A Branch-and-Price Approach to the k-Clustering Minimum Biclique Completion Problem

Stefano Gualandi [a,*] Francesco Maffioli [a] Claudio Magni [b]

[a] *Politecnico di Milano, Dipartimento di Elettronica e Informazione Piazza Leonardo da Vinci 32, 20133 Milano, Italy*

[b] *SAS Institute, Analytic Innovation Center via Darwin 20-22, 20143, Milano*

Key words: Biclique, Branch-and-Price, Meta-heuristic

1 Introduction

Given a bipartite graph $G = (S, T, E)$, the k-clustering Minimum Biclique Completion Problem (k-MinBCP) consists of finding k bipartite subgraphs (clusters), such that each vertex i of S appears in exactly one subgraph, every vertex j in T appears in each cluster in which at least one of its neighbors appears, and the total number of edges that would complete each subgraph into a complete bipartite subgraph, i.e., a biclique, is minimized. This problem was introduced in [1], as an application of the problem of bundling channels in multicast transmissions. k-MinBCP is NP-Hard, and its approximability, to the best of our knowledge, remains unknown. In the literature, k-MinBCP is tackled with two approaches: in [1], it is solved with an Integer Programming approach, a Bilinear Programming formulation and its standard linearization; and in [2], it is solved with an hybrid Constraint Programming–Semidefinite programming approach.

In this work, we present a Branch-and-Price algorithm that embeds a new meta-heuristic to find integer solutions, and a non-trivial branching rule. Computational results show that our algorithm outpeforms the state-of-the-art approaches to this problem.

* Corresponding author.
Email addresses: {maffioli,gualandi}@elet.polimi.it, claudio.magni@ita.sas.com (Claudio Magni).

2 Problem Formulation

In this work, we use a Column Generation formulation that is similar to the one proposed in [1]: the master problem is a set partitioning problem where each column represents the subset of vertices of S that induces a cluster t. The cost c_t of each cluster t is equal to the number of edges that would complete the corresponding subgraph into a biclique. Let λ_t be a 0–1 variable, equal to 1 if the cluster t is part of the solution, and 0 otherwise. Let c_t be the cost of the t-th cluster. Let $\mathcal{T}$ be the collection of every possible cluster, and let S_t be the subset of vertices of S that form the t-th cluster. The master problem formulation is as follows:

$$\begin{aligned}
\min \quad & \sum_{t \in \mathcal{T}} c_t \lambda_t & & (1)\\
\text{s.t.} \quad & \sum_{t \in \mathcal{T} | i \in S_t} \lambda_t = 1 & \forall\, i \in S \quad & (2)\\
& \sum_{t \in \mathcal{T}} \lambda_t = k & & (3)\\
& \lambda_t \in \{0,1\} & \forall\, t \in \mathcal{T} \quad & (4)
\end{aligned}$$

Constraints (2) is the partitioning constraints, one for each vertex in S. Constraint (3) is the cardinality constraint on the number of cluster to be selected. Let π_i and ν be the dual multipliers of constraints (2) and (3), respectively. Then, the pricing subproblem is the problem of finding a vertex– and edge–weighted biclique of negative reduced cost. The vertex weights are given by the dual multipliers π_i, while the edge weights gives the number of edges that would complete the subgraph into a biclique. The pricing subproblem is as follows:

$$\begin{aligned}
\min \quad & \sum_{\{i,j\} \in \bar{E}} z_{ij} - \sum_{i \in I} \pi_i x_i - \nu & & (5)\\
\text{s.t.} \quad & z_{ij} \geq x_i + x_l - 1 \quad \forall \{i,j\} \in \bar{E}, \forall \{l,j\} \in E & & (6)\\
& x_i, z_{ij} \in \{0,1\} \qquad \forall i \in I, \forall \{i,j\} \in \bar{E} & & (7)
\end{aligned}$$

Constraints (6) force the binary variable z_{ij} to be 1 if the corresponding edge is part of the biclique, and 0 otherwise. Note that an edge belongs to a biclique if it exists at least a pair of vertices i and l both in S such that a vertex $j \in T$ exists with $(i,j) \in \bar{E}$ and $(l,j) \in E$.

