Mathematical Programming Models and Their Relaxations for The Minimum Hub Cover Problem

Belma Yelbay, Ş. İlker Birbil and Kerem Bülbül
Sabancı University, Manufacturing Systems and Industrial Engineering, Orhanlı-Tuzla, 34956 Istanbul, Turkey.
byelbay@sabanciuniv.edu, sibirbil@sabanciuniv.edu, bulbul@sabanciuniv.edu

Abstract: The minimum hub cover is a new $\mathcal{NP}$-hard optimization problem that has been recently introduced to the literature in the context of graph query processing. Although a standard set covering formulation has been given, alternate models have not been studied before. This paper fills in that gap and proposes several new mathematical programming models. We give two binary integer programming formulations as well as a quadratic integer programming formulation. Our relaxation for the quadratic model leads to a semidefinite programming formulation. A solution to the minimum hub cover problem can be obtained by solving the relaxations of the proposed models and then rounding their solutions. We introduce two rounding algorithms after solving the linear programming and semidefinite programming relaxations, respectively. We also implement two well-known rounding algorithms designed for the set covering problem. Computational results demonstrate that our methods outperform those solutions obtained with the standard set covering formulations.

Keywords: minimum hub cover, linear programming, semidefinite programming, rounding heuristics

1. Introduction.

The *minimum hub cover* (MHC) problem is a combinatorial optimization problem. One well-known application of the MHC problem is query processing over graph databases. For readers not familiar with the subject, querying a graph database refers to searching the structural similarity between two graphs. For instance, Figure 1 demonstrates a query and a database graph of two molecular compounds. Note that the database graph on the right has a subgraph, which is structurally identical to the query graph on the left. Therefore, carrying out a query with this subgraph returns a positive response.

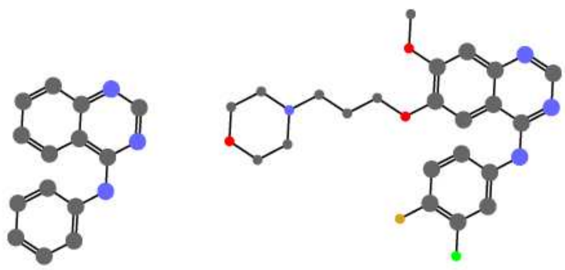


Figure 1: A query and a database graph of a molecular compound (Kawabata, 2014)

The idea of hub covers has been first introduced as a new graph representation model to expedite graph queries (Jamil, 2011). With this representation, a graph database can be represented by a small subset of graph vertices. Searching over only that subset of vertices decreases the response time of a query and increases the efficiency of graph query processing. Yelbay et al. (2013) introduce the problem of finding a subgraph including the minimum number of vertices as an optimization problem referred to as the MHC problem. Rivero and Jamil (2014a,b) demonstrate that searching a query over the vertices in MHC increases the efficiency of query processing and surpasses several other search methods.

The objective of the MHC problem is to cover all edges of a graph by selecting the minimum number of vertices. Selecting a vertex covers both its incident edges and the edges between its adjacent neighbors. For instance in Figure 2, vertex g covers edges $(f,g),(e,g),(c,g)$ and (c,e). The formal definition of the problem follows.

Definition 1.1 *Let $G=(V,E)$ be an undirected graph, where V is the set of vertices and E is the set of edges. A subset of the vertices, $HC \subseteq V$ is a* hub cover *of G if for every edge $(i,j) \in E$, either $i \in HC$ or $j \in HC$ or there exists a vertex k such that $(i,k) \in E$ and $(j,k) \in E$ with $k \in HC$. The* MHC problem *is about finding a hub cover that has the minimum number of vertices.*

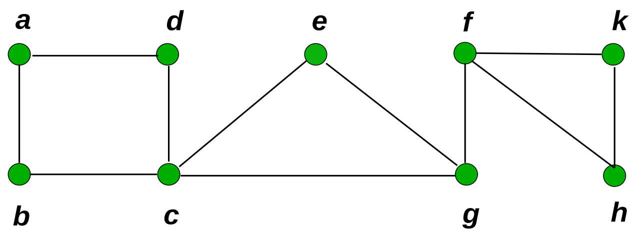


Figure 2: A sample graph for the minimum hub cover problem

The MHC problem is known to be $\mathcal{NP}$-hard in the strong sense (Yelbay et al., 2013). It remains $\mathcal{NP}$-hard even when restricted to planar graphs (Yelbay et al., 2014). Therefore, one may resort to algorithms that yield good, preferably near-optimal solutions fairly quickly. In the first category, we have approximation algorithms with proven performance bounds. To this date, the only approximation result for the MHC problem has been given by Yelbay et al. (2014) and this result applies exclusively to planar graphs. In the second category, we can consider fast heuristic approaches. Yelbay et al. (2013) implement two greedy algorithms and one mathematical programming-based heuristic. Their results indicate that the mathematical programming-based heuristic outperforms the greedy algorithms in terms of optimality gap.

