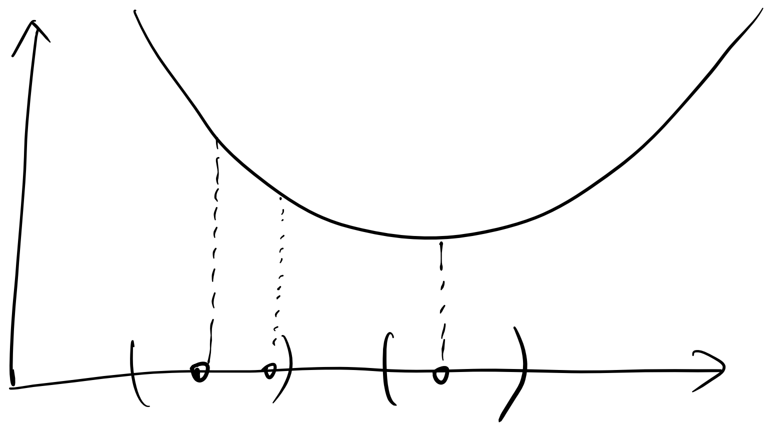
$f(n) = (n-10)^2$

find $\underset{n \in \mathbb{Z}}{\operatorname{argmin}} f(n)$



Optimization Problem (minization)

$\mathcal{C} = \{\text{feasible solution } S\}$

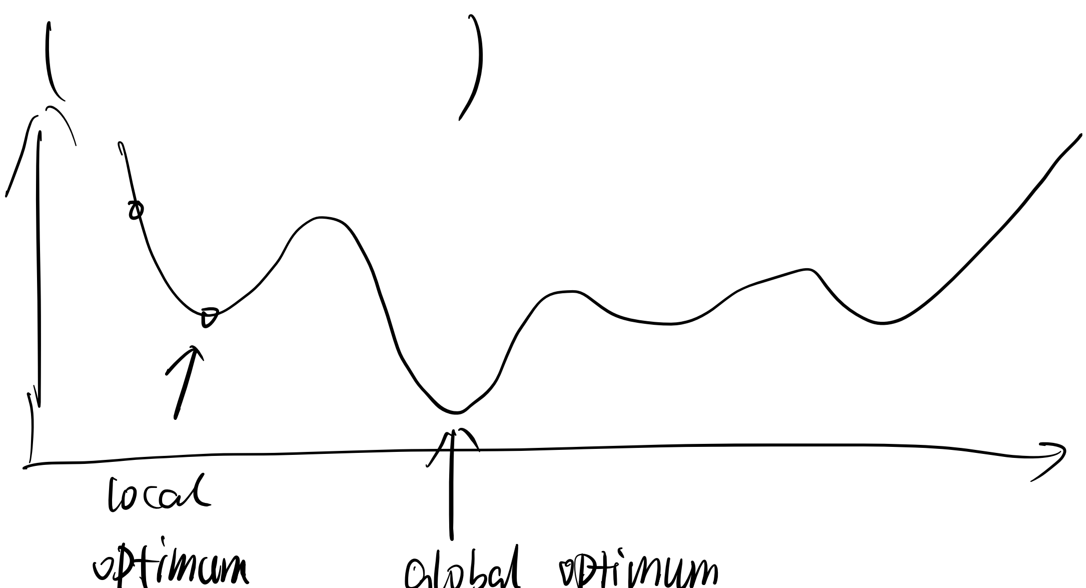
$c: \mathcal{C} \rightarrow \mathbb{Z}$

find $\underset{S \in \mathcal{C}}{\operatorname{argmin}}\ c(S)$

Neighborhood $N(S) = \{S' \mid S' \text{ is a neighbor of } S\}$

Local Search $(\mathcal{C}, c)$

1. pick a solution S from $\mathcal{C}$
2. while $N(S)$ has a better solution S' $(c(S') < c(S))$
3. $\quad S = S'$



Vertex Cover Problem

Given a graph $G=(V,E)$,

find a minimum vertex cover.

$S \subseteq V$ s.t every edge is incident a vertex in S.

