ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 331-336

The Maximum Weighted Clique Problem and Hopfield Networks

Brijnesh J. Jain, Fritz Wysotzki

Dept. of Electrical Engineering and Computer Science
Technical University of Berlin, Germany

Abstract. We propose a neural network solution of the *maximum weighted clique problem* (MWCP). The MWCP problem comprises the well-known maximum clique and maximum vertex-weighted clique problem as special cases. We present bounds for the parameter settings of a special Hopfield network to ensure energy descent to feasible solutions of the MWCP. To verify the theoretical results we show the effectiveness of the proposed approach in an experimental study.

1 Introduction

A clique of an undirected graph is a fully connected substructure and the *maximum clique problem* (MCP) is the problem of finding a clique with maximal number of vertices in that graph. The MCP is a well known classical NP-hard problem in combinatorial optimization, not only because of its theoretical but also for its practical implications. A straightforward generalization of the MCP is the *maximum weighted clique problem* (MWCP). Given an undirected graph with weights associated with each vertex and each edge, the MWCP is the problem of finding a clique, which maximizes a weight function $\omega(.)$. Since the MWCP comprises the MCP as a special case, it is at least of the same complexity as its unweighted counterpart.

Several real world applications as graph matching, information retrieval, experimental design, or signal transmission can be mapped to the MCP [3]. The MWCP arises, for example, in computer vision when matching two attributed graphs [2, 11]. Thus, due to its theoretical and practical importance, there are numerous approaches for approximating the classical MCP including several neural network heuristics [3]. But only few work has been devised to solve the more general MWCP. To our best knowledge investigations of neural energy minimizing methods to the MWCP in its most general form are missing.

In this paper we propose a Hopfield clique model for approximately solving the general MWCP. We present bounds for the parameters of the Hopfield clique network to ensure that the network performs an energy descent until it terminates in a stable state corresponding to a feasible solution.

This paper is organized as follows: Section 2 formally introduces the MWCP. Section 3 proposes the Hopfield clique model for solving the MWCP. Section 5 presents and discusses experiments. Finally, Section 6 concludes.

2 The Maximum Weighted Clique Problem

A *weighted graph* is a triple $Z = (V, E, \mu)$ consisting of disjoint sets V and E together with a mapping $\mu : V^2 \to \mathbb{R}_+$. The set V is the set of *vertices* and $E \subset V^2$ denotes the set of *edges*. Elements of E are unordered pairs $(i, j) \in V^2$ with $i \neq j$. The *weight mapping* μ assigns each vertex $i \in V$ and each edge $(i, j) \in E$ a positive *weight* $\mu(i, i) > 0$ and $\mu(i, j) = \mu(j, i) > 0$, respectively. For non-edges $(i, j) \in V^2 \setminus E$ the weights are given by $\mu(i, j) = 0$. The vertex set of a graph Z is referred to as $V(Z)$, its edge set as $E(Z)$, and its attribute function as μ_Z. A *normalized weighted graph* is a weighted graph Z with $\mu_Z(ij) \in [0, 1]$ for all $i, j \in V(Z)$. A *binary graph* is a normalized graph $Z = (V, \mu)$ with $\mu(V^2) \subseteq \{0, 1\}$. The number of vertices of a graph Z is its *order*, written as $|Z|$. By $\mathbb{G}^n$ we denote the set of weighted graphs of order n. The *adjacency matrix* of a graph Z of order n is the $n \times n$ matrix $A(Z) = (z_{ij})$ with entries $z_{ij} = \mu_Z(i, j)$. A subset $C \subseteq V(Z)$ is called *clique* of Z, if all vertices of C are mutually adjacent. The weight $\omega_{\alpha_E}^{\alpha_V}(C)$ of a clique C is defined by the weighted sum

$$\omega_{\alpha_E}^{\alpha_V}(C) = \alpha_V \cdot \sum_{i \in C} z_{ii} \; + \; \alpha_E \cdot \sum_{\substack{i,j \in C \\ j \neq i}} z_{ij} \tag{1}$$

where $\alpha_V \geq 0$ and $\alpha_E \geq 0$ are constants weighting the contribution of vertex and edge weights to the sum $\omega_{\alpha_E}^{\alpha_V}(C)$. A clique C is said to be *maximal* if $C \subseteq C'$ implies $C = C'$ for all cliques C'. A *maximum weighted clique* C of Z with respect to α_V and α_E is a clique with $\omega_{\alpha_E}^{\alpha_V}(C) \geq \omega_{\alpha_E}^{\alpha_V}(C')$ for all cliques C' of Z. The *maximum weighted clique problem* is the problem of finding a clique C in a graph Z with maximum weight $\omega_{\alpha_E}^{\alpha_V}(C)$.