In this paper, we focus on new mathematical programming relaxations and rounding heuristics. Our motivation is two-fold: First, with the advances in computational machinery, very large instances of the mathematical programming relaxations can be solved efficiently. Second, strong formulations coupled with rounding-based relaxation heuristics provide good feasible solutions in a reasonable time. With this motivation, we make the following research contributions: (i) We introduce new binary programming models along with a quadratic integer programming model. The relaxations of the binary models are linear programs whereas the relaxation of the last model is a semidefinite program. (ii) We also present several rounding heuristics to accompany the proposed relaxations. (iii) We conduct an extensive computational study to illustrate the empirical performances of the rounding heuristics.

2. Mathematical Programming Models. Our first formulation is a set covering formulation of MHC introduced in (Yelbay et al., 2013). If an edge corresponds to an item, and a set is defined for each vertex whose elements are the edges covered by that vertex, then the connection between the set covering and MHC can easily be established. This mathematical programming formulation is given as follows:

$$\text{minimize} \quad \sum_{j \in V} x_j, \tag{1}$$

$$\text{subject to} \quad x_i + x_j + \sum_{k \in \mathcal{K}^{(i,j)}} x_k \geq 1, \qquad (i,j) \in E, \tag{2}$$

$$x_j \in \{0,1\}, \qquad j \in V. \tag{3}$$

Here, x_j is a binary variable, which is equal to 1 when vertex j is selected. For $(i,j) \in E$, $\mathcal{K}^{(i,j)}$ denotes all those vertices $k \in V$ such that $(i,k) \in E$ and $(j,k) \in E$. The objective function (1) minimizes the number of selected vertices. Constraints (2) ensure that every edge is covered by at least one vertex in the hub cover. Finally, constraints (3) enforce binary restrictions on the variables.

The well-known minimum vertex cover problem is a special case of the MHC problem when the cardinality of the set $\mathcal{K}^{(i,j)}$ is zero; that is, $|\mathcal{K}^{(i,j)}| = 0$. The minimum vertex cover problem has a complementary problem formulation known as the maximum independent set problem. This relationship inspired us to introduce a new optimization problem, which we call the maximum triangular set (MTS) problem. The formal definition of MTS follows.

Definition 2.1 *For a given graph $G = (V, E)$, $TS \subseteq V$ is a triangular set if and only if for every edge (i,j) at most $|\mathcal{K}^{(i,j)}| + 1$ of the vertices in $\bar{\mathcal{K}}^{(i,j)} := \mathcal{K}^{(i,j)} \cup \{i,j\}$ are also in TS. The MTS problem is about finding a triangular set which has the maximum number of vertices.*

A careful reader may notice that MTS is equivalent to MHC in the sense that the solution of one problem will yield a solution for the other one. This is in fact the case as we prove in Lemma 2.1.

Lemma 2.1 *In any graph $G = (V, E)$, HC is a hub cover in G if and only if $V \setminus HC$ is a triangular set or TS is a triangular set in G if and only if $V \setminus TS$ is a hub cover.*

Proof. Suppose TS is a triangular set on G. Then for any edge (i,j), at most $|\mathcal{K}^{(i,j)}| + 1$ of the vertices in $\bar{\mathcal{K}}^{(i,j)}$ are in TS. Since the number of vertices that can cover edge (i,j) is equal to $|\mathcal{K}^{(i,j)}| + 2$, at least one of the vertices in S must be in $V \setminus TS$. Thus, $V \setminus TS$ must be a hub cover. Conversely, suppose $V \setminus TS$ is a hub cover. Then, at least one of the vertices in $\bar{\mathcal{K}}^{(i,j)}$ must be in $V \setminus TS$ so that $V \setminus TS$ is a hub cover. That is for each edge, the cardinality of the subset of $\bar{\mathcal{K}}^{(i,j)}$ included in TS is less than $|\mathcal{K}^{(i,j)}| + 2$ and hence, TS is a triangular set. □

The mathematical programming model of the MTS problem is given by

$$\text{maximize} \quad \sum_{j \in V} x_j, \tag{4}$$

$$\text{subject to} \quad x_i + x_j + \sum_{k \in \mathcal{K}^{(i,j)}} x_k \leq |\mathcal{K}^{(i,j)}| + 1, \qquad (i,j) \in E, \tag{5}$$

$$x_j \in \{0,1\}, \qquad j \in V, \tag{6}$$

where x_j is a binary variable that is equal to 1 when vertex j is selected. The objective function (4) maximizes the number of selected vertices. Constraints (5) ensure that the solution is a TS as defined in Definition 2.1. The final set of constraints (6) ensure the integrality of the binary variables.