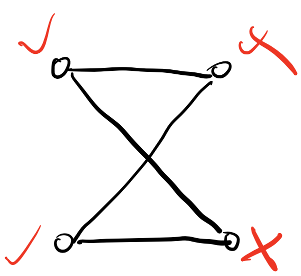
$\mathcal{C} = \{ \text{verex cover } S \}$

$c(S) = |S|$

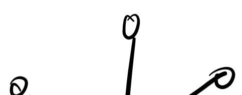
$N(S) = \{ \text{vertex cover } S' \mid S' \text{ can be obtained from } S \text{ by adding or deleting a single vertex} \}$

Local search $VC(V,E)$

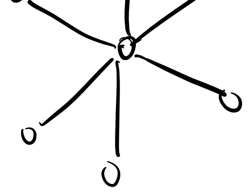
1. $S = V$
2. if $S - \{u\}$ is a vertex cover for some $u \in S$
3. $\quad S = S - \{u\}$

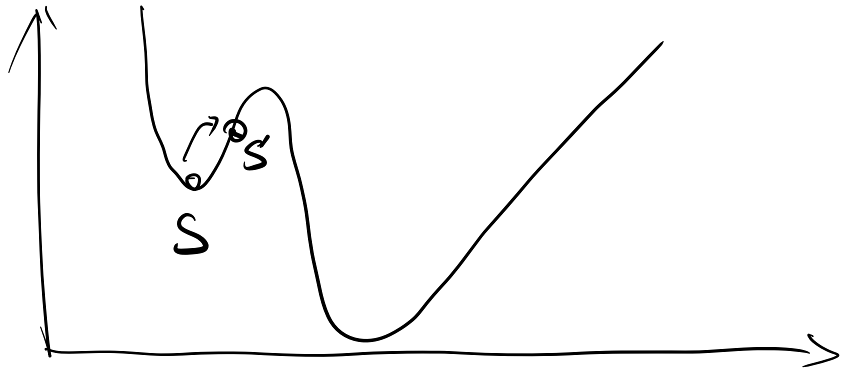


optimal



arbitrarily bad.





Metropolis Algorithm

Metropolis($\mathcal{C}$, c)

pick a solution S from $\mathcal{C}$

while true:

  randomly pick a solution S' from $N(S)$

  If $c(S') < c(S)$:

    $S = S'$ $\Delta c = c(S') - c(S)$

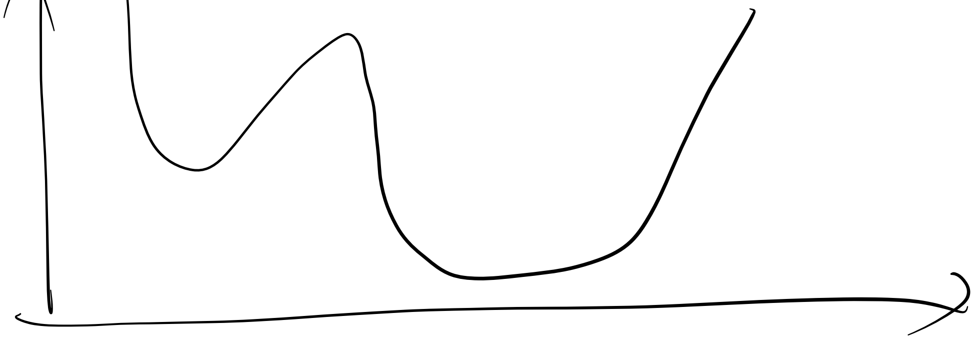
  else

    set $S = S'$ with probability $e^{-\Delta c / kT}$

  break when certain condition holds.

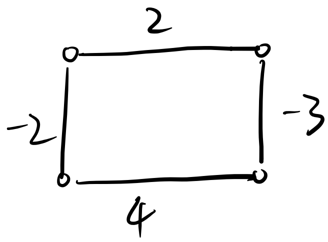
Simulated Annealing

gradually decrease T



Hopfield Neural Network Problem

Input: $G=(V,E)$ with edge weight $w: E \to \mathbb{Z}$



State of u: $S_u \in \{-1, +1\}$

A configuration S is an assignment of $+1$ or -1 to each node.

Given a configuration S, an edge $\overset{=(u,v)}{e}$ is good if

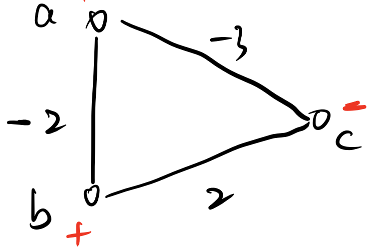
$$\left.\begin{array}{l}(1)\ W_e > 0 \text{ and } S_u \neq S_v \\ (2)\ W_e < 0 \text{ and } S_u = S_v\end{array}\right\} W_e S_u S_v < 0$$

, and bad otherwise

Objective 1. maximize $\sum_{e \text{ is good}} |W_e|$

Objective 2. every node is happy

+ good $= \{ab, bc\}$ good $= \{ab, ac\}$



bad = {ac} $\xrightarrow{c \to +1}$ bad = {bc}

Given a configuration S. a node u is satisfied if

$$\sum_{\substack{e=(u,v)\in E \\ e \text{ is good}}} |w_e| \geq \sum_{\substack{e=(u,v)\in E \\ e \text{ is bad}}} |w_e|$$

$$\Updownarrow$$

$$\sum_{e=(u,v)\in E} w_e s_u s_v \leq 0$$

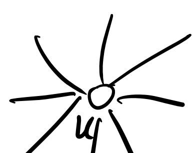
A configuration S is stable if all the nodes are satisfied.

