Solving the Maximum Edge Weight Clique Problem (MEWCP) Using Ant Colony Optimization (ACO)

Technical Report · July 2013

DOI: 10.13140/RG.2.1.4579.6321

1 author:

Rodrigo López
Consejo Nacional de Ciencia y Tecnología
14 PUBLICATIONS **32** CITATIONS

Network Theory Homework: Solving the Maximum Edge Weight Clique Problem (MEWCP) Using Ant Colony Optimization (ACO)

Rodrigo López Farías
Professors: G. Caldarelli, M. Riccaboni

August 31, 2013

1 Introduction

The clique problem is a problem related to find certain complete subgraphs in a graph. Theory about the clique problem is devoted to identifying special types of graph that admit more efficient algorithms, the clique problem has many applications such as bioinformatics and computational chemistry and social networks. The analysis of networks as such the detection of communities is commonly performed using Social Network Analysis (SNA) algorithms based on clustering. Their high computational costs and non-scalability on large-scale social networks is the main drawback[5].

The current work presents an extension of the study of the ACO capabilities for solving problems in networks as such the problem of finding the maximum clique problem [1]. Although in [2] is presented a generic algorithm for subset selection problems there is no a concrete study for solving MEWCP. ACO is used for solving several kind of problems as clustering [3][4], community detection [5], finding the shortest path and solving the TSP among others.

The current work presents preliminary results that shows potential applicability of ACO in large scale networks for solving the Maximum Edge Weight Clique Problem.

2 Problem Definition

The Maximum Edge Weighted Clique problem (MEWCP) can be defined as follows: Given a complete graph $G = (V, E)$ with V vertices and unrestricted edge weights c_{ij}, find a subclique of G with b or fewer nodes such that the sum of the weights in the clique is maximized. Since a given edge weight is included

in the sum only if the associated pair of nodes is in the subclique, a quadratic model of this problem is:

$$\texttt{max} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} x_i x_j \tag{1}$$

$$\texttt{s.t} \sum_{j=1}^{n} x_j \leq b \tag{2}$$

$$x_i \in \{0, 1\} \tag{3}$$

Where $x_j = 1$ if node j is in the subclique; else $x_j = 0$. This problem is *np*-hard [6].

3 Proposed Algorithm for maximum edge weight and clique

The algorithm described is based in the generic ACO algorithm in [1] for solving the maximum clique problem. This version of the algorithm finds the clique with a fixed size q maximizing the sum of the weights of the edges.

Data: A complete Graph $G = (V, E)$, the size Q of the desired clique
Result: A maximum weigthed clique of size Q
initialization;
repeat
  $C_k \longleftarrow \emptyset \ \forall k$;
  for *each ant* k **do**
    `Randomly choose a vertex` v_i;
    $C_k \longleftarrow v_i$;
    $Candidates \longleftarrow \{v_j \in V | (v_i, v_j) \in E\}$;
    $q \longleftarrow 0$;
    while $q < Q$ **do**
      `Choose a vertex` $v_i \in Candidates$ with probability $p(v_i)$;
      Remove the choosen vertex from *candidates*;
      $C_k \longleftarrow C_k \cup \{v_i\}$;
      $q \longleftarrow q + 1$;
    end
    `Compute the sum of all edges of the clique`
    $totalWeight = sumWeights(C_k)$;
    if $totalWeight > maxCliqueWeight$ **then**
      $maxCliqueWeight \longleftarrow totalWeight$;
      $C_{max} \longleftarrow C_k$;
    end
  end
  `Update pheromone trails`
until n *iterations*;

Algorithm 1: Search an approximate maximal weighted clique of size Q

The input of the algorithm is a complete weighted graph $G = (V, E)$ and the output is the approximated Maximum Edge Weighted Clique of size Q. The cliques C_k are initialized in order to construct a new clique each iteration. The vertex v_i is chosen using an uniform random distribution. The candidates are the vertex neighbors of the vertex v_i. In the next step a clique of Q is constructed choosing each vertex with a probability $p(v_i)$ and is added to clique C_k, the chosen vertex is removed from the set of $Candidates$. After constructing the clique C_k, the total sum of the weights are computed and the best clique is recorded in C_{max} according to highest total sum of weights so far. Finally the pheromone trails are updated.

3.1 Pheromone as learning component.

Pheromone trails are laid by ants on components of the graph G = (V,E) in which they are looking for a maximum clique.