Our last reformulation is a quadratic integer program, where each term is a product of two binary variables. This formulation shall form the basis of the semidefinite programming relaxation that we will introduce in Section 3:

$$\text{minimize} \quad \sum_{j \in V} (1 + y_0 y_j)/2, \tag{7}$$

$$\text{subject to} \quad (y_0 - y_i)(y_0 - y_j) + (2y_0 - y_i - y_j) \sum_{k \in \mathcal{K}^{(i,j)}} (y_0 - y_k) \leq 8|\mathcal{K}^{(i,j)}|, \qquad (i,j) \in E, \tag{8}$$

$$y_j \in \{+1, -1\}, \qquad j \in V \cup \{0\}. \tag{9}$$

The optimal solution of the MHC problem is given by those vertices $j \in V$ such that $y_j = y_0$. The set of constraints (8) is obtained after simplifying the following constraint for each $(i,j) \in E$.

$$(y_0 - y_i)(y_0 - y_j) + \sum_{k \in \mathcal{K}^{(i,j)}} (y_0 - y_i)(y_0 - y_k) + \sum_{k \in \mathcal{K}^{(i,j)}} (y_0 - y_j)(y_0 - y_k) \leq 8|\mathcal{K}^{(i,j)}|. \tag{10}$$

The following example illustrates relation (10) on a clique of three vertices.

Example 2.1 *Suppose that we consider the clique consisting of the vertices i, j, and k. Since in a clique, every two vertices are connected by an edge, the constraint (10) for edge (i,j) becomes*

$$(y_0 - y_i)(y_0 - y_j) + (y_0 - y_i)(y_0 - y_k) + (y_0 - y_j)(y_0 - y_k) \leq 8. \tag{11}$$

Given $y_i \in \{-1, +1\}$, this constraint ensures that the solution $y_0 \neq y_i = y_j = y_k$ is infeasible. Thus, at least one of the three vertices is selected.

Clearly, the mathematical programming models that we have introduced in this section are very difficult to solve to optimality. Nonetheless, the relaxations of these problems can be solved efficiently. These relaxations have two uses: (i) Their optimal objective function values can be used to give bounds. Then, these bounds could be used to increase the efficiency of exact methods. (ii) The optimal solutions of the relaxations can be used to obtain feasible solutions for the original problem. In certain cases, these solutions can even play a role to give approximation bounds. These relaxations as well as related rounding heuristics are given in the next section.

3. Relaxations and Rounding Heuristics. The mathematical programming relaxation is a modeling approach to replace a difficult problem with an easier one. In most cases, the solutions of the relaxed models are not feasible for the original problem. Then, one needs to resort to rounding heuristics. In this section, we focus on linear and semidefinite programming relaxations along with several rounding heuristics.

3.1 Linear Programming Relaxation. The linear programming (LP) relaxation is obtained simply by replacing the binary constraints on the variables with the inequalities $0 \leq x_j \leq 1$. We relax the integrality constraint in models (1)-(3) and (4)-(6) and obtain LP models for the MHC and the MTS problems that we shall refer to as LP1 and LP2, respectively.

In this section, we first introduce a new rounding algorithm for the MTS problem. As previously mentioned, our first model (1)-(3) is a special case of the set covering problem. Therefore, we also customize two other rounding algorithms that were originally proposed to solve the set covering problem.

Primal Rounding Algorithm for the MTS Problem (*PRMTS*): The algorithm uses the optimal solution of LP2. The pseudo-code of the algorithm is given in Algorithm 1. In line 3, we solve LP2 and obtain the optimal solution, x^*. We select the kth variable, which is the largest component in x^* in line 5. Then, kth vertex is selected and the right hand side of the constraints including that vertex is decreased by 1. The algorithm continues to select the next vertex with largest value as long as none of the constraints is violated.