3 Branch-and-Price Implementation

Differently from [1], that uses the column generation only to compute lower bounds, we have embedded the column generation into a branch-and-price exact algorithm. The branch-and-price we have implemented is based on three key feature: (i) a Variable Neighborhood Search (VNS) heuristic that computes very efficient primal solution, providing tight upper bounds; (ii) a slightly different VNS heuristic that compute nearly-optimal solutions for the pricing subproblem, and (iii) a non-trivial branching rule.

The VNS heuristic used to find integer solutions, hence yielding upper bounds, explores basically three different neighborhoods: (i) moving a single vertex from one subgraph to another subgraph, (ii) swapping two vertices in two different subgraphs, and (iii) selecting two vertices in two different subgraphs and moving them into new subgraphs. In addition, after each cycle of VNS, we perform a search in the space of the unfeasible solutions, by augmenting by one the number of clusters. Then, a greedy procedure is used to recover a feasible solution. Although the search in the unfeasible space is very simple, it does improve the performance of our meta-heuristic.

In our Column Generation approach to k-MinBCP the bottleneck is the solution of the pricing subproblem. We have implemented two methods for solving the pricing problem. The first method is again a VNS heuristic very similar to the heuristic used to find integer solutions to k-MinBCP: we basically look for a single subgraph with negative reduced cost. Whenever the heuristic is unable to find a negative reduced cost solution, we use an integer programming approach to find any solution of negative reduced cost, not necessarily the solution of minimum cost.

The meta-heuristic and the Column Generation are used to obtain upper and lower bounds within our branch-and-price algorithm. Though many instances are solved at the root node (the upper bounds obtained with our VNS heuristic are equal to the lower bounds obtained by Column Generation), this is not always the case. Therefore, we have devised a branching rule that exploits the problem structure. Once a pair of vertices i and j of S appearing in a fractional solution of the restricted master problem are selected, the algorithm adds two branching constraints: either i and j must appear in the same cluster, or they cannot. In the first branch, we merge the two nodes in a single new node, obtaining a new instance of the same problem. In the second branch, in order to force two vertices to appear in different clusters, we add the corresponding constraints to the pricing subproblem.

4 Computational Results

We tested our branch-and-price algorithm on two classes of instances: the first set of instances consists of random bipartite graphs, and the second set of instances extracted by the MovieLens data set (`http://movielens.umn.edu`).

Table 1

|I|	**|J|**	**k**	**UB**	**LB**	**Opt**	**N**	**Time**	**SCIP**
15	15	3	**72**	**72**	**72**	0	**6.38**s	11s
15	15	4	**59**	**59**	**59**	0	**2.92**s	370s
15	15	5	**50**	**50**	**50**	0	**4.14**s	–
18	18	3	**109**	**109**	**109**	0	**26**s	137s
18	18	4	**96**	**96**	**96**	0	**20**s	–
18	18	5	**86**	85	**86**	8	**47**s	–
20	20	3	**156**	154	**156**	12	**275**s	–
20	20	4	*139*	137	**138**	8	**206**s	–
20	20	5	**123**	121	**123**	38	**175**s	–

The branch-and-price algorithm is implemented in COMET [4], using `lp_solve` as linear solver. Extensive computational results are reported in [3].

Table 1 shows a summary of the comparison of the computational results obtained with our branch-and-price algorithm and with the ILP solver SCIP. The table gives the results for the most challenging instances, that are instances randomly generated with $|S| = |T|$. The first three columns give the $|S|$, $|T|$, and the number of required cluster k. Then, the table reports the **UB** obtained with our VNS heuristic at the root node, the lower bound **LB** obtained via column generation at the root node, the optimal solution **Opt** obtained via Branch-and-Price, the number of branching nodes **N**, and the computation time in seconds. For the SCIP solver, we just report the computation time in seconds. Note that our branch-and-price algorithm is able to solve instances of dimension up to 20×20. In addition, we remark that previous work solved only instances up to 10×10 in [1] and 12×12 in [2].

References

[1] Faure, N., Chrétienne, P., Gourdin, E., Sourd, F.: Biclique completion problems for multicast network design. Discr. Optim. **4**(3) (2007) 360–377

[2] Stefano Gualandi. k-clustering minimum biclique completion via a hybrid CP and SDP approach. In *Proc Integration of AI and OR Techniques in CP for Combinatorial Optimization*, LNCS 5547, pages 87–101. Springer, 2009.

[3] Claudio Magni. Biclique completion problem: models and algorithms. Master Project, Politecnico di Milano, September 2009.

[4] Comet web site, Brown University, `http://comet-online.org`