3 A Hopfield Clique Network

Following the seminal paper of Hopfield and Tank [5], the general approach to solve combinatorial optimization problems (COP) maps the objective function of the COP to the energy function of a neural network. The constraints of the COP are included in the energy function as penalty terms, such that the global minima of the energy function correspond to the solutions of the COP. Here the objective function to map onto an energy function is given by (1).

Let $Z \in \mathbb{G}^n$ be a normalized graph with adjacency matrix $A(Z) = (z_{ij})$.[1] The Hopfield clique network (HCN) $\mathcal{H}_Z$ associated with Z consists of n fully interconnected units. There is an excitatory connection between units i and j with weight $w_{ij} = w_{ij}^{+} \geq 0$, if $(i, j) \in E(Z)$ and an inhibitory connection with weight $w_{ij} = -w_{ij}^{-} < 0$, if $(i, j) \notin E(Z)$. Thus Z uniquely determines the topology of $\mathcal{H}_Z$. For this reason we identify the units of $\mathcal{H}_Z$ with the vertices

[1] If Z is not normalized, we can first transform Z to a normalized graph Z', compute the solution in Z', and back-transform the solution to Z.

of Z and excitatory connections between two units with edges between their corresponding vertices. Inhibitory connections in $\mathcal{H}_Z$ represent non-edges in Z. The dynamical rule of $\mathcal{H}_Z$ is of the form

$$x_i(t+1) = x_i(t) + \sum_{j \neq i} w_{ij} o_j(t) + \theta_i \tag{2}$$

where $x_i(t)$ denotes the activation of unit i at time step t and θ_i is a constant external input applied to unit i. The output $o_i(t)$ of unit i is computed by a piecewise linear limiter transfer function of the form

$$o_i(t) = \begin{cases} 1 & : \quad x_i(t) \geq \tau_t \\ 0 & : \quad x_i(t) \leq 0 \\ x_i(t)/\tau_t & : \quad \text{otherwise} \end{cases} \tag{3}$$

where τ_t is a time dependent control parameter called *pseudo-temperature*. The output function $o_i(t)$ has a lower and an upper saturation point at 0 and 1, respectively. Starting with a sufficient large initial value τ_0 the pseudo-temperature is decreased according to an *annealing-schedule* to a final value τ_f. The energy function of the network to be minimized is then of the form

$$E(t) = -\frac{1}{2} \sum_i \sum_{j \neq i} w_{ij} o_i(t) o_j(t) - \sum_i \theta_i o_i(t) \tag{4}$$

To simplify the formulation of Theorem 3.1 we introduce some technical terms and notations: Let $\deg_E(i)$ be the number of excitatory connections incident to unit i. We call $\deg_E = \max\{\deg_E(i) \mid 1 \leq i \leq n\}$ the *excitatory degree* of $\mathcal{H}_Z$ and $\deg_I = \max\{n - \deg_E(i) - 1 \mid 1 \leq i \leq n\}$ the *inhibitory degree* of $\mathcal{H}_Z$. Since any vertex can have at most $n-1$ adjacent neighbors, the excitatory degree deg_E is less than n and therefore $deg_I \geq 0$. Let $\alpha_V, \alpha_E \in [0,1]$. By

$$f_{ij} = \begin{cases} \alpha_V z_{ii} & : \quad \text{if } i = j \\ \alpha_E z_{ij} & : \quad \text{if } \alpha_E > 0 \text{ and } i \neq j \\ 1 & : \quad \text{if } \alpha_E = 0 \text{ and } i \neq j \end{cases}$$

we denote the *weighting coefficients* with respect to α_V and α_E. Let $\theta^* = \max_i\{f_{ii}\}$ be the weighted maximum vertex weight of Z weighted by α_V and $\zeta_* = \min_{i \neq j}\{f_{ij}\}$ the weighted minimum edge weight of Z weighted by α_E. By $w^+ > 0$ we denote a positive constant with upper bound

$$w^+ < \frac{2}{n + \deg_I(\deg_E - 1)} = w^* \tag{5}$$

and by $w^- > 0$ we denote a positive constant with lower bound

$$w^- > \deg_E w^+ + \theta^* = w_*. \tag{6}$$

Provided an appropriate parameter setting is given Theorem 3.1 proven in [7] states that the dynamical rule (2) performs a gradient descent with respect to the energy function E where the global (local) minima of E correspond to the maximum (maximal) weighted cliques of Z.