Algorithm 1 Primal Rounding Algorithm for the MTS Problem

1: $x_j = 0, \quad \forall j \in V$
2: $y_{ij} \leftarrow |\mathcal{K}^{(i,j)}| + 1$ ▷ Right hand side of (5)
3: $x^* \leftarrow$ Solve LP relaxation of (4)-(6)
4: **for** $i = 1$ **to** $|V|$ **do**
5:     **pick**, the kth variable, which is the ith largest component in x^*
6:     **find** the set of constraints $C \subseteq E$ including kth variable
7:     **if** $y_{ij} > 0, \quad \forall (i,j) \in C$ **then**
8:         $x_k \leftarrow 1$
9:         $y_{ij} \leftarrow y_{ij} - 1, \quad \forall (i,j) \in C$
10:     **end if**
11: **end for**
12: **Return** x

Primal Rounding Algorithm for the MHC Problem ($PRMHC$): Algorithm 2 is adapted from a set covering algorithm proposed by Hochbaum (1982). The algorithm uses the optimal solution of LP1 denoted by x^*. Any component of x^* with value greater than or equal to $1/f$ is set to 1. In the hub cover formulation, f is the maximum number of vertices that can cover an edge. This approach is guaranteed to yield a feasible solution for MHC. Suppose $PRMHC$ does not yield a feasible solution, then there exists at least one constraint for edge (i,j) such that $x_j^* < 1/f$ for all $j \in \bar{\mathcal{K}}^{(i,j)}$. If this is the case, then $x_i^* + x_j^* + \sum_{k \in \mathcal{K}^{(i,j)}} x_k^* < 1$ because $|\bar{\mathcal{K}}^{(i,j)}| \leq f$. This contradicts our assumption that x^* is the optimal solution of LP1.

Algorithm 2 Primal Rounding Algorithm for the MHC Problem

$x_j = 0, \quad \forall j \in V$
$x^* \leftarrow$ Solve LP relaxation of (1)-(3)
for all $j \in V$ **do**
    if $x_j^* \geq 1/f$ **then**
        $x_j \leftarrow 1$
    end if
end for
Return x

Dual Rounding for the MHC Problem ($DRMHC$): The algorithm proposed by Hochbaum (1982) for the set covering problem is applied to obtain an integral MHC. It uses the optimal solution of the dual problem given by

$$\text{maximize} \quad \sum_{(i,j)\in E} y_{(i,j)}, \tag{12}$$

$$\text{subject to} \quad \sum_{(i,j)\in E} y_{(i,j)} + \sum_{(j,i)\in E} y_{(j,i)} + \sum_{j\in\mathcal{K}^{(i,k)}} y_{(i,k)} \leq 1, \qquad j \in V, \tag{13}$$

$$y_{(i,j)} \geq 0, \qquad (i,j) \in E, \tag{14}$$

where $y_{(i,j)}$ is a dual variable corresponding to the coverage constraint for edge (i,j). The steps of the algorithm are given in Algorithm 3. The optimal solution of (12)-(14) is denoted by y^*. The main idea of the algorithm is to set the primal variable to 1 whenever the corresponding dual constraint is tight.

Algorithm 3 Dual Rounding Algorithm for the MHC Problem

```
x_j = 0  ∀j ∈ V
Solve LP relaxation of (12)-(14)
for all j ∈ V do
    if Σ_{(i,j)∈E} y*_(i,j) + Σ_{(j,i)∈E} y*_(j,i) + Σ_{j∈K^(i,k)} y*_(i,k) = 1 then
        x_j ← 1
    end if
end for
Return x
```

3.2 Semidefinite Programming Relaxation. Semidefinite programming (SDP) is about optimizing a linear function of a symmetric matrix over the cone of positive semidefinite matrices. LP is a special case of SDP. Today, many $\mathcal{NP}$-hard optimization problems have semidefinite relaxations. The important point is that very good approximation bounds can be obtained after solving the SDP relaxations of hard combinatorial problems (Goemans and Williamson, 1995; Halperin, 2002; Karakostas, 2005).

Before introducing the SDP relaxation, we first remove the integrality constraint from (7)-(9) and obtain

$$\text{minimize} \quad \sum_{j\in V}(1 + y_0 y_j)/2, \tag{15}$$

$$\text{subject to} \quad (y_0 - y_i)(y_0 - y_j) + (2y_0 - y_i - y_j) \sum_{k\in\mathcal{K}^{(i,j)}} (y_0 - y_k) \leq 8|\mathcal{K}^{(i,j)}|, \qquad (i,j) \in E, \tag{16}$$

$$y_j^2 = 1, \qquad j \in V \cup \{0\}. \tag{17}$$

We next introduce the matrix variable $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$, where $\mathbf{y}$ is the vector consisting of components y_0 and $y_i, i \in V$. We also define $\mathbf{A} \bullet \mathbf{B}$:=trace$(\mathbf{A}^T\mathbf{B})$. Using now this notation, we can give the following equivalent formulation:

