A note on the Clustered Set Covering Problem
Laurent Alfandari, Jérôme Monnot

To cite this version:

HAL Id: hal-01508784
https://hal.archives-ouvertes.fr/hal-01508784
Submitted on 14 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.
A note on the Clustered Set Covering Problem*

Laurent Alfandari†, Jérôme Monnot‡

Abstract
We define an NP-hard clustered variant of the Set Covering Problem where subsets are partitioned into \( K \) clusters and a fixed cost is paid for selecting at least one subset in a given cluster. We show that the problem is approximable within ratio \((1+\varepsilon)(e/e-1)H(q)\), where \( q \) is the maximum number of elements covered by a cluster and \( H(q) = \sum_{i=1}^q \frac{1}{i} \).

Key-words: Integer Programming, Set Covering, Maximal Coverage, Approximation.

1 Problem statement
In the classical Set Covering Problem (SCP), we are given a set of elements \( C = \{1, \ldots, n\} \), a collection \( S = \{S_1, \ldots, S_m\} \subseteq 2^C \) of subsets of \( C \) covering \( C \) and a non-negative weight \( c(S_j) \geq 0 \) for each set \( S_j \in S \). The goal is to find a set cover \( S' = \{S_{j_1}, \ldots, S_{j_t}\} \subseteq S \), verifying \( \bigcup_{j=1}^t S_{j_j} \cap S_{j_k} \neq \emptyset \) for every \( j_k \neq j \), and minimizing \( c(S') = \sum_{j=1}^t c(S_{j_j}) \). This problem has been widely studied by the computer science community and the main results given on it are the following: SCP is NP-hard, even in the unweighted case, i.e., \( c(S_j) = 1 \forall j = 1, \ldots, m \) [8]. SCP is \( H(\Delta) \)-approximable where \( H(\Delta) = \sum_{i=1}^\Delta \frac{1}{i} \) and \( \Delta \) is the maximum size of a set of \( S \), i.e., \( \Delta = \max_{j \leq m} |S_j| \) [4]; this gives a \((1 + \ln n)\)-approximation for SCP since \( \Delta \leq n \) and \( H(n) \leq 1 + \ln n \). On the other hand, SCP is not \((1-\varepsilon)\ln n\)-approximable for every \( \varepsilon > 0 \) [7] closing the gap between positive and negative results on this problem. Finally, the restriction of SCP where \( \Delta \) and \( \delta \) are upper bounded by some constants is APX-complete [12]: here \( \delta \) is the maximum number of sets of \( S \) containing a given element of \( C \), i.e., \( \delta = \max \{ p : \exists S_{j_1}, \ldots, S_{j_p} \text{ such that } r_{i=1}^p S_{j_i} \neq \emptyset \} \).

We define the following variant of SCP, called Clustered Set Covering Problem (Clustered-SCP). Let \( C = \{1, \ldots, n\} \) be a set of elements and \( S = \{S_1, \ldots, S_m\} \) be a collection of subsets of \( C \). A positive cost \( c_j = c(S_j) \) is associated with every subset \( S_j \in S \). Moreover, we assume that the index set \( J = \{1, \ldots, m\} \) is partitioned into \( K \) disjoint subsets \( J = \{J_k : k = 1, \ldots, K\} \), i.e., \( \cup_{k=1}^K J_k = J \) and \( J_k \cap J_{k'} = \emptyset \) for \( k \neq k' \). For \( k = 1, \ldots, K \), cluster \( F_k \subseteq S \) is defined by \( F_k = \{S_j \in S : j \in J_k\} \), and a fixed-cost \( f_k \geq 0 \) is paid as soon as at least one subset is selected within cluster \( F_k \) for \( k = 1, \ldots, K \). The Clustered Set Covering Problem is to cover all elements of \( C \) by a collection of subsets \( S' \subseteq S \) minimizing the sum of the costs of the selected subsets and the fixed costs. In other words, we want to find a set cover \( S' \) minimizing \( c(S') \) plus the cost of the clusters used in \( S' \), i.e.,

