Optimization algorithms for the Max Edge Weighted Clique problem with Multiple Choice constraints

Alberto Ceselli [a,*] Roberto Cordone [a] Yari Melzani [a]
Giovanni Righini [a]

[a] *Dipartimento di Tecnologie dell'Informazione, Università degli Studi di Milano*
{alberto.ceselli,roberto.cordone,giovanni.righini}@unimi.it,
yari.melzani@gmail.com

Key words: Max Clique, semidefinite programming, tabu search, branch-and-bound

1 Introduction

Many relevant problems can be described and successfully solved using graph-based models. For instance, the side-chain placement problem in biology, the choice of efficient communication protocols in management science, backbone design in telecommunication networks and data mining applications share a common structure: they all require to find a maximum weighted clique in a suitable graph [1].

We tackle the *Max Edge Weighted Clique problem with Multiple choice Constraints* (MEWCMC). Given a graph with weights on both vertices and edges, and a partition of the vertex set, the MEWCMC asks to find a maximum weight clique, choosing one vertex for each class of the partition.

The classical Max Weighted Clique problem has been studied from many aspects. State of the art algorithms include [3], [4] and [5]. The version of the problem involving multiple choice constraints, instead, has been tackled only recently [2].

In this paper we first introduce models for the MEWCMC based on Binary Quadratic Programming (BQP) and Integer Linear Programming (ILP); we also introduce a model suitable to obtain a semidefinite relaxation of the MEWCMC. Since using commercial solvers on these models allows to solve only small size instances, we present an exact algorithm exploiting a semidefinite relaxation of the MEWCMC, combinatorial bounding, rounding and problem reduction procedures in a branch-and-bound framework. We also describe a Tabu Search heuristic, able to quickly find good quality solutions.

* Corresponding author

2 Models

It is given a graph $G(V,E)$, where V is a set of n vertices and $E \subseteq V \times V$ is a set of ℓ edges. It is also given a partition $K = \{V_1, V_2, .., V_m\}$ of V. Let $w_E : E \to \mathbb{R}$ and $w_V : V \to \mathbb{R}$ be two functions mapping respectively every edge and every vertex to a weight value. The MEWCMC consists in finding a clique in G, that is finding a set of vertices $M \subseteq V$, such that $(i,j) \in E \;\; \forall i,j \in M$, having maximum weight $z(M) = \frac{1}{2}\sum_{v_i \in M}\sum_{v_j \in M} w_{ij}$. Furthermore, in order to fulfill multiple choice constraints, M has to include exactly 1 vertex for each class, that is $|M \cap V_k| = 1 \, \forall V_k \in K$.
By introducing a binary variable x_i for each vertex $i \in V$, such that x_i takes value 1 if vertex $i \in M$, 0 otherwise, we can formulate the MEWCMC as the following BQP problem:

$$\text{maximize} \sum_{i \in V}\sum_{j \in V} w_{ij} x_i x_j \tag{1}$$

$$\text{s.t.} \sum_{i \in V_k} x_i = 1 \qquad \forall k \in K \tag{2}$$

$$x_i \in \{0,1\} \qquad \forall i \in V. \tag{3}$$

Multiple choice constraints (2) impose that exactly one vertex is selected for each class $V_k \in K$. This model can be either directly used by general purpose BQP solvers, or linearized using standard techniques, obtaining an ILP problem, suitable to be optimized by standard solvers. Both approaches, however, show to fail as the size of instances increases, mainly due to the poor quality of the continuous relaxation of both models.
Hence, we derive a matrix-based formulation of the problem as follows. First we define a matrix W, where each element w_{ij} with $i \neq j$ is the weight associated to edge $(i,j) \in E$ and each element w_{ii} is the weight associated to vertex $i \in V$, and we introduce a square matrix Y of variables y_{ij} of dimension n, where $y_{ij} = x_i x_j$ for each $i,j \in V$. The MEWCMC can then be stated as follows:

$$\text{maximize } W \bullet Y \tag{4}$$

$$\text{s.t. } \operatorname{rank}(Y) = 1 \tag{5}$$

$$\sum_k S_k \bullet Y = 1 \quad k = 1, ..., m \tag{6}$$

$$Y \succeq 0 \tag{7}$$

$$y_{ij} \in \{0,1\} \tag{8}$$

where $\bullet$ is the *Frobenius* matrix product, $rank(Y)$ is the rank of matrix Y, $Y \succeq 0$ imposes Y to be positive semidefinite, I is the identity matrix of dimension n. Constraints (5) and (7) guarantee that matrix Y can be represented as the product xx^T, and constraints (8) guarantee that each value in the matrix is binary. Inequalities (6) represent multiple choice constraints: matrices S_k are built in such a way that

$$\sum_{i\in V_k} Y_{ii} = \sum_{i\in V}\sum_{j\in V}(S_k)_{ij}Y_{ij} = S_k \bullet Y = 1 \quad \forall V_k \in K,$$

that is, having value 1 only on some diagonal elements, and value 0 elsewhere.

3 Algorithms

We devised an exact branch-and-bound algorithm exploiting formulation (4)–(8) and combinatorial arguments. In the remainder we sketch its main components.