$$\begin{aligned} \text{minimize} \quad & \mathbf{C} \bullet \mathbf{Y} && && (18) \\ \text{subject to} \quad & \mathbf{A}^{(i,j)} \bullet \mathbf{Y} \leq 8|\mathcal{K}^{(i,j)}|, && (i,j) \in E, && (19) \\ & \text{diag}(\mathbf{Y}) = \mathbf{e}, && && (20) \\ & \mathbf{Y} \succeq 0, && && (21) \\ & \text{rank}(\mathbf{Y}) = 1, && && (22) \end{aligned}$$

where $\mathbf{C}$ and $\mathbf{A}^{(i,j)}$ are symmetric matrices, $\mathbf{e}$ is the vector of ones and $\mathbf{Y} \succeq 0$ means that the matrix $\mathbf{Y}$ is positive definite. Before specifying $\mathbf{C}$ and $\mathbf{A}^{(i,j)}$, let us relax the constraint (22) and obtain the SDP relaxation of the MHC problem given by

$$\begin{aligned} \text{minimize} \quad & \mathbf{C} \bullet \mathbf{Y} && && (23) \\ \text{subject to} \quad & \mathbf{A}^{(i,j)} \bullet \mathbf{Y} \leq 8|\mathcal{K}^{(i,j)}|, && (i,j) \in E, && (24) \\ & \text{diag}(\mathbf{Y}) = \mathbf{e}, && && (25) \\ & \mathbf{Y} \succeq 0. && && (26) \end{aligned}$$

The symmetric matrices in the SDP relaxation are defined as follows: Let C_{mn} denote the components of the matrix $\mathbf{C}$. Then,

$$C_{mn} = \begin{cases} 1/4, & \text{if } m = 0 \text{ and } n \in V; \\ 1/4, & \text{if } m \in V \text{ and } n = 0; \\ 0, & \text{otherwise.} \end{cases}$$

When it comes to the matrix $\mathbf{A}^{(i,j)}$, we observe that

$$(y_0 - y_i)(y_0 - y_j) = \mathbf{M} \bullet \mathbf{y}\mathbf{y}^T,$$

where $\mathbf{M}$ is a symmetric matrix and its nonzero components are given by

$$M_{00} = 1, \quad M_{0i} = M_{i0} = M_{0j} = M_{j0} = -1/2, \quad M_{ij} = M_{ji} = 1/2.$$

Note for a given $(i,j) \in E$ that the constraint (10) is constructed by summing up matrices like $\mathbf{M}$ above. Consequently, the matrix $\mathbf{A}^{(i,j)}$ also becomes a symmetric matrix.

Formally, we write $\mathbf{Y} = \mathbf{V}^T\mathbf{V}$, where the columns of $\mathbf{V}$ are given by $\mathbf{v}_m, m \in V \cup \{0\}$. Then, we obtain,

$$\begin{aligned} \text{minimize} \quad & \sum_{m,n} C_{mn} \mathbf{v}_m^T \mathbf{v}_n && && (27) \\ \text{subject to} \quad & \sum_{m,n} \mathbf{A}_{mn}^{(i,j)} \mathbf{v}_m^T \mathbf{v}_n \leq 8|\mathcal{K}^{(i,j)}|, && (i,j) \in E, && (28) \\ & \mathbf{v}_m^T \mathbf{v}_n = 1, && m \in V \cup \{0\}, && (29) \\ & \mathbf{v}_m \in \mathbb{R}^{|V|+1}, && m \in V \cup \{0\}. && (30) \end{aligned}$$

SDP Rounding Algorithm for the MHC Problem (*RSDP*): We implemented a rounding algorithm inspired from another method proposed for the minimum vertex cover problem (Halperin, 2002). This rounding method uses the optimal solution of the SDP relaxation, $\mathbf{v}^*$, and returns the set $S = \{j \in V | \mathbf{v}_0^{*T}\mathbf{v}_j^* \geq 0\}$ as an approximate solution. This solution is not necessarily a feasible hub cover but the number of uncovered edges is much less with respect to the number of covered edges. On the other hand, we observe that the method results in many redundant nodes in the hub cover. Alternatively, we propose Algorithm 4, which obtains $S = \{j \in V | \mathbf{v}_0^{*T}\mathbf{v}_j^* > 0\}$ and then repairs the feasibility by iteratively selecting a vertex $i \in V \setminus S$ which covers the highest number of uncovered edges until all edges are covered.