---

*Research supported by the French Agency for Research under the DEFIS program TODO, ANR-09-EMER-010.
†ESSEC Business School, Av. B. Hirsch, 95021 Cergy Pontoise, France, alfandari@essec.fr
‡CNRS, UMR 7243, F-75775 Paris, France, and Université Paris-Dauphine, LAMSADE, F-75775 Paris, France, monnot@lamsade.dauphine.fr
\( \bar{c}(S') = c(S') + \sum_{k: S_i \cap F_k \neq \emptyset} f_k; \) to simplify, such a value \( \bar{c}(S') \) will be called the \textit{clustered set value} of \( S' \). The problem can be formulated as the following Integer Linear Program:

\[
\begin{align*}
\text{Minimize} & \quad \sum_{k=1}^{K} f_k y_k + \sum_{j=1}^{m} c_j x_j \\
\text{s.t.} & \quad \sum_{j=1}^{m} a_{ij} x_j \geq 1 \quad \text{for } i = 1, \ldots, n \\
& \quad y_k \geq x_j \quad \text{for } k = 1, \ldots, K, \ j \in J_k \\
& \quad x_j, y_k \in \{0, 1\}
\end{align*}
\]

where binary data \( a_{ij} = 1 \) iff \( i \in S_j \). Various minimum-cost multi-commodity flow problems in transportation planning can be formulated as Clustered-SCP. This is the case for the Crew Pairing problem in air transportation when crews are partitioned into clusters \( k = 1, \ldots, K \) according to their assigned airport basis for example ([10] [2]), or for fleet scheduling problems when vehicles or planes are of different types \( k = 1, \ldots, K \) ([2]), and a fixed cost is paid for using a given resource type. Also consider the labeled weighted Vertex Cover problem defined as follows: given a simple graph \( G = (V, E) \) where each vertex \( v \in V \) has a weight \( w(v) \geq 0 \) and a label \( \mathcal{L}(v) \in \{c_1, \ldots, c_K\} \) (ie., a color) and each label \( c_i \) has a cost \( f_i \geq 0 \), we want to find a vertex cover \( V' \) minimizing its weight \( w(V') = \sum_{v \in V'} w(v) \) plus the cost of the labels used by \( V' \) \( w(V) = \sum_{v \in V} \mathcal{L}(v) \). Recall that a vertex cover of a graph \( G = (V, E) \) is a subset of vertices \( V' \subseteq V \) such that \( \forall e = (u, v) \in E, V' \cap \{u, v\} \neq \emptyset \). Labeled optimization has been investigated for many graph problems [3] [11] [9]. Clearly, the labeled weighted Vertex Cover problem is a particular case of the Clustered Set Covering Problem.

As SCP is a particular case of the Clustered SCP where \( f_k = 0 \) for all \( k = 1, \ldots, K \) (or Clustered SCP is equivalent to solve SCP when \( J \) are the trivial partitions, i.e., either \( K = m \) and \( J_i = \{i\} \) or \( K = 1 \) and \( J_1 = \{1, \ldots, m\} \)), then the Clustered SCP is also \textbf{NP-hard}. The approximation approach proposed in this paper extends in some way the master-slave approach of [1] to more general fixed-charge covering problems.

\section{Some complexity results}

It is well known that SCP can be solved in polynomial time when \( \Delta = \max_{i \leq m} |S_i| \leq 2 \) (see for instance comment of problem [SP5] page 222 in [3]). For Clustered SCP, it is not the case. It also depends on the structure of the clusters. For instance, Clustered SCP is \textbf{NP-hard} even if \( \Delta = 1 \).

\textbf{Lemma 1.} \textit{Clustered SCP is \textbf{NP-hard} even if \( \Delta = 1 \).}

\textit{Proof.} The reduction is done from SCP. From an instance \( I = (C, S, c) \) of SCP where \( S = \{S_1, \ldots, S_m\} \) and \( C = \{1, \ldots, n\} \), we build an instance \( I_0 = (C_0, S_0, c_0, K, J, f) \) of Clustered-SCP by setting \( C_0 = C \), \( K = m \), and replacing \( S_k = \{i_1, i_2, \ldots, i_{|S_k|}\} \in S \), for \( k = 1, \ldots, m \), by \( |S_k| \) sets \( S_{(k, i_j)} = \{i_j\} \) for \( j = 1, \ldots, |S_k| \) in the Clustered-SCP instance. Moreover, we set \( J_k = \{(k, i_j) : i_j \in S_k\} \); Hence, the cluster set \( F_k \) is defined by \( F_k = \{S_{(k, i_j)} : i_j \in S_k\} \). Finally, we set \( c_0 = 0 \) and \( f_k = c(S_k) \). Thus, the clusters play the role of the sets of the SCP instance. In \( I_0 \), each set contains only one element and any set
We build an instance \( K = \text{number} \) of SCP approximation algorithm is defined as follows.

