Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Discrete Mathematics 36 (2010) 159–166
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

The biclique k-clustering problem in bipartite graphs and its application in bioinformatics

V. Acuña[1,*], C. E. Ferreira[2,†], A. S. Freire[3,†] and E. Moreno[4,‡]

∗ *Université Claude Bernard, Lyon I, France*

† *IME-USP, São Paulo, Brazil*

‡ *Universidad Adolfo Ibáñez, Santiago, Chile*

Abstract

In this paper we study the biclique k-clustering problem (BkCP) in bipartite graphs, a generalization of the maximum edge biclique problem which has several applications in biological data analysis. We present an application of the BkCP in bioinformatics, and introduce two integer linear formulations for the problem. Finally, we discuss the approximability of the problem and show computational experiments with random generated instances and also with instances that come the application.

Keywords: maximum edge biclique, biclique clustering, biclustering.

1 Introduction

A *biclique* is a complete bipartite graph, and a *biclique k-clustering* is a set of k vertex disjoint bicliques. In this paper we study the *biclique k-clustering*

[1] Email: viacuna@biomserv.univ-lyon1.fr
[2] Email: cef@ime.usp.br
[3] Email: afreire@ime.usp.br
[4] Email: eduardo.moreno@uai.cl

1571-0653/$ – see front matter
doi:10.1016/j.endm.2010.05.021

problem (BkCP) in bipartite graphs, a generalization of the *maximum edge biclique problem* (MEBP). The MEBP in bipartite graphs is NP-hard [9] and for sufficiently dense graphs there is a 2-approximation algorithm introduced in [5] for the problem. But, as shown in [1], the weighted version of the problem (i.e. with weights on the edges) is hard to approximate. There are many applications of the MEBP in biological data analysis and other areas (see [7] for a very comprehensive survey, and see [6] for a slightly different variant of the problem with application in multicast network design).

The aim that has motivated this work is to facilitate the biological interpretation of some mathematical structures describing the metabolism of a cell. One of these models is elementary modes analysis of metabolic networks. These methods reduce the set of all manners the metabolic network can work (the modes of the system) into a much smaller, but equivalent, subset of them. An elementary mode (EM) corresponds to a minimal set of reactions that can be used maintaining the system in steady state (when the amount produced of each compound is equal to the amount consumed).

Since the number of EMs obtained in real networks is still huge [11], it is desired to cluster groups of EMs that share a common group of reactions. This could led to identify sub-structures in the metabolic network that could explain, for example, a common function in the metabolism. Up to now, biologists construct cluster of modes [10] and cluster of reactions [8] separately, but it has been observed inconsistencies between the clusters obtained. Our objective is to provide a procedure to cluster both, reactions and EMs, simultaneously, in order to guarantee the desired consistency among the clusters. More details about this application is beyond the scope of this paper, thus here we focus on the BkCP itself.

In section 2 we discuss the approximability of the BkCP and introduce two integer linear formulations for it. The first one is a generalization of an existing formulation for the MEBP introduced in [5]. The second formulation is meant to be solved by a column generation algorithm. In section 3 we show computational experiments with random generated instances and also with instances that come from the application. Finally, in section 4 we conclude the paper and point some future work.

2 Formulations for the biclique k-clustering problem

First we describe the *biclique k-clustering problem*. It is given a bipartite graph $G = (V, E)$, where $V := R \cup C$, $R \cap C = \emptyset$, $|R| = n$ and $|C| = m$. We say that a subgraph $B = (V_B, E_B)$ of G, where $V_B \subseteq V$ and $E_B := \{uv \in E \mid u, v \in V_B\}$,

is a *biclique* of G if B is a complete bipartite graph, and the size of B is given by $s(B) := |E_B|$. A *biclique k-clustering* of G is a set $\mathcal{B}$ of k vertex disjoint bicliques of G, and the size of $\mathcal{B}$ is given by $S(\mathcal{B}) := \sum_{B\in\mathcal{B}} s(B)$. The *biclique k-clustering problem* (BkCP) consists of finding a biclique k-clustering with maximum size. As shown in [9], the MEBP (i.e. the BkCP for $k = 1$) is NP-hard, thus in general the BkCP is NP-hard as well.