Algorithm 5 SDP Algorithm for the MHC Problem

$x_j = 0 \quad \forall j \in V$
$\mathbf{v}^* \leftarrow$ Solve SDP relaxation of (27)-(30)
for all $j \in V$ **do**
    if $\mathbf{v}_0^{*T}\mathbf{v}_j^* > 0$ **then**
        $x_j \leftarrow 1$
    end if
end for
Find the set of uncovered edges $U \subseteq E$
while $|U| > 0$ **do**
    Find the vertex j that covers the maximum number of edges in U
    $x_j \leftarrow 1$
    $U = U \setminus (i, k) \quad \forall (i, k)$ covered by vertex j
end while
Return x

4. Computational Study.

In this section, we conduct a set of experiments to test the performance of the LP and SDP relaxations as well as the rounding algorithms using the optimal solutions of those relaxations. We first define our problem classes and experimental set-up and then discuss our results.

4.1 Problem Classes and Experimental Set-up.

Here, we list our problem classes and their descriptions. Our data set includes a total of 210 graphs (30 graphs from each class) with known optimal solutions. The first five classes are from a well-known graph database by Santo et al. (2003) and the others are synthetically generated graphs used in various application areas. The LP and SDP relaxations are obtained by MATLAB 2010b. To solve the SDP relaxation, we used the SDPA-M solver which is a MATLAB interface for the semidefinite programming algorithm (SDPA) solver developed by Kojima et al. (2005). The solver is developed to solve small and medium size semidefinite programming models. Therefore, our problem set includes only small to medium size instances. The number of vertices and edges range from 20 to 1000.

- ⋄ **(a) Random graphs:** The randomly generated graphs with varying densities.
- ⋄ **(b) Bounded valence graphs:** The graphs including vertices with the same number of neighbors. Bounded valence graphs are generally employed in the modeling of molecular structures.
- ⋄ **(c) Irregular bounded valence graphs:** The graphs obtained by introducing irregularities to the graphs in (b) by deleting and adding some edges.

- ◇ **(d) Regular Meshes:** The 2D, 3D and 4D meshes where each vertex has connections with 4, 6 and 8 neighbors. 3D objects can be represented as 3D mesh graphs in object recognition.
- ◇ **(e) Irregular Meshes:** Meshes obtained by introducing irregularities into the graphs in (d) by adding extra edges.
- ◇ **(f) Scale-free graphs:** The graphs whose degree distribution follows a power law and obtained by scale-free graph generator of the C++ Boost Graph Library. World Wide Web, social networks, and flight networks are a few examples for scale-free graphs.
- ◇ **(g) Planar graphs:** The graphs which have planar embeddings such that no edges cross. Molecular structures and graph databases in biometric identification satisfy the planarity condition.

4.2 Postprocessing. Rounding algorithms may return a solution, in which some edges are covered several times. Therefore, we applied a postprocessing algorithm to decrease the number of redundant nodes in the hub cover and improve the solution quality. Algorithm 6 summarizes the iterations of the postprocessing algorithm. After obtaining the solution by any of the rounding algorithms in line 1, we compute the number of times that each edge is covered by the selected vertices. In line 4, for each vertex in the solution, we check if the vertex is redundant. If it is redundant, then we remove that vertex from the solution and update the number of times each edge is covered by the remaining vertices.

Algorithm 6 Postprocessing Algorithm

1: Get the solution x from any one of the rounding algorithms
2: $C_{(i,j)} = x_i + x_j + \sum_{k \in \mathcal{K}^{(i,j)}} x_k \quad \forall (i,j) \in E$
3: $V' = \{j \in V \mid x_j = 1\}$
4: **for all** $j \in V'$ **do**
5: Find the set of edges, E' covered by vertex j
6: **if** $C_{(i,k)} > 1, \quad \forall (i,k) \in E'$ **then**
7: $x_j \leftarrow 0$
8: $C_{(i,k)} \leftarrow C_{(i,k)} - 1$
9: **end if**
10: **end for**

4.3 Experimental Results In this section, we carried out a computational experiment to test the performances of the relaxation models and rounding methods on various types of graph databases. First, we compare the lower bounds obtained by the LP and SDP relaxations over all instances. The empirical cumulative distributions in Figure 3 indicate that the LP relaxation gives a tighter lower bound relative to the SDP relaxation. The optimal solutions are denoted by IP in the figure. In almost 90% of the instances, the gaps between the optimal and the LP solutions are less than 10%. On the other hand, the SDP relaxation achieves that gap in 75% of the instances.