The pheromone trails on the vertices V of the graph in the proposed algorithm. The quantity of pheromone on a vertex $v\ i \in V$ is denoted as τ_i. This quantity represents the learned desirability to select a vertex v_i from $Candidates$ to construct a clique.

3.2 Using pheromone trails.

The pheromone trails affects the probability of each vertex to be chosen for constructing the clique. A vertex v_i contains certain amount of pheromone $tau_i > 0$ limited by $tau_{min} < tau_i < tau_{max} \in R$. The vertex v_i is chosen among the elements in the set $Candidates$ with respect to a probability $p(v_i)$. This probability is defined in Equation 4

$$p(v_i) = \frac{\tau_i^\alpha}{\sum_{v_j \in Candidates} \tau_j^\alpha} \tag{4}$$

where α is a parameter which weights pheromone factors

3.3 Updating pheromone trails

When the cliques are constructed the amounts of pheromone are decreased in order to simulate evaporation for avoiding the convergence to a locally optimal solution with a factor of persistence $0 < \rho < 1$, this persistence is implemented as $\tau_i = \tau_i * \rho$. The pheromone trails in the clique C_k are updated according to Equation 5.

$$\tau_i = \tau_i + \frac{1}{1 + (maxCliqueWeight - sumWeights(C_k))K} \tag{5}$$

where $maxCliqueWeight$ is the maximum sum of weights of a clique so far, $sumWeights(C_k)$ is the sum of the weight of the clique C_k and $K > 1$ is a proposed constant in this document to weight the $sumWeights(C_k)$ difference respect $maxCliqueWeight$. This constant decreases the possibility to choose the bad solutions.The value of K is related with the ranges of weight values (narrower range of values, a bigger K value should be used).

4 Prove of concept

4.1 Test

The instance to prove the algorithm is a complete graph $G = (V, E)$ with 100 vertices, where the maximum edge weight clique known a priori is $C_{optimum} = \{21, 22, 30, 40, 50, 60, 70, 80, 90, 99\}$, all the edges in clique $C_{optimum,i,j} = 30$, the reminder edges not contained in $C_{optimum}$ are filled with an uniform random number [0,29]. The parameters of the ACO algorithm chosen were:

- $Iterations = 500$
- $k = 20$
- $Q = 10$
- $\alpha = 1$

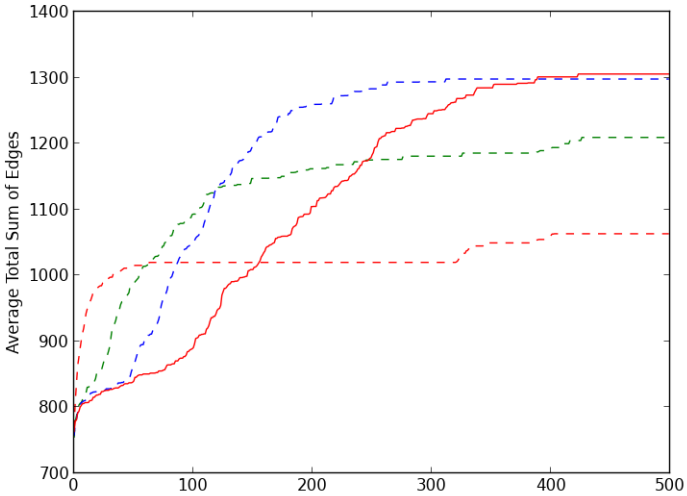

Figure 1: Convergence plot with $k = 20$

In order to test the algorithm we set different values of ρ, for each value the algorithm was run 20 times. The graph describes the mean convergence for each value ρ in 20 experiments through the iterations.

The Figure 1 and Figure 2 show the evolution of ACO with numbers of ants of $k = 20$ and $k = 30$respectively for different values of persistence ρ : The dotted red line is the evolution of ACO runing with $\rho = 0$, the green dotted line is when $\rho = 0.8$, the blue dotted line when $\rho = 0.9$ and the red line is when $rho = 0.95$. The Table 1 shows the time reference in seconds about the duration of one execution of the algoritm, the performance of the algorithm with the mean of maximum edge weight clique reached in 20 runs in the columns μ and the times the algorithm succeeded founding the optimal clique in the column *Success* (with an optimal weighted sum of 1350) for 20 and 30 ants for the different values of ρ.