State-flipping:

1. pick an arbitrary configuration S
2. while S is not stable
3. choose an unsatisfied node u
4. flip the state of u
5. return S

$$\Phi(S) = \sum_{e \text{ is good}} |w_e|$$

$S \xrightarrow{\text{flip } u} S'$



$$\Phi(S') = \Phi(S) - \sum_{\substack{e=(u,v)\in E \\ e \text{ is good} \\ \text{in } S}} |w_e| + \sum_{\substack{e=(u,v)\in E \\ e \text{ is bad in } S}} |w_e|$$

$$> \Phi(S)$$

$$\Downarrow$$

$$\Phi(S') \geq \Phi(S) + 1$$

$$0 \leq \Phi(S) \leq \sum_{e\in E} |w_e| = W$$

$\Rightarrow$ at most $O(W)$ flips

pseudo-polynomial

ϕ

input : $\log W$ bits

$N(S) = \{ S' \mid S'$ can be obtained from S by flipping the state of exactly one node $\}$

$$c(S) = \sum_{\substack{e \text{ is good} \\ \text{in } S}} |w_e|$$

Stateflip :

1. pick a configuration S
2. While $N(S)$ has a config S' with $c(S') > c(S)$
3. $S = S'$

every local maximum is a stable configuration

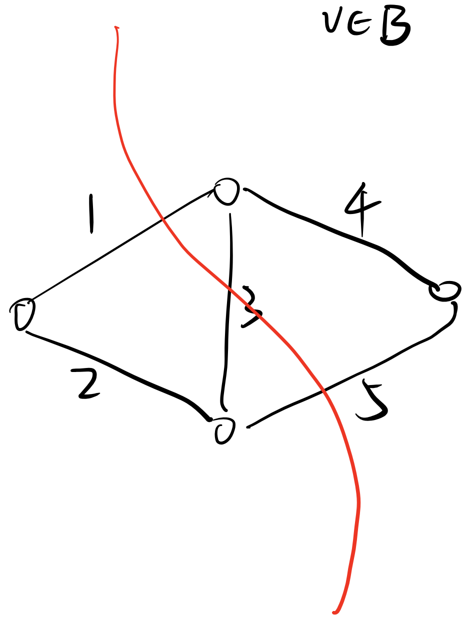
Maximum Cut

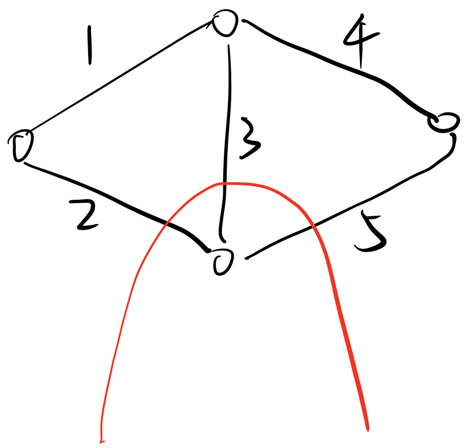
Given a weighted undirected graph $G = (V, E)$,

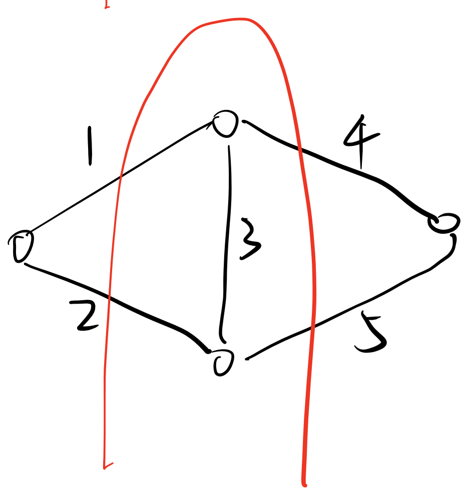
a cut (A, B) is a partition of V into two non-empty subsets A and B.

The weight of a cut (A, B) is

$$w(A,B) = \sum_{\substack{e=(u,v) \\ u \in A \\ v \in B}} w_e$$

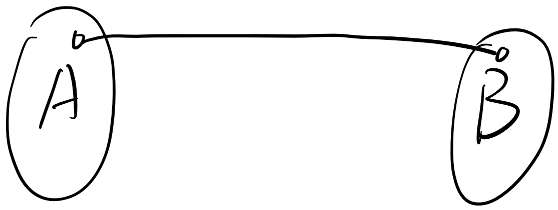






Input: a weighted undirected graph $G=(V,E)$

Output: a maximum cut.



