The Design and Analysis of Algorithms

Lecture 24 Local Search I

Zhenbo Wang

Department of Mathematical Sciences, Tsinghua University

Content

Coping With NP-Hardness

Q. Suppose I need to solve an *NP-hard* problem. What should I do?

A. Theory says you're unlikely to find poly-time algorithm.

- *Must sacrifice one of three desired features*.

  Solve arbitrary instances of the problem.

  Solve problem in polynomial time.

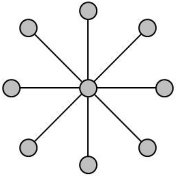
  Solve problem to optimality.

Gradient Descent: Vertex Cover

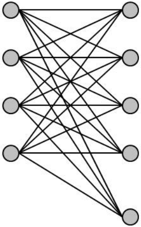
- *Vertex cover*. Given a graph $G = (V, E)$, find a subset of nodes S of minimal cardinality such that for each $(u, v) \in E$, either u or v (or both) are in S.
- *Neighbor relation*. $S \sim S'$ if S' can be obtained from S by adding or deleting a single node. Each vertex cover S has at most n neighbors.
- *Gradient descent*. Start with $S = V$. If there is a neighbor S' that is a vertex cover and has lower cardinality, replace S with S'.
- *Remark*. Algorithm terminates after at most n steps since each update decreases the size of the cover by one.

Gradient Descent: Vertex Cover

- *Local optimum*. No neighbor is strictly better.



optimum = center node only
local optimum = all other nodes



optimum = all nodes on left side
local optimum = all nodes on right side

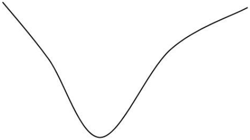
optimum = even nodes
local optimum = omit every third node

Local Search

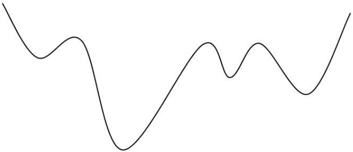
- *Local search*. Algorithm that explores the space of possible solutions in sequential fashion, moving from a current solution to a "nearby" one.

- *Neighbor relation*. Let $S \sim S'$ be a neighbor relation for the problem.

- *Gradient descent*. Let S denote current solution. If there is a neighbor S' of S with strictly lower cost, replace S with the neighbor whose cost is as small as possible. Otherwise, terminate the algorithm.



A funnel



A jagged funnel

Metropolis Algorithm

- *Metropolis algorithm*. [Metropolis, Rosenbluth, Rosenbluth, Teller and Teller 1953]

  Simulate behavior of a physical system according to principles of statistical mechanics.

  Globally biased toward "downhill" steps, but occasionally makes "uphill" steps to break out of local minima.

THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

Gibbs-Boltzmann function

- *Gibbs-Boltzmann function*. The probability of finding a physical system in a state with energy E is proportional to $e^{-E/(kT)}$, where $T > 0$ is temperature and k is a constant.
- For any temperature $T > 0$, function is monotone decreasing function of energy E.
- System more likely to be in a lower energy state than higher one.

  T large: high and low energy states have roughly same probability;

  T small: low energy states are much more probable.

Metropolis Algorithm

- *Metropolis algorithm*.

  Given a fixed temperature T, maintain current state S.

  Randomly perturb current state S to new state $S' \in N(S)$.

  If $E(S') \leq E(S)$, update current state to S'.

  Otherwise, update current state to S' with probability $e^{-\triangle E/(kT)}$, where $\triangle E = E(S') - E(S) > 0$.

Simulated Annealing

- *Simulated annealing*.

  T large $\Rightarrow$ probability of accepting an uphill move is large.

  T small $\Rightarrow$ uphill moves are almost never accepted.

  Cooling schedule: $T = T(i)$ at iteration i.

Optimization Problem	$\leftrightarrow$	Metal Object
Solutions	$\leftrightarrow$	States
Objective Function	$\leftrightarrow$	Energy
Optimal Solution	$\leftrightarrow$	State with Minimal Energy

Heuristic Methods

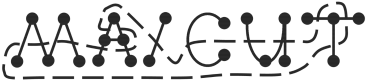
- *Simulated annealing*.
- *Tabu Search*.
- *Neural networks*.
- *Genetic algorithm*.
- *Ant colony algorithm* $\cdots$
- These topics are referred to the course "Modern Optimization Methods".

Maximum Cut

- *Cut*. Given an undirected graph $G = (V, E)$ with positive integer edge weights w_e, if $A, B \subseteq V$, $A \cap B = \emptyset$ and $A \cup B = V$, we say $(A, B) = \{(u, v) | u \in A, v \in B\}$ is a *cut* of G.

- *Maximum cut*. Find a cut (A, B) such that the total weight of edges crossing the cut is maximized.

$$w(A, B) := \sum_{u \in A, v \in B} w_{uv}.$$



Maximum Cut

- *Toy application*.

  n activities, m people.

  Each person wants to participate in two of the activities.

  Schedule each activity in the morning or afternoon to maximize number of people that can enjoy both activities.

- *Real applications*. Circuit layout, statistical physics.

- *Single-flip neighborhood*. Given a cut (A, B), move one node from A to B, or one from B to A if it improves the solution.

Maximum Cut: Local Search

MAX − CUT − LOCAL(G, w)

```
(A, B) ← random cut.
while there exists an improving node v do
    if v ∉ A then
        A ← A ∪ {v}.
        B ← B − {v}.
    else
        B ← B ∪ {v}.
        A ← A − {v}.
    end if
end while
return (A, B).
```

Local Search: Analysis

Theorem 2

Let (A, B) be a locally optimal cut and let (A^*, B^*) be an optimal cut. Then $w(A, B) \geq 1/2 \sum_e w_e \geq 1/2 w(A^*, B^*)$.