ρ	Ex. Time	0		0.8		0.9		0.95	
#Ants	secs	μ	*Success*	μ	*Success*	μ	*Success*	μ	*Success*
$k = 20$	$\approx 40s$	1063	1/20	1209.5	12/20	1297.9	17/20	1305.5	17/20
$k = 30$	$\approx 57s$	1121	4/20	1308.4	17/20	1346.95	19/20	1346.09	19/20

Table 1: Effectiveness of ACO solving the MEWC problem.

The Table 1 shows the effectiveness of the pheromone demonstrating a poor performance when there is no pheromone persistence $\rho = 0$ and good accuracy when $\rho = .90, \rho = .95$ and this results can be improved increasing the number

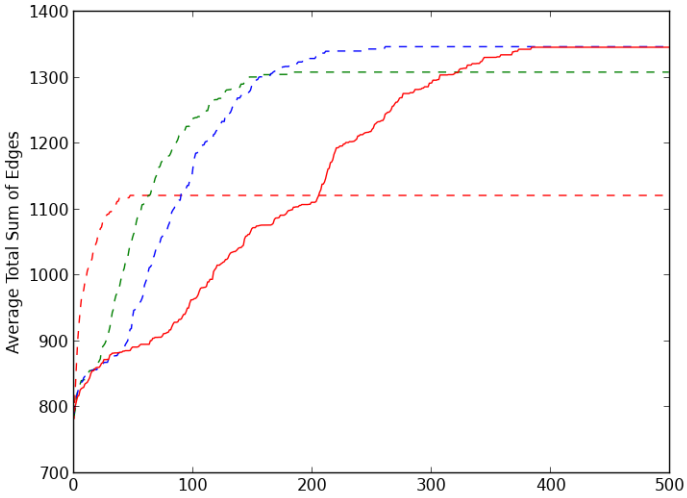

Figure 2: Convergence plot with $k = 30$

of ants.

4.2 Conclusions and proposals

An ACO algorithm for finding maximum edge weight clique is proposed as an extension of the work presented in [1] , the preliminary results confirm the capability of this algorithm for solving clique problems. This work propose the use of the parameter K in Equation 5, K has a strong influence in the performance of the algorithm but the value of this parameter was chosen empirically. In order to avoid the arbitrary chosen big value of K is important to find the way to compute the minimum value of this parameter to improve the performance of the algorithm. Test with "standard " benchmark functions against well known algorithms like RLS[7], DAGS [8] and GLS [9].

5 References

- [1] C. Solnon and S. Fenet, "A study of ACO capabilities for solving the maximum clique problem," Journal of Heuristics, pp. 1–31, 2006.
- [2] C. Solnon and D. Bridge, "Chapter I An Ant Colony Optimization Meta-Heuristic for Subset Selection Problems," pp. 1–23.
- [3] W. Dai, S. Liu, and S. Liang, "An improved ant colony optimization cluster algorithm based on swarm intelligence," Journal of Software, vol.

4, no. 4, pp. 299–306, 2009.

- [4] M. Hinne and E. Marchiori, "Cutting graphs using competing ant colonies and an edge clustering heuristic," Evolutionary Computation in Combinatorial Optimization 2011.
- [5]S. Sadi, Ş. Etaner-Uyar, and Ş. Gündüz-Öğüdücü, "COMMUNITY DETECTION USING ANT COLONY OPTIMIZATION TECHNIQUES," Proc. Int. Conf. Soft Computing, 2009.
- [6] B. Alidaee, F. Glover, G. Kochenberger, and H. Wang, "Solving the maximum edge weight clique problem via unconstrained quadratic programming," European Journal of Operational Research, vol. 181, no. 2, pp. 592–597, Sep. 2007.
- [7]R. Battiti and M. Protasi. Reactive local search for the maximum clique problem. Algorithmica, 29(4):610–637, 2001.
- [8]A. Grosso, M. Locatelli, and F. Della Croce. Combining swaps and node weights in an adaptive greedy approach for the maximum clique problem. Journal of Heuristics, 10(2):135–152, 2004
- [9]E. Marchiori. Genetic, iterated andmultistart local search for the maximum clique problem. In S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf, and G.R. Raidl, editors, Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2002: EvoCOP, EvoIASP, EvoSTim, volume 2279 of lncs, pages 112–121. Springer-Verlag, 2002