We say that two edges uv and pq of E, such that $u, p \in R$ and $v, q \in C$, are *incompatible* if $u \neq p$, $v \neq q$ and $uq \notin E$ or $pv \notin E$. Using the following lemma we derive a simple formulation for the BkCP.

Lemma 2.1 (M. Dawande, P. Keskinocak and S. Tayur [5]) *A bipartite graph $B = (V_B, E_B)$ is a biclique if and only if for each pair of edges $e, f \in E_B$ we have that e and f are not incompatible.*

Let $\mathcal{I} := \{\{uv, pq\} \in E \times E \mid uv \text{ and } pq \text{ are incompatible}\}$ be the set of all pairs of incompatible edges of E. We define variables x^b_{uv} for all $uv \in E$ and $b = 1, 2, \ldots, k$ with the following interpretation: if $x^b_{uv} = 1$ then the edge uv is in the biclique with label b, and $x^b_{uv} = 0$ otherwise. For all $uv \in E$ with $u \in R$ and $v \in C$, let $\delta(uv) := \{pq \in E \mid p \in R;\ q \in C \text{ and } p = u \text{ or } v = q\}$ be the set of all edges adjacent to uv, including uv itself. Now we introduce a formulation for the BkCP:

$$
\begin{aligned}
& \max \sum_{b=1}^{k} \sum_{uv\in E} x^b_{uv} \\
(\mathrm{P_{MBC}}) \quad & \text{s.t. } x^b_{uv} + x^b_{pq} \leq 1, && \text{for all } (uv, pq) \in \mathcal{I} \text{ and } b = 1, 2, \ldots, k && (1)\\
& x^b_{uv} + x^d_{pq} \leq 1, && \forall\ uv \in E,\ pq \in \delta(uv) \text{ and } \forall\ 1 \leq b, d \leq k \text{ s.t. } b \neq d && (2)\\
& x^b_{uv} \in \{0, 1\}, && \text{for all } uv \in E \text{ and } b = 1, 2, \ldots, k && (3)
\end{aligned}
$$

This formulation is a generalization of a formulation for the MEBP. Indeed, if $k = 1$ the formulation reduces to the one presented in [5]. Let X_{MBC} be the set of all feasible solutions of ($\mathrm{P_{MBC}}$). Given a feasible solution $x \in X_{\mathrm{MBC}}$, let $B_b(x) = (V_b, E_b)$ be the subgraph of G induced by the edge set $E_b := \{uv \in E \mid x^b_{uv} = 1\}$, for $b = 1, 2, \ldots, k$, and define $\mathcal{B}(x) := \{B_1(x), B_2(x), \ldots, B_k(x)\}$. By lemma 2.1, constraints (1) guarantee that $B_b(x)$ is a biclique of G, for $b = 1, 2, \ldots, k$. Constraints (2) guarantee that $V_b \cap V_d = \emptyset$, for all $b \neq d$ such that $1 \leq b, d \leq k$. Thus, $\mathcal{B}(x)$ is a biclique k-clustering of G. Clearly, the converse also holds, i.e. there exists a feasible solution in X_{MBC} for each biclique k-clustering of G (actually, for each biclique k-clustering of G there are many equivalent solutions in X_{MBC} if we consider symmetric solutions, i.e. solutions which are essentially the same, except for label switching).

Note that an equivalent way to formulate the BkCP is to define variables

$y^b_{uv} = 1 - x^b_{uv}$, for all $uv \in E$ and $b = 1, 2, \ldots, k$, replace each constraint $x^k_{uv} + x^l_{pq} \leq 1$ from ($\mathrm{P}_{\mathrm{MBC}}$) by $y^k_{uv} + y^l_{pq} \geq 1$ and minimize over $\sum_{v \in V_F} y^b_{uv}$. In this case the interpretation is that an edge uv is in the biclique b if and only if $y^b_{uv} = 0$, and we want to minimize the number of edges outside the bicliques. Let us denote this second formulation by ($\mathrm{P}_{\overline{\mathrm{MBC}}}$).