Pf. Local optimality implies that $\forall u \in A : \sum_{v \in A} w_{uv} \leq \sum_{v \in B} w_{uv}$.

Adding up all these inequalities yields:

$$2 \sum_{\{u,v\} \subseteq A} w_{uv} \leq \sum_{u \in A, v \in B} w_{uv} = w(A, B).$$

Similarly

$$2 \sum_{\{u,v\} \subseteq B} w_{uv} \leq \sum_{u \in A, v \in B} w_{uv} = w(A, B).$$

Now,

$$\sum_e w_e = \sum_{\{u,v\} \subseteq A} w_{uv} + \sum_{u \in A, v \in B} w_{uv} + \sum_{\{u,v\} \subseteq B} w_{uv} \leq 2w(A, B). \square$$

Maximum Cut: Big Improvement Flips

- *Local search*. Within a factor of 2 for *MAX-CUT*, but not poly-time!

- *Big-improvement-flip algorithm*. Only choose a node which, when flipped, increases the cut value by at least $\frac{2\epsilon}{n} w(A, B)$.

Claim. Upon termination, big-improvement-flip algorithm returns a cut (A, B) such that $(2 + \epsilon) w(A, B) \geq w(A^*, B^*)$.

Pf. Add $\frac{2\epsilon}{n} w(A, B)$ to each inequality in original proof.

Local optimality implies that
$\forall u \in A : \sum_{v \in A} w_{uv} \leq \sum_{v \in B} w_{uv} + \frac{2\epsilon}{n} w(A, B)$.

Adding up all these inequalities yields:

$$2 \sum_{\{u,v\} \subseteq A} w_{uv} \leq w(A, B) + \sum_{u \in A} \frac{2\epsilon}{n} w(A, B).$$

Maximum Cut: Big Improvement Flips–Con't

Similarly

$$2\sum_{\{u,v\}\subseteq B} w_{uv} \le w(A,B) + \sum_{u\in B}\frac{2\epsilon}{n}w(A,B).$$

Now,

$$\sum_e w_e = \sum_{\{u,v\}\subseteq A} w_{uv} + \sum_{u\in A, v\in B} w_{uv} + \sum_{\{u,v\}\subseteq B} w_{uv} \le (2+\epsilon)w(A,B). \square$$

Big Improvement Flips–Running Time

Claim. Big-improvement-flip algorithm terminates after $O(\epsilon^{-1} n \log W)$ flips, where $W = \sum_e w_e$.

Pf. Each flip improves cut value by at least a factor of $(1 + \epsilon/n)$.

Since $(1 + 1/x)^x \geq 2$ for any $x \geq 1$, we have $(1 + \epsilon/n)^{n/\epsilon} \geq 2$.

After n/ϵ iterations the cut value improves by a factor of at least 2.

Cut value can be doubled at most $\log W$ times. □

Maximum Cut: Context

Theorem 3 (Sahni-Gonzales 1976)

There exists a $\frac{1}{2}$-approximation algorithm for MAX-CUT.

Theorem 4 (Goemans-Williamson 1995)

There exists an 0.878-approximation algorithm for MAX-CUT.

Theorem 5 (Håstad 1997)

Unless $P = NP$, no 0.942-approximation algorithm for MAX-CUT.

Theorem 6 (Khot et al. 2007)

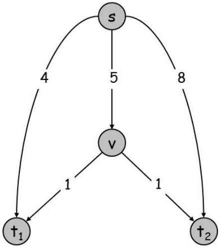
Under UGC, for any $\delta > 0$, and no algorithm that computes a $(0.878 + \delta)$-approximation to MAX-CUT.

Multicast Routing

- *Multicast routing*. Given a directed graph $G = (V, E)$ with edge costs $c_e \geq 0$, a source node s, and k agents located at terminal nodes $t_1, \cdots, t_k$. Agent j must construct a path P_j from node s to its terminal t_j.

- *Fair share*. If x agents use edge e, they each pay c_e/x.

1	2	1 pays	2 pays
outer	outer	4	8
outer	middle	4	5 + 1
middle	outer	5 + 1	8
middle	middle	5/2 + 1	5/2 + 1



Multicast Routing

- *Best response dynamics*. Each agent is continually prepared to improve its solution in response to changes made by other agents.
- *Nash equilibrium*. Solution where no agent has an incentive to switch.
- *Fundamental question*. When do Nash equilibria exist?

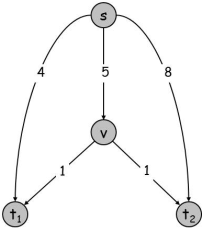
Ex.

Two agents start with outer paths.

Agent 1 has no incentive to switch paths (since $4 < 5 + 1$), but agent 2 does (since $8 > 5 + 1$).

Once this happens, agent 1 prefers middle path (since $4 > 5/2 + 1$).

Both agents using middle path is a Nash equilibrium.



Nash Equilibrium and Local Search

- *Local search algorithm*. Each agent is continually prepared to improve its solution in response to changes made by other agents.
- *Analogies*.

  Nash equilibrium: local search.

  Best response dynamics: local search algorithm.

  Unilateral move by single agent: local neighborhood.
- *Contrast*. Best-response dynamics need not terminate since no single objective function is being optimized.

Homework

- Read Chapter 12 of the textbook.
- Exercises 2 & 3 in Chapter 12.