Combinatorial optimization using Hopfield Net

Chintan B. Shah (06305013)
Nirav S. Uchat (06305906)

Computer Science and Engineering
IIT Bombay

Introduction and Motivation

Optimization Problem

- What is an "Optimization Problem" ?
- Many of them are NP-Complete (Hard in some sense)
- Finding an optimum solution is tough
- Finding a 'good' solution may be easy

Why Hopfield Nets

- Optimization problem maximizes/minimizes some quantity
- Hopfield Net minimizes energy
- This motivates us to model (and possibly solve) the optimization problem using Hopfield Net

How to solve the problem using Hopfield Net

- Choose the representation such that output of the neurons corresponds to the solution of the problem
- Choose the energy function for the problem such that its minimum value gives the best solution to the problem
- Compare the energy equation of the problem E_p to the energy equation of the Hopfield Net E_n
- Find the weight and threshold values

Agenda

Revision

- Travelling Salesman Problem
- Sorting

Problems in Graph Theory

- Vertex Cover
- Independent Set
- Clique
- Matching

Travelling Salesman Problem

Energy Equations

$E_1 = \frac{A}{2}\left[\sum_i(\sum_\alpha x_{i,\alpha} - 1)^2 + \sum_\alpha(\sum_i x_{i,\alpha} - 1)^2\right]$
$E_2 = \frac{1}{2}\left[\sum_i \sum_j \sum_\alpha d_{ij} x_{i\alpha}(x_{j,\alpha+1} + x_{j,\alpha-1})\right]$
$E_p = E_1 + E_2$
$E_n = -\sum_i \sum_\alpha \sum_j \sum_\beta w_{i\alpha,j\beta} x_{i\alpha} x_{j\beta} - \sum_i \sum_\alpha x_{i\alpha}\theta_{i\alpha}$

Weight and threshold values

Row Weight $w_{ij} = -A$
Column Weight $w_{ij} = -A$
Cross Weight $w_{ij} = -\left(\frac{d_{ij}+d_{ji}}{2}\right)$
$\theta_{ij} = 2A$

Sorting

Representation

Create a neuron corresponding to i^{th} value and j^{th} position
If the neuron fires, i^{th} value will be in j^{th} position

Equations

- A value can be in at most 1 position
- A position can accomodate at most 1 value
  $E_1 = \frac{A}{2}\left[\sum_i(\sum_\alpha x_{i,\alpha} - 1)^2 + \sum_\alpha(\sum_i x_{i,\alpha} - 1)^2\right]$
- Summation over *value* $*$ *position* should be minimized
  $E_2 = B\sum_i\sum_\alpha x_{i\alpha} * S_i * \alpha$

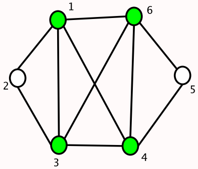
E_p and E_n

$E_p =$
$\frac{A}{2}\left[\sum_i(\sum_\alpha x_{i,\alpha} - 1)^2 + \sum_\alpha(\sum_i x_{i,\alpha} - 1)^2\right] + B\sum_i\sum_\alpha x_{i\alpha} * S_i * \alpha$
$E_n = -\sum_i\sum_\alpha\sum_j\sum_\beta w_{i\alpha,j\beta}x_{i\alpha}x_{j\beta} - \sum_i\sum_\alpha x_{i\alpha}\theta_{i\alpha}$

Weight and threshold values

$w_{ij} = -A$
$\theta = 2A - S_i B\alpha$

Vertex Cover (VC)



Representation

Create a neuron for each vertex
If neuron fires, the vertex will be in the VC

Equations

- Minimize the no. of vertices in the VC

$$E_1 = \sum_i v_i$$

- Cover each edge

$$E_2 = \sum_i \sum_j d_{ij} \overline{v_i \vee v_j}$$

Simplified energy equation

$$
\begin{aligned}
E_p &= A\sum_i \sum_j d_{ij}\overline{v_i \vee v_j} + B\sum_i v_i \\
&= A\sum_i \sum_j d_{ij}(1 - v_i \vee v_j) + B\sum_i v_i \\
&= A\sum_i \sum_j d_{ij}(1 - v_i - v_j + v_i v_j) + B\sum_i v_i \\
&= A\sum_i \sum_j d_{ij} v_i v_j - A\sum_i \sum_j d_{ij}(v_i + v_j) + B\sum_i v_i + A\sum_i \sum_j d_{ij}
\end{aligned}
$$

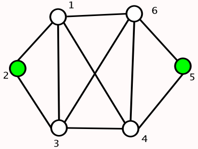
E_p and E_n

$$
\begin{aligned}
E_p &= A\sum_i \sum_j d_{ij} v_i v_j - A\sum_i \sum_j d_{ij}(v_i + v_j) + B\sum_i v_i \\
E_n &= -\sum_i \sum_j w_{ij} x_i x_j - \sum_i x_i \theta_i
\end{aligned}
$$