A simple way to obtain a 2-approximation algorithm for the ($\mathrm{P}_{\overline{\mathrm{MBC}}}$) is rounding an optimal solution of its linear relaxation (all variables with value $\geq \frac{1}{2}$ are rounded up to 1 and all variables with value $< \frac{1}{2}$ are rounded down to 0). But in ($\mathrm{P}_{\mathrm{MBC}}$) if we set each variable to $\frac{1}{2}$ we get a feasible solution with weight $(|E|k)/2$. So, the optimal solution for the linear relaxation of ($\mathrm{P}_{\mathrm{MBC}}$) is at least $(|E|k)/2$. This upper bound is completely useless for $k \geq 2$. Hence, the approach of rounding the solution of the linear relaxation in this case does not provide a good feasible solution as in ($\mathrm{P}_{\overline{\mathrm{MBC}}}$). This can be explained by the fact that ($\mathrm{P}_{\mathrm{MBC}}$) is a special case of the (maximum) independent set problem and ($\mathrm{P}_{\overline{\mathrm{MBC}}}$) is a special case of the (minimum) vertex cover problem, where the input graph is defined in the following way: there is one vertex for each variable of ($\mathrm{P}_{\mathrm{MBC}}$) and there is one edge between each pair of vertices which appear in a same constraint of ($\mathrm{P}_{\mathrm{MBC}}$). In fact, Berman and Schnitger [3] show that the independent set problem is NP-hard to approximate by a factor of $O(|V|^c)$, for some constant $c > 0$. As shown in [1] the same result holds for the maximum weighted edge biclique problem. In the case of the BkCP we can consider formulation ($\mathrm{P}_{\overline{\mathrm{MBC}}}$) to obtain an approximation result.

Below we introduce another formulation for the BkCP. Let $\mathcal{B}$ be the set of all bicliques of G, and let $\mathcal{B}(v) := \{B \in \mathcal{B} \mid v \in B\}$ be the set of all bicliques which contain the vertex v, for all $v \in V$. Define the variables $x_B \in \{0, 1\}$, for all $B \in \mathcal{B}$, with the following interpretation: $x_B = 1$ if and only if the biclique B is in the biclique k-clustering of G. Now, consider the following formulation for the BkCP.

$$(\mathrm{P}^2_{\mathrm{MBC}}) \qquad \begin{aligned} \max \ & \sum_{B \in \mathcal{B}} s(B) \cdot x_B \\ \text{s.t.} \ & \sum_{B \in \mathcal{B}(v)} x_B \leq 1, \qquad \text{for all } v \in V \qquad (4) \\ & x_B \in \{0, 1\}, \text{ for all } B \in \mathcal{B} \qquad (5) \end{aligned}$$

Constraints (4) guarantee that the chosen bicliques (i.e. the bicliques which the corresponding variables are equal to one) are vertex disjoint, and the objective function maximizes the size of the biclique k-clustering. Note that if $k < \min(n, m)$ then we need the additional constraint $\sum_{B \in \mathcal{B}} x_B \leq k$ in ($\mathrm{P}^2_{\mathrm{MBC}}$). We decided not include this constraint in the formulation because we do not need it in the application we are interested in.

Note that in ($\mathrm{P}^2_{\mathrm{MBC}}$) there is no symmetry. But, on the other hand, the number of variables of ($\mathrm{P}^2_{\mathrm{MBC}}$) is exponential in the size of G. Thus, the approach we propose here is to develop a column generation algorithm (see [2] for a nice survey on the object) to solve the linear relaxation of ($\mathrm{P}^2_{\mathrm{MBC}}$). To this end we need to investigate the corresponding pricing problem (i.e. find a variable x_B with negative reduced cost or give a proof that no such variable exists). First, consider the dual linear program of the linear relaxation of ($\mathrm{P}^2_{\mathrm{MBC}}$), where the constraint $x_B \in \{0,1\}$ is replaced by $0 \leq x_B \leq 1$.

$$
\begin{aligned}
(\mathrm{D}^2_{\mathrm{MBC}}) \qquad & \min \sum_{v \in V} \alpha_v \\
& \text{s.t.} \sum_{v \in V_B} \alpha_v \geq s(B), \text{ for all } B \in \mathcal{B} && (6) \\
& \alpha_v \geq 0, \qquad \text{for all } v \in V && (7)
\end{aligned}
$$

