European Journal of Operational Research 269 (2018) 834–843

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Towards effective exact methods for the Maximum Balanced Biclique Problem in bipartite graphs

Yi Zhou[a,b], André Rossi[b], Jin-Kao Hao[b,c,*]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[b] LERIA, Université d'Angers, 2 bd Lavoisier, Angers 49045, France
[c] Institut Universitaire de France, 1 rue Descartes, Paris 75231, France

ARTICLE INFO

Article history:
Received 14 January 2017
Accepted 7 March 2018
Available online 13 March 2018

Keywords:
Combinatorial optimization
Clique
Exact algorithms
Techniques for tight bounds
Mathematical formulation

ABSTRACT

The Maximum Balanced Biclique Problem (MBBP) is a prominent model with numerous applications. Yet, the problem is NP-hard and thus computationally challenging. We propose novel ideas for designing effective exact algorithms for MBBP in bipartite graphs. First, an Upper Bound Propagation (UBP) procedure to pre-compute an upper bound involving each vertex is introduced. Then we extend a simple Branch-and-Bound (B&B) algorithm by integrating the pre-computed upper bounds. Based on UBP, we also study a new integer linear programming model of MBBP which is more compact than an existing formulation (Dawande, Keskinocak, Swaminathan, & Tayur, 2001). We introduce new valid inequalities induced from the upper bounds to tighten these mathematical formulations for MBBP. Experiments with random bipartite graphs demonstrate the efficiency of the extended B&B algorithm and the valid inequalities generated on demand. Further tests with 30 real-life instances show that, for at least three very large graphs, the new approaches improve the computational time with four orders of magnitude compared to the original B&B.

1. Introduction

Given a bipartite graph $G = (U, V, E)$ with two disjoint vertex sets U, V and an edge set $E \subseteq U \times V$, a biclique $A \cup B$ (or (A, B)) is the union of two subsets of vertices $A \subseteq U$, $B \subseteq V$ such that $\forall i \in A$, $\forall j \in B$, $\{i, j\} \in E$. In other words, the subgraph induced by vertex set $A \cup B$ is a complete bipartite graph. If $|A| = |B|$, then biclique (A, B) is a balanced biclique. The Maximum Balanced Biclique Problem (MBBP) is to find a balanced biclique (A, B) of maximum cardinality. As $|A| = |B|$ holds for a balanced biclique (A, B), MBBP is then to find the maximum half-size balanced biclique. MBBP is a special case of the conventional maximum clique problem (Wu & Hao, 2015).

MBBP is a prominent model with a large range of applications, such as nanoelectronic system design (Al-Yamani, Ramsundar, & Pradhan, 2007; Tahoori, 2006), biclustering of gene expression data in computational biology (Cheng & Church, 2000) and PLA-folding in the VLSI theory (Ravi & Lloyd, 1988). In all these applications, the given graphs are bipartite graphs. In terms of computational complexity, the decision version of MBBP is NP-Complete (Alon, Duke, Lefmann, Rodl, & Yuster, 1994; Garey & Johnson, 1979), though the maximum biclique problem in bipartite graphs (without requiring $|A| = |B|$) is polynomially solvable by the maximum matching algorithm (Cheng & Church, 2000).

Considerable effort has been devoted to the pursuit of effective algorithms for MBBP in bipartite graphs, both theoretically and practically. Heuristic algorithms represent the most popular approach for MBBP, though they do not guarantee the optimality of the attained solutions. The majority of existing heuristic algorithms solve the equivalent maximum balanced independent set (a vertex set such that no two vertices are adjacent) problem in the complement bipartite graph, rather than directly seeking the maximum balanced biclique from the given graph. For example, several greedy heuristic algorithms were proposed, which apply vertex-deletion rules on the complement bipartite graph in the period from 2006 to 2014 (Al-Yamani et al., 2007; Tahoori, 2006; Yuan & Li, 2011; 2014), while an evolutionary algorithm combining structure mutation and repair-assisted restart was introduced in 2015 (Yuan, Li, Chen, & Yao, 2015).