Theorem 2. Assume that the size of each cluster is upper bounded by a constant, i.e., \( \forall k \leq K, |J_k| \leq a \). Then, Clustered-SCP is equivalent to approximate SCP.

Proof. Let \( I = (C, S, c) \) be an instance of SCP where \( S = \{S_1, \ldots, S_m\} \). We build an instance \( I_0 = (C_0, S_0, c_0, K, J, f) \) of Clustered-SCP by setting \( C_0 = C, S_0 = S, c_0 = c, K = m, J = \{J_1, \ldots, J_m\} \) where \( J_k = \{k\} \) (hence, the corresponding cluster set is given by \( \mathcal{F}_k = \{S_k\} \) and fixed costs \( f_k = 0 \) for \( k = 1, \ldots, K \). Clearly, \( I_0 \) is computed from \( I \) in polynomial time and each cluster is upper bounded by a constant (\( a = 1 \)). It is easy to see that \( S' \) is a set cover of \( I \) with value \( c(S') \) if \( S' \) is a set cover of \( I_0 \) with clustered set value \( c_0(S') = c(S') \).

Now, let us prove that we can polynomially reduce Clustered-SCP when each cluster is upper bounded by a constant to SCP. Let \( I = (C, S, c, K, J, f) \) be an instance of Clustered-SCP where \( S = \{S_1, \ldots, S_m\}, J = \{J_1, \ldots, J_k\} \) and \( \forall k \leq K, |J_k| \leq a \) for some constant \( a \).

We build an instance \( I_0 = (C_0, S_0, c_0) \) of SCP as follows: \( C_0 = C \) and for each \( J_k \in J \), we build \( 2^{|J_k|} - 1 \) sets \( S_A = \bigcup_{j \in A} S_j \) for \( A \subseteq J_k, A \neq \emptyset \), with weight \( c_0(S_A) = f_k + \sum_{j \in A} c(S_j) \). In other words, we generate all non-empty subset induced by a cluster, for every cluster. Thus, \( S_0 = \{S_A : \exists J_k \in J \text{ and } A \subseteq J_k \text{ with } A \neq \emptyset\} \). This instance \( I_0 \) of SCP can be constructed within \( 2^m m = O(m) \) time.

Note that Theorem 2 also holds when \( a = \text{poly}(\log m) \) (i.e., \( \forall k \leq K, |J_k| \leq \text{poly}(\log m) \)) where \( \text{poly} \) is any polynomial. We recall that \( \delta = \max\{j : \exists S_{i_1}, \ldots, S_{i_j} \text{ such that } \cap_{p=1}^{j} S_{i_p} \neq \emptyset\} \) and \( \Delta = \max_{i \leq m} |S_i| \).

Corollary 3. Assume that each cluster is upper bounded by a constant. Then, Clustered-SCP is \( \text{APX} \)-complete when \( \Delta \) and \( \delta \) are upper bounded by constants.

Proof. The \( \text{APX} \)-completeness of SCP when \( \Delta \) and \( \delta \) are upper bounded by constants (for instance, when \( \Delta = 3 \) and \( \delta = 2 \)) is known from [12]. Hence, the result follows from Theorem 2.

Before introducing in section 4 the main result of the paper, i.e. the log-approximability of Clustered-SCP, we first need to approximate in section 3 another \( \text{NP} \)-hard problem, that we call Minimum Cover-Ratio Problem (MCRP), which appears as a subproblem in the general approximation algorithm for Clustered-SCP.

3 Approximation of the Minimum Cover-Ratio Problem

The Minimum Cover-Ratio Problem (MCRP) that appears as a subproblem in the Clustered-SCP approximation algorithm is defined as follows.