For a given biclique B of G, the reduced cost of the variable x_B is given by $\bar{x}_B := \sum_{v \in V_B} \alpha_v - s(B)$. Note that to solve the pricing problem we can find a variable x_B with minimum reduced cost. So, now we introduce a formulation for the pricing problem. We define the variables $w_{uv} \in \{0,1\}$ for all $uv \in E$ and $z_v \in \{0,1\}$ for all $v \in V$ with the following interpretation: $w_{uv} = 1$ if and only if the edge uv is inside the biclique, and $z_v = 1$ if and only if the vertex v is inside the biclique. Thus, the pricing problem for the variables x_B can be formulated as follows (where $\delta(v)$ denotes the vertices adjacent to v).

$$
\begin{aligned}
(\mathrm{P}_{\mathrm{MRC}}) \qquad & \min \sum_{v \in V} \alpha_v z_v - \sum_{uv \in E} w_{uv} \\
& \text{s.t. } w_{uv} + w_{pq} \leq 1, \qquad \text{for all } (uv, pq) \in \mathcal{I} && (8) \\
& z_v \geq w_{uv}, \quad \text{for all } u \in \delta(v) && (9) \\
& w_{uv}, z_v \in \{0,1\}, \text{ for all } uv \in E \text{ and } v \in V && (10)
\end{aligned}
$$

Constraints (9) guarantee that if an edge uv is chosen (i.e. $w_{uv} = 1$) then the vertices u and v are chosen as well (i.e. $z_u = z_v = 1$). Note that if $\alpha_v = 0$ for all $v \in V$, then we can remove (9) from ($\mathrm{P}_{\mathrm{MRC}}$), and also observe that the formulation ($\mathrm{P}_{\mathrm{MRC}}$) without the constraints (9) is equivalent to ($\mathrm{P}_{\mathrm{MBC}}$) for $k = 1$. Thus, to solve ($\mathrm{P}_{\mathrm{MRC}}$) in general is NP-hard. This does not give an efficient algorithm to solve the pricing problem. But, as we show in the computational results, we are able to solve instances of medium size in reasonable time.

3 Computational experiments

We implement three heuristics for the BkCP. The first one is a greedy algorithm which finds a maximum edge biclique $B = (V_B, E_B)$ in G and then apply

the same process to the subgraph induced by $V \setminus V_B$, and so forth. The second heuristic uses the optimal solution of the linear relaxation of (P^2_{MBC}) to define weights in the edges of E in the following way: (1) set the weight of each edge to zero; (2) for each nonzero variable x_B of the solution add the value of x_B to the weight of each edge in E_B. The final step is to run the greedy heuristic, as described above, but using these weights in the objective function, instead of defining unitary weight for every edge. The last heuristic solves (P^2_{MBC}) restricted to the columns generated to solve its linear relaxation (i.e. all not generated columns are fixed to 0). We denote these three heuristics by Hr1, Hr2 and Hr3, respectively.

We organize the results of the computational experiments with random generated instances in table 3.1. Each line corresponds to the result of ten different instances with the same size. The first three columns have the size of the input (i.e. $n = |R|$, $m = |C|$ and $|E|$, respectively). The 4th column has the density of the graph (i.e. $(100|E|)/(nm)$). The next four columns have the average time spent to solve the linear relaxation of (P^2_{MBC}) and the average time spent to run the three heuristics, respectively. The next three columns have the average gap between the value of the linear relaxation of (P^2_{MBC}) and the solutions found by the three heuristics. The column "Best" has the average gap between the value of the linear relaxation of (P^2_{MBC}) and the best feasible solution found. The last column has the percentage of solutions which are optimal. We used *Gurobi 2.0* [4] as the linear and mixed integer programming solver, and the machine configurations are the following: 8 processors Intel® Xeon® E5440 (2.83GHz) and 32GB of RAM memory.

Table 3.1: computational experiments with random generated instances.

Size of the input				Average time				Average gap				Opt. sol.
n	m	$\|E\|$	Density	LP	Hr1	Hr2	Hr3	Hr1	Hr2	Hr3	Best	% found
11	12	100	75.76%	136s	0s	0s	0s	7.65%	1.25%	1.71%	1.03%	70%
14	14	100	51.02%	245s	0s	0s	0s	7.53%	2.76%	1.20%	0.72%	70%
20	20	100	25.00%	605s	0s	0s	0s	15.65%	2.67%	1.02%	0.51%	80%
14	14	150	76.53%	805s	0s	0s	0s	7.24%	2.05%	1.90%	1.42%	70%
17	17	150	51.90%	1575s	0s	0s	0s	10.62%	2.85%	3.41%	2.66%	30%
24	25	150	25.00%	5675s	1s	0s	0s	20.90%	2.58%	0.72%	0.54%	70%
16	16	200	78.12%	7467s	0s	0s	0s	7.84%	1.26%	3.70%	1.26%	50%
20	20	200	50.00%	7047s	5s	1s	0s	12.46%	5.73%	5.30%	4.66%	10%
28	28	200	25.51%	21959s	9s	3s	0s	26.76%	5.57%	4.29%	3.48%	10%

