then  T = T ∪  minimum path from  u to v
break cycles (by taking out the most expensive edge)
```

The algorithm above gives $c(T') \geq c(T)$.

The Steiner Tree approximation takes and instance of a steiner tree problem and converts it into its corresponding metric steiner tree. Then we create an MST on the metric steiner tree restricted to R and this MST is a feasible solution.

Given an optimal steiner tree of cost OPT if we double the edges we obtain an Eulerian graph. Construct a tour and the cost is $2 \cdot OPT$. Construct a tree from this eulerian tour by "shortcutting" (skipping to) steiner vertices and vertices previous visited in the tour. Thus the MST is a 2 approximation.

3.4.2 Hardness

The best known approximation is $1 + \frac{ln3}{2} \approx 1.55$ by Robins and Zelikonsky 2000[12]. The best known hardness approximation of lower bound is $1 + \frac{1}{135} \approx 1.007$ subject to $CO - RP \neq NP$ by Thimm [13].

3.5 Traveling Salesperson

Given a complete graph with nonnegative edge weights, find a minimum cost cycle visiting every vertex exactly once.

3.5.1 Approximation

> 1. Find a MST, T, of G
> 2. Double every edge of the MST to get an Eulerian graph
> 3. Find and Eulerian tour, T_E on this graph
> 4. Output the tour that visits vertices of G in order of their first appearance in T.
>
> Let C be this tour

Because OPT is a spanning tree $cost(T) \leq OPT$. As T_E contains each edge of T twice $cost(T_E) = 2 \cdot cost(T)$ after shortcutting $cost(C) \leq cost(T_E)$. Thus we get $cost(C) \leq 2 \cdot OPT$. Thus we have a 2 approximation.

There is an better $\frac{3}{2}$ approximation due to Christofides 1976 [1]. The last algorithm constructed an euler tour by doubling all the edges on a MST to get an euler tour. If we observe that an a graph has an eulerian tour if all vertices have an even degree then we can focus on only the vertices with odd degree. Note there are an even number of odd degree vertices because the sum of the degree of all the vertices in the MST must be even. The algorithm is given below.

1. Find a MST, T, of G
2. Compute a minimum cost perfect matching M, on the set
of odd degree vertices. Add M to T to obtain a Eulerian graph.
3. Find an Euler tour, T_E of T
4. Output the tour that visits vertices of G in order of their first appearance in T_E. Let C be this tour

As before $COST(T) \leq OPT$. Consider an optimal TSP tour of G τ. Let τ' be the tour on the odd degree vertices (V') of V obtained by shortcutting τ. By the triangle inequality $COST(\tau') \leq COST(\tau)$. τ' is the union of perfect matchings on V' consisting of alternate edges of τ. The cheapest of these matchings M' has cost $COST(M') \leq COST(\tau') \leq OPT/2$. The cost of the Euler tour $COST(T_E) \leq COST(T)+COST(M) \leq \frac{3}{2}OPT$ Thus this is a $\frac{3}{2}$ approximation.

3.6 Clustering

Given as set $S = \{x_i | 1 \leq i \leq n\}$ and a metric function $d : S \times S \rightarrow Q^+$ Partition S into k "clusters" $\{C_i | 1 \leq i \leq k\}$. The diameter of a cluster $diameter(C_i) = \max_{x \in C_i, y \in C_i} d(x, y)$ The *clustering problem* is to partition a set so as to minimize the diameters of the clusters.

3.6.1 Approximation

This $O(kn)$ (k unary) 2 approximation is due to Gonzalez 1985 [5]. It picks k centers and finds r with the understanding that given the centers and r one can generate the clusters. The idea is to always pick the point that is the furthest from known cluster centers.