Figures 4(a) and 4(b) compare the upper bounds obtained by the rounding methods applied to the optimal solutions of the LP and SDP relaxations before and after postprocessing. The results without postprocessing indicate that the SDP rounding algorithm is superior to the primal and dual rounding algorithms by providing tighter upper bounds. In 70% of the instances, SDP rounding algorithm provides upper bounds with optimality gaps less than 30%. The fraction of the instances decreases to 25% and

15% to obtain the same upper bound by primal and dual rounding algorithms, respectively. On the other hand, surprisingly the rounding algorithm developed for MTS outperforms all other rounding algorithms. The rounding algorithm using the optimal LP solution of MTS provides the optimal solution in almost 45% of the instances. With the postprocessing, the percentage of the instances that can be solved to optimality increases to 55%. The results indicate that postprocessing algorithm eliminates the redundant vertices and improves the solution quality considerably for other algorithms as well. On the other hand, postprocessing does not change the relative performances of the algorithms. Nonetheless, $PRMHC$ and $DRMHC$ derive the most benefit from postprocessing. The cumulative fraction of the instances for which $PRMHC$ returns optimal solutions changes from 15% to 40% by postprocessing. The corresponding change for $DRMHC$ is from 8% to 32%. After the postprocessing algorithm, in 70% of the instances, the SDP rounding algorithm provides upper bounds with optimality gaps less than 5%. Without postprocessing the same optimality gap is achieved in about 25% of the instances. The fraction of the instances decreases to 45% and 35% to obtain the same upper bound by $PRMHC$ and $DRMHC$ with the postprocessing algorithm.

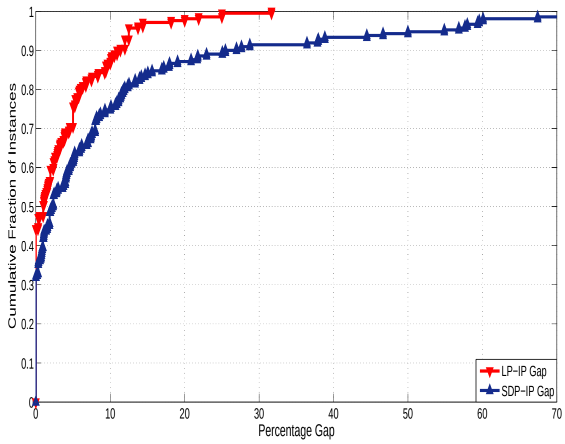
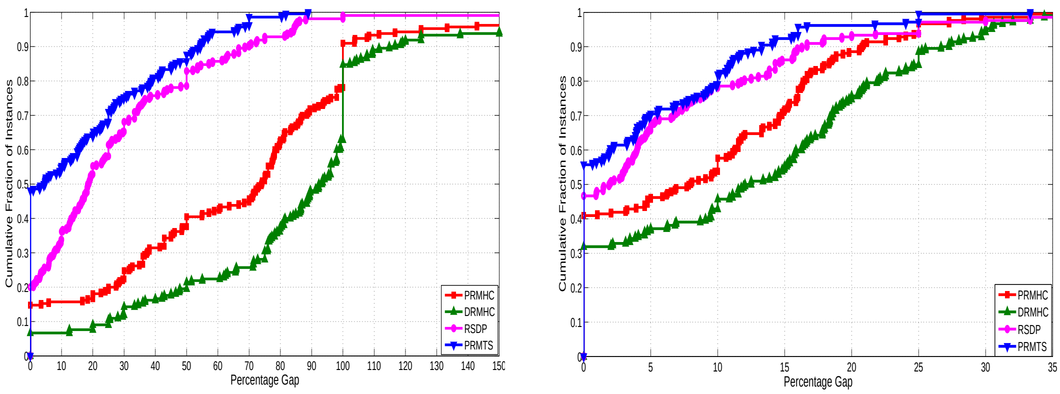


Figure 3: The empirical cumulative distributions of the optimality gaps of LP and SDP relaxations



(a) Before postprocessing (b) After postprocessing

Figure 4: The empirical cumulative distributions of the optimality gaps of the rounding algorithms before and after postprocessing

Finally, we analyze the rounding algorithms over problem classes to figure out if the performance of the algorithms changes with respect to the problem classes. Figure 5 summarizes and compares the performances of the rounding algorithms over different problem classes. The results indicate that the performances of the algorithms depend on the problem classes. The primal and the dual rounding algorithms applied to the optimal LP relaxation of MHC are the most sensitive algorithms. On the other hand, the primal rounding algorithm for the complementary problem MTS is the least sensitive algorithm over the problem classes. The variations of the performances of the algorithms are generally low for the mesh graphs in classes (d) and (e).