As shown in table 3.1 our approach was able to solve random generated instances of medium size in reasonable time. The "average gap" shows that the gap between the linear relaxation of (P^2_{MBC}) and the optimal solution is

quite tight (in many cases the gap is zero, as shown in the last column). But, the gap increased with the size of the input. The "average time" shows that the three heuristics are very fast, but the heuristics Hr2 and Hr3 depend on the solution of the LP, which takes a long time to be obtained. From table 3.1 we can also conclude that the density of the graph plays an important role in the running time needed to solve the LP and also in the behavior of the greedy heuristic (Hr1). Intuitively, the problem becomes harder when the graph is sparse (except in the cases where the graph is so sparse that the solution is trivial) because the number of incompatible edges increases in this case.

In table 3.2 we show the computational results with instances that come from the application in bioinformatics.

Table 3.2: computational experiments with instances that come from the application.

Size of the input				Time spent				Gap			
n	m	$\|E\|$	Density	LP	Hr1	Hr2	Hr3	Hr1	Hr2	Hr3	Best
16	11	101	57.39%	185s	0s	0s	0s	6.38%	2.04%	2.04%	2.04%
26	13	228	67.46%	9966s	1s	0s	0s	0.96%	5.00%	0.96%	0.96%

As shown in table 3.2 our approach obtained a quite tight gap in both instances. The second row shows that the greedy heuristic (Hr1) obtained a very good solution in a very short running time.

4 Conclusion remarks and future work

We presented in this paper an application of the BkCP in bioinformatics and two integer linear formulations for the the problem. The first formulation enabled us to discuss the approximability of the problem (we presented a simple 2-approximation algorithm for the problem, which is similar to the LP rounding algorithm for the vertex cover problem). We introduced a second formulation and proposed a column generation algorithm for solving its linear relaxation. Our computational experiments (with random generated instances and with instances that come from the application in bioinformatics) showed that our approach can be effectively used to solve medium size instances.

As discussed in section 2, the BkCP is a special case of the vertex cover problem (or equivalently, the independent set problem). But it is still not clear if we can take some advantage of this special case. The computation experiments showed that our column generation algorithm was able to get a tight bound for the problem, and it may suggest that if we try to solve the problem in a branch-and-price algorithm we could get the optimal solution in almost the same running time. Thus, there are many possibilities for future

work on this problem.

References

[1] *Theory and Applications of Models of Computation*, chapter Inapproximability of Maximum Weighted Edge Biclique and Its Applications. Springer Berlin, 2008.

[2] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operational Research*, (3):316–329, 1998.

[3] P. Berman and G. Schnitger. On the complexity of approximating the independent set problem. *Information and Computation*, 96(1):77 – 94, 1992.

[4] R. Bixby, Z. Gu, and E. Rothberg. *Gurobi Optimizer Reference Manual*, 2009. http://www.gurobi.com/html/doc/refman/index.html.

[5] M. Dawande, P. Keskinocak, and S. Tayur. On the biclique problem in bipartite graphs. Technical report, Carnegie-Mellon University, 1997.

[6] Nathalie Faure, Philippe Chrétienne, Eric Gourdin, and Francis Sourd. Biclique completion problems for multicast network design. *Discrete Optimization*, 4(3-4):360 – 377, 2007.

[7] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[8] R. A. Notebaart, B. Teusink, R. J. Siezen, and B. Papp. Co-regulation of metabolic genes is better explained by flux coupling than by network distance. *PLoS Comp. Biol.*, 2008.

[9] R. Peeters. The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, pages 651–654, 2003.

[10] S. Pérès, M. B. Aimar, and J. P. Mazat. Pathway classification of tca cycle. In *IEE Proc. Systems Biology.*

[11] M. Terzer and J. Stelling. Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics*, 24(19):2229–2235, 2008.