Dual bounds. By considering model (4)–(8) and relaxing constraints (5) and (8) we obtain a semidefinite programming problem; this is a special case of convex optimization problem, that can be solved very efficiently by special purpose algorithms [6]. The relaxed model obtained in this way is strengthened in two ways. First, since one vertex has to be selected from each class of the partition, m vertices have to be selected overall; therefore, the constraint $I \bullet Y = m$ can be added to the formulation. Second, we state constraints in stronger forms, by disaggregating and exploiting some properties of BQP problems. At the same time, we consider the following value

$$DB_c = \frac{1}{2}\sum_{V_s\in K}\max_{i\in V_s}\{w_{ii} + \sum_{V_t\in K, t\neq s}\max_{j\in V_t}\{\frac{w_{jj}}{m-1} + w_{ij}\}\}; \tag{9}$$

and we prove by combinatorial arguments that expression (9) gives a valid dual bound to the MEWCMC. The best of the two bounds is kept as the final dual bound.

Primal bounds. At each node of the branch-and-bound tree we try to find good feasible solutions by (a) rounding the optimal solution of the semidefinite relaxation (b) correcting the solutions given by the combinatorial bounding procedure (9). Both methods are able to find good solutions in the early nodes of the branch-and-bound tree. We also devised a Tabu Search heuristic; it works as follows. We start from a clique M either corresponding to a known feasible solution or obtained by selecting a random vertex for each class of the partition K; at each step we explore the neighborhood given by all solutions obtained by replacing one the vertices in M by a different vertex of the same class, moving to the best solution in the neighborhood. We keep two tabu lists: the first keeps track of vertices recently removed from M, that cannot be selected again; the second keeps track of vertices recently introduced in M, that cannot be removed. We stop after a fixed number of steps. Preliminary experiments shows that by setting the length of the first tabu list to 8, that of the second tabu list to 1 and the maximum number of steps to 1000, this heuristic is able to produce optimal solutions on a large set of instances; besides being useful on its own, this heuristic is used to obtain tight primal bounds at the root node of the branching tree.

Branching. The diagonal elements of Y basically represent how much any vertex is (fractionally) selected in the semidefinite relaxation solution. Therefore, when primal and dual bounds do not match, we perform binary branching by selecting the class V_k having the highest number of these fractional entries; then we try to partition the vertices of class V_k in two subsets V_k^l and V_k^r, having the most balanced sum of fractional entries. In the left branch and right branches we respectively forbid to choose in the clique any vertex of the set V_k^l, and any vertex of the set V_k^r. We visit the branching tree in a best bound order.
Performance improving techniques. We improved the performances of our algorithm by (a) switching to complete enumeration when, due to the remotion of forbidden vertices, the size of the subproblem gets small enough (b) performing problem reduction tests, that is trying to fix each vertex to be part of the solution, computing the combinatorial dual bound on the remaining subproblem and removing from the problem such a vertex if the dual bound computed in this way is worse than the best known primal bound.

4 Experimental results

We implemented our algorithms in C, we used ILOG CPLEX 11.2 as both ILP and BQP solver, and DSDP5 as semidefinite programming solver. We performed experiments on two datasets. Dataset S1 is drawn from [2]; it is composed by 168 instances of four types, having weights on vertices and edges randomly drawn in different ranges. These instances require the optimization on graphs with up to 65 vertices. Dataset S2, instead, consists of 165 new instances of three types.The number of vertices of the graphs in these instances range from 30 to 300. Our experiments ran on a Centrino Core 2 3GHz PC, equipped with 2GB of RAM, in Linux 32 bit environment. A full description of Datasets and computational experiments is available online [1].
Our experimental campaign allowed us to find that (a) the tabu search algorithm is able to find the optimal solution on 79.76% of the instances in Dataset S1, and on 88.48% of the instances in Dataset S2; (b) the BQP solver of CPLEX using formulation (1)–(3) is not competitive with other methods (c) the exact algorithm proposed in [2] is outperformed by both CPLEX and our algorithm on both datasets (d) using the best of our ILP formulations, CPLEX could solve only 44% of the instances in Dataset S2 within a time limit of one hour, while our exact algorithm solved 77% of them; when both CPLEX and our algorithm terminate within the time limit, our algorithm is up to three orders of magnitude faster, and when none of them terminates, the remaining gap between primal and dual bounds for our algorithm is a fraction of that of CPLEX.

[1] http://www.dti.unimi.it/∼ceselli/EWCMC.html

References

[1] Y. Melzani (2009) “Mathematical programming algorithms for the Max edge weighted clique problem with multiple choice constraints”, Master thesis, Dipartimento di Tecnologie dell’Informazione, Università degli Studi di Milano.

[2] A. Bosio (2005) “A mathematical programming algorithm for the Max edge weighted clique problem with multiple choice constraints”, Degree thesis, Dipartimento di Tecnologie dell’Informazione, Università degli Studi di Milano.

[3] B. Alidaee, F. Glover, G. Kochenberger, H. Wang (2007) “Solving the maximum edge weight clique problem via unconstrained quadratic programming”, European Journal of Operational Research, 181(1): 592–597.

[4] M. M. Sørensen (2004) “New facets and a branch-and-cut algorithm for the weighted clique problem”, European Journal of Operational Research, 154(1): 57–70.

[5] M. Hunting, U. Faigle, W. Kern (2001) “A Lagrangian Relaxation Approach to the Edge-Weighted Clique Problem”, European Journal of Operational Research, 131

[6] S. J. Benson, Y. Ye (2005) “DSDP5: Software for Semidefinite Programming”, Technical report, Mathematics and Computer Science Division, Argonne National Laboratory.