On the other hand, according to our literature review, there are only two studies on exact algorithms. In Tahoori (2006), a recursive exact algorithm for searching a maximum balanced independent set with a given half-size in the complement graph was proposed. However, the computational time of this algorithm becomes prohibitive when the number of vertices of the given graph

* Corresponding author at: LERIA, Université d'Angers, 2 bd Lavoisier, Angers 49045, France.
E-mail address: jin-kao.hao@univ-angers.fr (J.-K. Hao).

https://doi.org/10.1016/j.ejor.2018.03.010
0377-2217/

exceeds (32,32). In McCreesh and Prosser (2014), a Branch-and-Bound (B&B) algorithm for MBBP (named BBClq) for general graphs (including non-bipartite graphs) was studied. The algorithm incorporates a clique cover technique for upper bound estimation (an equivalent technique of using graph coloring to estimate the upper bound for the maximum clique problem) and employs lex symmetry breaking techniques for general graphs. As far as we know, this algorithm is currently the best performing exact algorithm, even though the bounding technique and symmetry breaking techniques are only effective for non-bipartite graphs.

In addition to specifically designed exact algorithms, the general Integer Linear Programming (ILP) constitutes an interesting alternative for solving hard combinatorial problems such as MBBP. Commercial mixed ILP solvers, like IBM CPLEX, can even solve some hard instances which cannot be handled by other approaches. Meanwhile, the success of a ILP solver highly depends on the tightness of the mathematical formulation of the problem. For MBBP, an ILP formulation has been proposed in Dawande, Keskinocak, Swaminathan, and Tayur (2001), which is based on the complement graph. Another mathematical formulation that defines the constraints on the original graph was presented in Yuan et al. (2015). However, this formulation was not applicable for ILP solvers as it contains non-linear constraints.

In this work, we introduce new ideas for developing effective exact algorithms for MBBP, which help to solve very large MBBP instances from applications like social networks. Our main contributions can be summarized as follows.

First, we elaborate an Upper Bound Propagation (UBP) procedure inspired from Soto, Rossi, and Sevaux (2011), which produces an upper bound of the maximum balanced biclique involving each vertex in the bipartite graph. UBP propagates the initial upper bound involving each vertex and achieves an even tighter upper bound for each vertex. UBP is independent from the search procedure and is performed before the start of the search algorithm. An extended exact algorithm, denoted by ExtBBClq, is proposed by taking advantage of UBP to improve BBClq, the branch-and-bound algorithm introduced in McCreesh and Prosser (2014).

Second, we introduce a new and more compact formulation that requires a largely reduced number of constraints compared to the previous formulation presented in Dawande et al. (2001). In the previous model of Dawande et al. (2001), the number of constraints equals the number of edges in the complement bipartite graph, making it inapplicable to solve large real-life sparse graphs. The proposed model reduces the number of constraints to the number of vertices in the graph, which allows for dealing with very large instances. We also introduce new inequalities to tighten both previous and new formulations. Our computational results suggest that the new formulation and tightened inequalities improve the performance of the ILP solver CPLEX.

The remainder of the paper is organized as follows. Section 2 introduces the notations that will be used throughout the paper and Section 3 reviews the BBClq algorithm introduced in McCreesh and Prosser (2014). In Section 4, we present our Upper Bound Propagation procedure for upper bound estimation and explain how to use it to improve BBClq. In Section 5, we discuss the existing ILP formulation for MBBP and present our new ILP model. We also study how the upper bounds can lead to new valid inequalities to tighten the ILP formulations. Computational results and experimental analyses are presented in Section 6, followed by conclusions and future working directions.