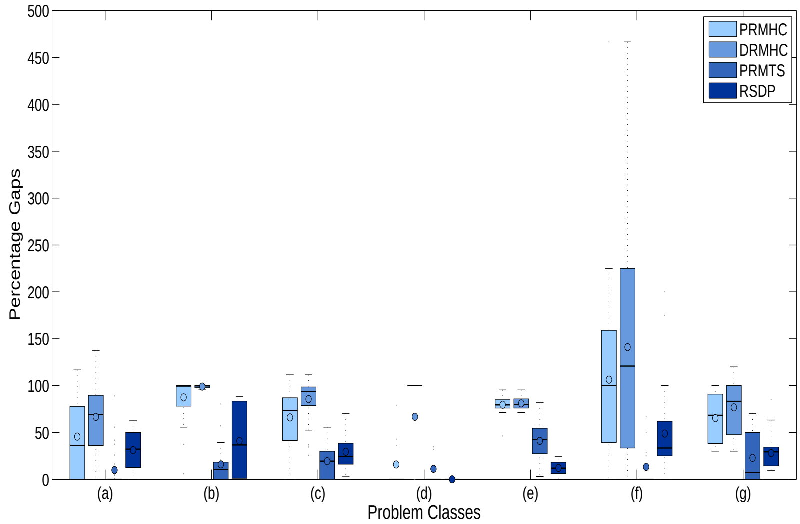


Figure 5: The variation of the optimality gaps of the rounding methods with respect to the problem classes.

5. Conclusion. In this study, we presented several new mathematical programming formulations along with their relaxations for the minimum hub cover problem. We also introduced two novel rounding algorithms $RSDP$ and $PRMTS$, and compared those with two well-known algorithms proposed for the set covering problem in the literature. The results indicate that the algorithms proposed in this study are superior to the benchmark algorithms in terms of solution quality.

Our semidefinite programming relaxation may be used to give an approximation bound for the minimum hub cover problem. However, at this point it is difficult to give such a result for the minimum hub cover problem. Even for special problem classes, where the number of candidate vertices to cover an edge is less than or equal to three, a formal analysis to obtain an approximation bound seems beyond reach. Nonetheless, based upon our empirical results, we conjecture that our SDP relaxation may achieve an approximation bound less than two for the minimum hub cover problem.

We used the SDPA-M solver, which is developed to solve small to medium size instances limited to 2,000 constraints and 2,000 variables. On the other hand, the parallel version of SDPA referred to as SDPARA, can solve instances with up to a million constraints (Fujisawa et al., 2014). As a future study, we plan to employ the parallel implementation of the semidefinite programming solver and test the performance of the SDP relaxation in terms of solution time.

References

Fujisawa, K., Endo, T., Yasui, Y., Sato, H., Matsuzawa, N., Matsuoka, S., and Waki, H. (2014). Peta-scale general solver for semidefinite programming problems with over two million constraints. In *The 28th IEEE International Parallel and Distributed Processing Symposium*, India.

Goemans, M. X. and Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfability problems using semidefinite programming. *Journal of ACM*, 42:1115–1145.

Halperin, E. (2002). Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM Journal on Computing*, 31:1608–1623.

Hochbaum, D. (1982). Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on Computing*, 11:555–556.

Jamil, H. M. (2011). Computing subgraph isomorphic queries using structural unification and minimum graph structures. In *SAC*, pages 1053–1058, Taichung, Taiwan.

Karakostas, G. (2005). A better approximation ratio for the vertex cover problem. In *32nd International Colloquium on Automata, Languages and Programming(ICALP)*, pages 1043–1050, Lisboa, Portugal.

Kawabata, T. (2014). What is MCS? http://strcomp.protein.osaka-u.ac.jp/kcombu/doc/README_mcs.html.

Kojima, M., Fujisawa, K., Nakata, K., and Yamashita, M. (2005). SDPA (semidefinite programming algorithm) user's manual. Technical report, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Japan.

Rivero, C. and Jamil, H. M. (2014a). On isomorphic matching of large disk resident graphs using an xquery engine. The 5th International Workshop on Graph Data Management: Techniques and Applications, Chicago, USA.

Rivero, C. R. and Jamil, H. M. (2014b). Exact subgraph isomorphism using graphlets and minimum hub covers. Submitted to PVLDB.

Santo, M., Foggia, P., Sansone, C., and Vento, M. (2003). A large database of graphs and its use for benchmarking graph isomorphism algorithms. *Pattern Recognition Letters*, 24:1067–1079.

Yelbay, B., Birbil, Ş. İ., Bülbül, K., and Jamil, H. (2014). Approximating the minimum hub cover problem on planar graphs. submitted for publication.

Yelbay, B., Birbil, Ş. İ., Bülbül, K., and Jamil, H. M. (2013). Trade-offs computing minimum hub cover toward optimized graph query processing. http://arxiv.org/abs/1311.1626.