A special case of Hopfield with $w_e > 0$ for all e

(e cross the cut iff e is good)

State-flip-Max-Cut

1. pick an arbitary cut (A, B)
2. while some node u is unsatisfied

$$\sum_{e=(u,v)} w_e < \sum_{e=(u,v)} w_e$$

~~e is good~~ e cross the cut

~~e is bad~~ e not cross the cut

3. flip the membership of u

in $O(W)$ iterations, reach local maximum

2-approximation

(A, B) is stable

for any $u \in A$,

$$\sum w_e \leq \sum_{e=(u,v)\in E} w_e$$

$e=(u,v)\in\tilde{E}$ ~~e not cross the cut~~ $v\in A$ ~~e cross (A,B)~~ $v\in B$

$$2\sum_{\substack{e=(u,v)\in\tilde{E}\\ u\in A\\ v\in A}} w_e = \sum_{u\in A}\sum_{\substack{e=(u,v)\in\tilde{E}\\ v\in A}} w_e \le \sum_{u\in A}\sum_{\substack{e=(u,v)\in\tilde{E}\\ v\in B}} w_e = w(A,B)$$

similarly,

$$2\sum_{\substack{e=(u,v)\in\tilde{E}\\ u\in B\\ v\in B}} w_e \le w(A,B)$$

$$w(A^*,B^*) \le \sum_{e\in\tilde{E}} w_e = \sum_{\substack{e=(u,v)\in\tilde{E}\\ u,v\in A}} w_e + \sum_{\substack{e=(u,v)\in\tilde{E}\\ u,v\in B}} w_e + \sum_{\substack{e=(u,v)\in\tilde{E}\\ u\in A\\ v\in B}} w_e$$

$$\le \frac{w(A,B)}{2} + \frac{w(A,B)}{2} + w(A,B)$$

$$= 2w(A,B)$$

iteration $= O(W)$

Idea: require big improvement in each iteration

flip a node u only when it increases $w(A,B)$ by a fraction of at least

$\frac{\varepsilon}{|V|}$ (ε is a small constant >0)

$$w(A',B') \ge \left(1+\frac{\varepsilon}{|V|}\right) w(A,B) \ge \left(1+\frac{\varepsilon}{n}\right) w(A,B)$$

$$\left(1+\frac{\varepsilon}{n}\right)^{n/\varepsilon} \ge 2$$

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

(A, B) is stable

for any $u \in A$,

$$\sum_{\substack{e=(u,v)\in E \\ \cancel{e \text{ not cross}} \\ \cancel{\text{the cut}} \\ v\in A}} w_e \le \sum_{\substack{e=(u,v)\in E \\ \cancel{e \text{ cross } (A,B)} \\ v\in B}} w_e + \frac{\varepsilon}{|V|} w(A,B)$$

$$2\sum_{\substack{e=(u,v)\in E \\ u\in A \\ v\in A}} w_e = \sum_{u\in A} \sum_{\substack{e=(u,v)\in E \\ v\in A}} w_e \le \sum_{u\in A} \sum_{\substack{e=(u,v)\in E \\ v\in B}} w_e + \sum_{u\in A} \frac{\varepsilon}{|V|} w(A,B) = w(A,B) + \frac{|A|}{|V|} \varepsilon\, w(A,B)$$

$$2\sum_{\substack{e=(u,v)\in E \\ u\in A \\ v\in A}} w_e \le w(A,B) + \frac{|B|}{|V|} \varepsilon\, w(A,B)$$

$$w(A^*, B^*) \le 2w(A,B) + \frac{|B|}{2|V|} \varepsilon\, w(A,B) + \frac{|A|}{2|V|} \varepsilon\, w(A,B)$$

$$\le \left(2+\frac{\varepsilon}{2}\right) w(A,B) \quad (\because |A|+|B|=|V|)$$

big $N(S)$ → avoids some local optimum

small $N(S)$ → fast search $N(S)$

$N(A,B) = \{(A',B') \mid (A',B')$ can be obtained be from (A,B) by flipping exactly one node $\}$

$O(n)$

↓

$O(n^k)$

k

Kernighan and Lin (1970)

K-L heuristic (A, B)

1. $(A_0, B_0) = (A, B)$
2. mark all nodes as unflipped.
3. for $i = 1, 2, \dots, n$
4. let (A_i, B_i) be the best cut that can be obtained from (A_{i-1}, B_{i-1}) by flipping exactly one unflipped node.
5. mark the node as flipped
6. return $\{(A_1, B_1), \dots, (A_n, B_n)\}$ as $N(A,B)$