Weight and threshold values

$$
\begin{aligned}
w_{ij} &= -2Ad_{ij} \\
\theta_{ij} &= 2A\sum_i d_{ij} - B
\end{aligned}
$$

Independent set (IS)



Representation

Create a neuron for each vertex

If neuron fires, the vertex will be in the IS

Equations

- Maximize the no. of vertices in the IS

$$E_1 = \sum_i \overline{v}_i$$

- There shouldn't be an edge between the vertices in the IS

$$E_2 = \left[\sum_i \sum_j e_{ij}(v_i \wedge v_j)\right]$$

Simplified energy equation

$$E_p = A\left[\sum_i \sum_j e_{ij}(v_i \wedge v_j)\right] + B\sum_i \overline{v}_i$$
$$= A\left[\sum_i \sum_j e_{ij}(v_i v_j)\right] - B\sum_i v_i + B$$

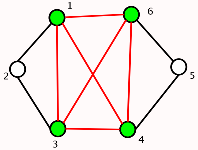
E_p and E_n

$$E_p = A\left[\sum_i \sum_j e_{ij}(v_i v_j)\right] - B\sum_i v_i$$
$$E_n = -\sum_i \sum_j w_{ij}x_i x_j - \sum_i x_i\theta_i$$

Weight and threshold values

$$w_{ij} = -Ae_{ij}$$
$$\theta_i = B$$

Clique



Representation

Create a neuron for each vertex

If neuron fires, the vertex will be in the Clique

Equations

- Maximize the no. of vertices in the Clique

$$E_1 = \sum_i \overline{v}_i$$

- Confirm that there is an edge between every pair of vertices

$$E_2 = -\left[\sum_i \sum_j (e_{ij} - 1)(v_i \wedge v_j)\right]$$

Simplified energy equation

$$E_p = -A\left[\sum_i \sum_j (e_{ij}-1)(v_i \wedge v_j)\right] + B\sum_i \overline{v}_i$$
$$= -A\left[\sum_i \sum_j (e_{ij}-1)(v_i v_j)\right] - B\sum_i v_i + B$$

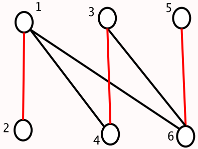
E_p and E_n

$$E_p = -A\left[\sum_i \sum_j (e_{ij}-1)(v_i v_j)\right] - B\sum_i v_i$$
$$E_n = -\sum_i \sum_j w_{ij} x_i x_j - \sum_i x_i \theta_i$$

Weight and threshold values

$$w_{ij} = Ae_{ij} - 1$$
$$\theta_i = B$$

Matching



Representation

Create a neuron for each edge

If neuron fires, the edge will be in the Matching

Equations

- Maximize the no. of edges in the Matching

$$E_1 = \sum_i \overline{e}_i$$

- The edges in the Matching shouldn't be adjacent

$$E_2 = \left[\sum_i \sum_j d_{ij}(e_i \wedge e_j) \right]$$

Simplified energy equation

$$E_p = A\left[\sum_i \sum_j d_{ij}(e_i \wedge e_j)\right] + B\sum_i \overline{e}_i$$
$$= A\left[\sum_i \sum_j d_{ij}(e_i e_j)\right] - B\sum_i e_i + B$$

E_p and E_n

$$E_p = A\left[\sum_i \sum_j d_{ij}(e_i e_j)\right] - B\sum_i e_i$$
$$E_n = -\sum_i \sum_j w_{ij} x_i x_j - \sum_i x_i \theta_i$$

Weight and threshold values

$$w_{ij} = -A d_{ij}$$
$$\theta_i = B$$

Conclusion

- mathematical formulation for the problems (requires ingenuity)
- intuitive values for weights and thresholds
- cute programs

References

- Xiaoming CHEN, Zheng TANG, "An Hopfield Network Learning for Minimum Vertex Cover Problem some references", SICE Annual Conference in Sapporo, pp. 1150-1155, 2004.
- Jenn-Shiang Lai, Sy-Yen Kuo, Ing-Yi Chen, "Neural networks for optimization problems in graph theory", Circuits and Systems, ISCAS '94, Vol.6, pp. 269-272, 1994.
- J. Ramanujam and P. Sadayappan, "Optimization using neural networks", in Proceedings of the 2nd IEEE International Conference on Neural Networks, vol. 2, pp. 325-332, San Diego, CA, July 1988
- Mark Atkins, "Sorting by Hopfield Net",IJCNN, Vol.2, pp. 65-68, 1989.