Chapter 12 Local Search

aims at local optimum

Neighbor Relation

$S \sim S'$: S' is a neighboring solution of S - S' can be obtained by a small modification of S.

$N(S)$: neighborhood of S - the set $\{S' : S \sim S'\}$

The Vertex Cover Problem

The Metropolis Algorithm

```
SolutionType Metropolis()
{ Define constants k and T
  Start from a feasible solution S ∈ FS;
  MinCost = cost(S);
  while (1) {
    S' = Randomly chosen from N(S);
    CurrentCost = cost(S');
    if (CurrentCost < MinCost) {
      MinCost = CurrentCost; S=S';
    }
    else {
      With a probability e^(-Δcost/(kT)), let S=S';
      else break;
    }
  }
  return S;
}
```

Hopfield Neural Networks

Graph $G=(V,E)$ with integer edge weights w (positive or negative).

if $e=(u,v)$

$w_e \cdot u \cdot v < 0$, then it is good

In a configuration S, a node u is satisfied if the weight of incident good edges $\geq$ weight of incident bad edges.

A configuration is stable if all Nodes are satisfied.

$$\sum_{v:\, e=(u,v)\in E} w_e S_u S_v \leq 0$$

```
ConfigType StateFlipping()
{ start from arbitrary configuration S;
  while (! IsStable(S)) {
    u = GetUnsatisfied(S);
    S_u = -S_u;
  }
  return S;
}
```

the algorithm terminates at a stable configuration after at most $W = \sum_e |w_e|$ iterations

The Maximum Cut Problem

circuit layout, statistical physics

How good is this local optimum?

(A,B) be a local optimal partition and let (A^*, B^*) be a global optimal partition.

Then $w(A,B) \geq \frac{1}{2} w(A^*, B^*)$

big-improvement-flip:

Only choose a node which, when flipped, increase the cut value by at least

$$\frac{2\varepsilon}{|V|} w(A,B)$$

$$(2+\varepsilon)\, w(A,B) \geq w(A^*, B^*)$$

$$O\left(\frac{n}{\varepsilon} \log W\right) \text{ flips}$$

Chapter 10 NP-completeness

easiest: $O(N)$ we need to read inputs at least once

hardest: undecidable problems. (halting problem)

Deterministic Turing Machine: goes to next unique instruction

Nondeterministic Turing Machine: free to choose next step

Not all decidable problems are in NP

$P \subseteq NP$

Chapter 11 Approximation

deal with hard problems

getting around NP-completeness

— Find near-optimal solutions in polynomial time

— approximation algorithm

An algorithm has an approximation ratio of $\rho(n)$

C: the cost of the solution produced by the algorithm

C^*: the cost of an optimal solution

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n)$$

$\rho(n)$- approximation algorithm

polynomial-time approximation scheme (PTAS)

fully polynomial-time approximation scheme (FPTAS)

$O((1/\varepsilon)^2 n^3)$

Approximate Bin Packing:

Next Fit:

M — optimal number of bins

no more than $2M-1$

If next fit generates $2M$ (or $2M+1$) bins, then the optimal solution must generate at least $M+1$ bins.

First Fit:

not more than $\frac{17}{10}M$ bins

exist $17(M-1)/10$

Best Fit:

in the tightest spot

$T = O(N \log N)$ and bin no. $\leq 1.7M$

No online algorithm can always give an optimal solution.

all online algorithm use at least $\frac{5}{3}$ the optimal number of bins

Off-line algorithms

first fit decreasing never uses more than $\frac{11M}{9} + \frac{6}{9}$ bins

simple greedy heuristics can give good results

The knapsack Problem - 0-1 version

The approximation ratio is 2

The K-center Problem

Binary search for r

2-approximation

unless $P=NP$, there is no ρ-approximation for center-selection problem for any $\rho < 2$

Three aspects to be considered:

A: optimality

B: Efficiency

C: All instances

Even if $P=NP$, still we cannot guarantee $A+B+C$