Theorem 3.1 *Let Z be a normalized weighted graph and $\mathcal{H}_Z$ be a* HCN *associated with Z. Assume that $\tau_t \geq 1$ for all $t \geq 0$. If $w^+_{ij} = f_{ij} \cdot w^+$, $w^-_{ij} = w^-$, and $\theta_i = f_{ii} \cdot w^+$ for all $i, j \in V(Z)$, then*

1. *$E(t+1) \leq E(t)$ for all $t \geq 0$.*
2. *There is an one-to-one correspondence between the local minima of E and the maximal cliques of Z.*
3. *There is an one-to-one correspondence between the global minima of E and the maximum weighted cliques C of Z with respect to α_V and α_E.*

Proof: The basic idea to show the assertion proceeds as follows: (i) Use (6) to show that the local and global minima of E reside in the corners of the unit hypercube. Establish a bijection between global (local) minimizers of E and maximum weighted (maximal) cliques. (ii) To prove that (2) minimizes E, show $E(t+1) - E(t) = -1/2\Delta'(W + \tau I)\Delta$ where $\Delta = \boldsymbol{o}(t+1) - \boldsymbol{o}(t)$, W is the weight matrix, and I is the identity matrix. From (5) and (6) follows that $W + \tau I$ is positive definite for all $\tau \geq 1$ giving $E(t+1) - E(t) < 0$. With (i) and (ii) at hand it is then straightforward to show the theorem. □

From the first implication together with the fact that E is bounded follows that $\mathcal{H}_Z$ converges. Since the local minima of $E(t)$ correspond to maximal cliques, we can not guarantee that the network $\mathcal{H}_Z$ converges to an optimal solution corresponding to a maximum weighted clique. In addition the network can converge to unstable equilibrium points $\boldsymbol{u} \in \mathbb{R}^n$ of $E(t)$. Due to their instability, imposing random noise onto $\mathcal{H}_Z$ may shift the state vector $\boldsymbol{o}(t)$ away from $\boldsymbol{u}$. The upper bound of $w^+_{ij} < w^*$ ensures that $\mathcal{H}_Z$ performs a gradient descent with respect to E. The lower bound of $w^-_{ij} > w_*$ guarantees that $\mathcal{H}_Z$ converges to a feasible solution provided that unstable equilibrium points are avoided by imposing random noise onto the network.

4 Experiments

In our experiments we focused on the maximum vertex weighted problem (MVWCP), i.e. the MWCP with $\alpha_V = 1$ and $\alpha_E = 0$. This setting allows comparison with the Exponential Replicator Equations (REP) [4]. The Replicator dynamics is derived from evolutionary game theory and is despite its simplicity a powerful method to approximately solve the maximum clique problem. The Replicator approach is based on an expanded version of the Motzkin-Strauss Theorem [8] for the MVCP. Fast exact solutions to the MWCP as proposed in [1], [9] are confined to discrete vertex weights and therefore are not considered in our comparative study. Both algorithms, the HCN and the REP, were implemented in Java using JDK 1.2. All experiments were conducted on a multi-server Sparc SUNW Ultra-4.

Since there is no widely accepted test suite for the MWCP, we adopted *weighted random graphs* (RG), *weighted irregular random graphs* (IRG) [4] and *k-random cliques graphs* (kRCG) [6] as a test bed. Vertex weights of RG, IRG, and kRCG are identically distributed real values from the interval $[0, 1]$. Random

p	weight			time [sec]		
	ω_{HCN}	ω_{REP}	$\omega_{REP}/\omega_{HCN}$	t_{HCN}	t_{REP}	t_{REP}/t_{HCN}
Vertex weighted random graphs						
0.10	2.23	2.42	1.09	0.009	0.018	2.14
0.25	3.18	3.36	1.06	0.009	0.025	2.75
0.50	5.18	5.43	1.05	0.009	0.038	4.07
0.75	9.33	9.63	1.03	0.009	0.063	6.81
0.90	16.80	17.13	1.02	0.009	0.111	12.60
Vertex weighted irregular random graphs						
0.10	4.08	4.22	1.03	0.009	0.048	5.29
0.25	8.08	8.23	1.02	0.009	0.102	10.80
0.50	14.83	14.94	1.01	0.010	0.172	17.50
0.75	21.67	21.76	1.00	0.010	0.191	19.09
0.90	25.90	26.01	1.00	0.010	0.196	19.41
Vertex weighted k-random cliques graphs						
5	19.56	19.48	1.00	0.011	0.381	36.0
10	21.54	21.66	1.01	0.011	0.274	25.6
15	23.30	23.45	1.01	0.011	0.218	20.8
20	26.03	26.20	1.01	0.011	0.200	18.2
25	29.38	29.59	1.01	0.010	0.195	18.7

Table 1: Results of test series on vertex weighted graphs.

graphs are graphs where the occurrence of each edge has an prespecified edge probability p. Irregular random graphs are random graphs with edge probability p but non-identically distributed vertex degrees. The k-random cliques graphs consist of the union of k cliques of randomly chosen size. As opposed to RG, IRG and kRCG are irregular in the sense that their vertex degrees differ substantially – a feature shared by many real-world instances.