2. Notations

Given a bipartie graph $G = (U, V, E)$ ($|U| \le |V|$ if not specifically stated), let $(A, B) \subseteq (U, V)$ be a balanced biclique of G (i.e., $|A| = |B|$). The half-size of the balanced biclique (A, B) is the cardinality of $|A|$

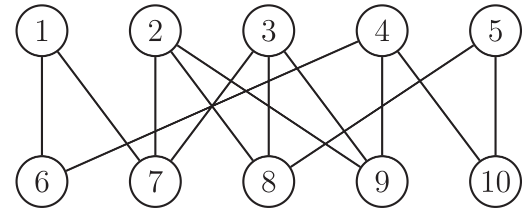

Fig. 1. A bipartite graph $G = (U, V, E)$, $U = \{1, 2, 3, 4, 5\}$, $V = \{6, 7, 8, 9, 10\}$.

(or $|B|$). For example, in Fig. 1, ({2, 3}, {7, 8}) is a balanced biclique of half-size of 2. For all $S \subseteq U \cup V$, $G[S]$ denotes the subgraph of G induced by S. Given a vertex i in G, the set of vertices adjacent to i is denoted by $N(i) = \{j : \{i, j\} \in E\}$ and $deg_G(i) = |N(i)|$ is the degree of vertex i. The upper bound involving vertex i, denoted by ub_i, is an upper bound of the half-size of the maximum balanced biclique containing vertex i. For example, in Fig. 1, a possible value for ub_1 could be 2, since $deg_G(1) = 2$.

3. Review of the BBClq algorithm

To our knowledge, the B&B algorithm BBClq presented in McCreesh and Prosser (2014) is the current best-performing exact algorithm for MBBP for general graphs. The algorithm is mainly inspired from the well-known algorithms (Segundo, Rodríguez-Losada, & Jiménez, 2011; Tomita & Kameda, 2007) for the maximum clique problem. Algorithm 1 shows the general search scheme of BBClq.

Algorithm 1: BBClq(G, A, B, C_A, C_B), the trimmed B&B algorithm for MBBP from McCreesh and Prosser (2014).

Input: Graph instance $G = (U, V, E)$, A, B – current sets that form a biclique, C_A, C_B – the sets of eligible vertices that can be added to A and B, respectively
Output: A maximum balanced biclique of G.

1 **if** $|A| > lb$ **then**
2 $\quad lb \leftarrow |A|$
3 $\quad$ Record current best biclique in (A^*, B^*)
4 **while** $C_A \neq \emptyset$ **do**
5 $\quad$ **if** $|A| + |C_A| \le lb$ **then**
6 $\quad\quad$ **return**
7 $\quad i \leftarrow branch\_vertex(C_A)$
8 $\quad C_A \leftarrow C_A \setminus \{i\}$
9 $\quad$ **if** $upper\_bound(A \cup \{i\}) > lb$ **then**
10 $\quad\quad A' \leftarrow A \cup \{i\}$
11 $\quad\quad C'_B \leftarrow C_B \cap N(i)$
12 $\quad\quad$ BBClq(G, B, A', C'_B, C_A)
13 **return** $make\_balance(A^*, B^*)$

BBClq recursively builds two sets A and B such that (A, B) forms a biclique. It maintains a candidate set C_A (C_B) that includes vertices which are eligible to join A (B) while ensuring that (A, B) is a biclique (i.e., $C_A = \bigcap_{i \in B} N(i)$, $C_B = \bigcap_{i \in A} N(i)$). Initially, the algorithm sets lb, the global lower bound on the maximum biclique half-size to 0 and starts the search by calling BBClq(G, $\emptyset$, $\emptyset$, U, V).

At each recursive call to BBClq, a vertex i (called branch vertex) is moved from C_A (lines 7 and 8). The algorithm then considers the branches (possibilities) of $i \in A$ (lines 9–12) and $i \notin A$ in the next `while` loop. The bounding procedure (line 9) prunes the branch of $i \in A$ if the upper bound after estimation in this context is not larger than the global lower bound. The upper bound estimating method, which is classically a key point concerning the