```
if k ≥ |S|
  C = {x|x ∈ S}
else
  while |C| < k
    i ← i + 1
    pick s ∈ (S − C) which maximizes d(s, C)
    r_i ← d(s, C)
    C = C ∪ {s}
  end while
  r ← r_i
return (C, r)
```

The first observation is that $r_i \geq r_{i+1}$. Consider the clusters from the optimal solution $\{D_i \ | 1 \leq i \leq k\}$ with radius r_{opt}. If every D_i contains an element from C then we are done because by the triangle inequality every point is at most $2r_{opt}$ from the center of D_i thus $r \leq 2r_{opt}$. If there are two points of C in the same disk D_i then consider the r_i which the first one was added. The two point are within the diameter of the D_i thus $r \leq r_i \leq 2r_{opt}$. Thus this algorithm gives a 2 approximation.

3.6.2 Hardness

$2 - \epsilon$ hardness of approximation subject to P $\neq$ NP, $\forall \epsilon > 0$ Feder, Greene [3].

3.7 Makespan problem

In the scheduling problem we are given m machines and a set of n jobs represented by the set $\{1...n\}$. The weight function is given by $\omega : \{1...n\} \rightarrow \mathbb{Q}^+$. The jobs assignment is given by $\sigma(i) = j$ meaning job i is placed on processor j. The objective function

$$COST(\sigma) = \max_{k:1\leq k\leq m} \sum_{i:1\leq i\leq n \text{ and } \sigma(i)=k} \omega(i)$$

The *scheduling problem* is to find a job assignment minimizing $COST$.

3.7.1 Approximation : Greedy

The problem was first studied by Ron Graham [6], [7] at Bell labs in 1966. Along with laying the foundation for concepts such as worst case analysis in scheduling theory, on-line algorithms and amortized analysis, he also developed the following greedy algorithms.

Online greedy : take the least busy processor and give it a job. This is an $2 - \frac{1}{m}$ [6] approximation (Graham's List Algorithm).

If the jobs are sorted so that $\omega(i) \leq \omega(i+1)$ then the greedy algorithm gives $\frac{4}{3} - \frac{1}{3m}$ approximation [7].

4 References

References

[1] N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem, Report 388, Graduate School of Industrial Administration, CMU, 1976.

[2] I. Dinur and S. Safra, On the hardness of approximating minimum vertex cover

[3] T. Feder and D. Greene, Optimal algorithms for approximate clustering. Proc. 20th ACM Symp. Theory of computing, pages 434–444, Chicago, Illinois, 1988.

[4] Uriel Feige A threshold of ln n for approximating set cover, Journal of the ACM, 45(4):634-652,July 1998

[5] Clustering to minimize the maximum inter-cluster distance Theoretical Computer Science, 38:293–306, 1985.

[6] R.L. Graham Bounds for certain multiprocessing anomalies, Bell System Technical Journal, 45 (1966):1563–1581.

[7] R.L. Graham Graham, R. Bounds for multiprocessing timing anomalies. SIAM Journal on Applied Mathematics 17 (1969), 263–269

[8] S. Guha and S. Khuller, Greedy strikes back: Improved facility location algorithms, Journal of Algorithms, 31:228-248, 1999

[9] S. Har-Peled, A practical approach for computing the diameter of a point-set, In Proc. 17th ACM Sympos. Comput. Geom., pages 177-186, 2001.

[10] M. Mahdian, Y. Ye, and J. Zhang A 1.52-approximation algorithm for the uncapacitated facility location problem. In Proceedings, International Workshop on Approximation Algorithms for Combinatorial Optimization Problems,pages 127–137, 2002.

[11] B. Monien and E. Speckenmeyer, Ramsey numbers and an approximation algorithm for the vertex cover problem. Acta Informatica, 22(1):115–123, 1985.

[12] G. Robins and A. Zelikovsky, Improved Steiner tree approximation in graphs In Proc. 11th ACM-SIAM Symposium on Discrete Algorithms, pp. 770-779, 2000.

[13] M. Thimm On the approximability of the Steiner tree problem, Theor. Comput. Sci. 1-3: 387-402 (2003)