In our experiments we considered RG, IRGs and kRCGs with 100 vertices. For each edge probability $p = 0.1, 0.25, 0.5, 0.75, 0.9$ we generated 500 RGs and IRGs. Similarly we generated 500 test instances of kRCGs for each $k = 5, 10, 15, 20, 25$. Thus, we tested both algorithms on 7500 randomly generated weighted graphs.

Table 1 summarizes the results. The first half of the table shows the average maximal weight ω_{HCN} and ω_{REP} of a clique found by HCN and REP, resp., and the ratio of the average maximal weights $\omega_{REP}/\omega_{HCN}$. The second half of each table summarizes the average time t_{HCN} and t_{REP} required by HCN and REP, resp., to find a solution, and their ratio of the average times t_{REP}/t_{HCN}. The time is measured in sec.

The results show that REP returns in average slightly better solutions than HCN for almost all cases. For dense and irregular graphs, however, computation time of REP increases while the time of HCN is roughly constant irrespective of the type of input graph such that HCN is twice up to 36 times faster than REP. Altogether, HCN appears to be more robust on different types of weighted random graphs with respect to speed and might be a good alternative for large scale problems.

5 Conclusion

We have proposed a Hopfield clique network for approximately solving the MWCP. We presented bounds for the parameters to ensure that the network performs an energy descent until it converges to a local minimum corresponding to a feasible solution of the MWCP. In experiments we have demonstrated the effectiveness of the proposed algorithm for the MVWCP with respect to quality of solution and computational time.

Appendix: Parameter Setting

We used the following parameters in our experiments:

HCN	REP
$w^+ = \dfrac{2}{n + deg_I(deg_E - 1)}$ $w^- = deg_E w^+ + \theta^*$ $\tau_0 = n$ $\tau_{t+1} = a_t \tau_t$ $a_t = \begin{cases} 0.9 & : \text{ if } t \leq 3 \\ 0.5 & : \text{ if } t > 3 \end{cases}$	$\kappa = 5$ $\epsilon = 10^{-6}$ The parameters κ and ϵ control the speed and precision of REP. For details we refer to [10].

References

[1] L. Babel. A fast algorithm for the maximum weight clique problem. *Computing*, 52(1):31–38, 1994.

[2] S. Bischoff, D. Reuss, and F. Wysotzki. Applied connectionist methods in computer vision to compare segmented images. In A. Günter, R. Kruse, and B. Neumann, editors, *KI 2003: Advances in Artificial Intelligence. 26th Annual Conference on AI*, LNAI 2821, pages 312–326. Springer-Verlag, 2003.

[3] I. Bomze, M. Budinich, P. Pardalos, and M. Pelillo. The maximum clique problem. In D.-Z. Du and P. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 4, pages 1–74. Kluwer Academic Publishers, Boston, MA, 1999.

[4] I. Bomze, M. Pelillo, and V. Stix. Approximating the maximum weight clique using replicator dynamics. *IEEE Transactions on Neural Networks*, 11:1228–1241, 2000.

[5] J. Hopfield and D. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.

[6] A. Jagota. Approximating maximum clique with a Hopfield network. *IEEE Trans. Neural Networks*, 6:724–735, 1995.

[7] B. Jain and F. Wysotzki. The maximum ω-clique problem and the Hopfield ω-clique model. Submitted for publication.

[8] T. Motzkin and E. Strauss. Maxima for graphs and a new proof of a Theorem of Turan. *Canadian Journal of Mathematics*, 17:533–540, 1965.

[9] S. Niskanen and P. Östergard. Cliquer user's guide, version 1.0. Technical Report T48, Communications Laboratory, Helsinki University of Technology, Espoo, Finland, 2003.

[10] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11(8):1933–1955, 1999.

[11] M. Pelillo, K. Siddiqi, and S. Zucker. Attributed tree matching and maximum weight cliques. In *Proc. ICIAP'99-10th Int. Conf. on Image Analysis and Processing*, pages 1154–1159. IEEE Computer Society Press, 1999.