Definition 1. Given two subsets \( I \) and \( J \) of elements, a cost function \( c : J \rightarrow \mathbb{N} \), a positive number \( f \) and binary data \( a_{ij} \in \{0, 1\} \) for \( (i, j) \in I \times J \), the Minimum Cover-Ratio Problem
(MCRP) formulates as follows:

\[ \text{Minimize } r(x) = \frac{f + \sum_{j \in J} c_j x_j}{\sum_{i \in I} \text{cov}_i(x)} \]

\[ x_j \in \{0, 1\} \]

where \( \text{cov}_i(x) = \min(1; \sum_{j \in J} a_{ij} x_j) \) is equal to 1 if \( x \) covers element \( i \), 0 otherwise.

To our knowledge, the MCRP has never been studied before. If the fixed cost \( f \) is equal to zero, then the problem is trivial since it suffices to select index \( j \) with minimum ratio \( c_j / \sum_{i \in I} a_{ij} \), set \( x_j = 1 \) and all other variables to zero for obtaining the optimal solution. The general problem is \( \text{NP} \)-hard as shown in the following proposition.

**Proposition 4.** MCRP is \( \text{NP} \)-hard.

**Proof.** The reduction is from the Set Covering Problem (SCP). Consider an instance of SCP such that the element set \( C = \{1, \ldots, n\} \) is to be covered by \( S = \{S_1, \ldots, S_m\} \). Construct the MCRP instance by setting \( I = C \), \( J = \{1, \ldots, m\} \), \( a_{ij} = 1 \) if and only if \( i \in S_j \), and \( f = n \sum_{j=1}^{m} c_j \). Let \( \bar{x} \) (resp. \( x \)) denote an arbitrary MCRP solution covering exactly all \( n \) (resp. at most \( n-1 \)) elements of \( I = C \). We have

\[ r(\bar{x}) \geq \frac{n \sum_{j=1}^{m} c_j + c_x}{n-1} \]

\[ = \sum_{j=1}^{m} c_j + \frac{\sum_{j=1}^{m} c_j}{n-1} + \frac{c_x}{n-1} \]

\[ \geq \sum_{j=1}^{m} c_j + \frac{\sum_{j=1}^{m} c_j}{n} \]

\[ = \frac{f + \sum_{j=1}^{m} c_j}{n} \]

\[ \geq r(x) \]

Thus, an MCRP optimal solution necessarily satisfies that \( \sum_{i \in I} \text{cov}_i(x) = n \), i.e. all elements of \( I = C \) are covered, which means that an optimal SCP solution is found.

For approximating MCRP, we use existing approximation results for the Budgeted Maximum Coverage Problem (BMCP) defined in [10].

**Definition 2.** [10] Given two subsets \( I \) and \( J \) of elements, a cost function \( c : J \to \mathbb{N} \), a budget \( B > 0 \) and binary data \( a_{ij} \in \{0, 1\} \) for \( (i, j) \in I \times J \), the Budgeted Maximum Coverage Problem (BMCP) is formulated as follows:

\[ \text{Maximize } \sum_{i \in I} \text{cov}_i(x) \]

\[ \text{s.t. } \sum_{j \in J} c_j x_j \leq B \]

\[ x_j \in \{0, 1\} \]

where, again, \( \text{cov}_i(x) = \min(1; \sum_{j \in J} a_{ij} x_j) \).
In [10], it is proved that there exists a polynomial-time algorithm APPROX-BMCP($B$) approximating BMCP with input budget bound $B$ within performance ratio $1 - \frac{1}{e}$. We derive from this result the following approximation result for MCRP.

**Theorem 5.** MCRP is approximable within performance ratio $(1 + \epsilon)(e/e - 1)$ in time polynomial in both $n$ and $1/\epsilon$.

*Proof.* Let $c_{\text{min}} = \min\{c_j : j \in J\}$, and

$$T = \left\lceil \frac{\ln(\sum_{j \in J} c_j / c_{\text{min}})}{\ln(1 + \epsilon)} \right\rceil$$

Set $B_t = (1 + \epsilon)^t c_{\text{min}}$ for $t = 0, 1, \ldots, T$, and remark that $B_0 = c_{\text{min}}$, $B_t = (1 + \epsilon)B_{t-1}$, and

$$B_T = c_{\text{min}} \exp\left( \frac{\ln(\sum_{j \in J} c_j / c_{\text{min}})}{\ln(1 + \epsilon)} \ln(1 + \epsilon) \right)$$

$$= \sum_{j \in J} c_j$$

As $c_{\text{min}} \leq B_t \leq \sum_{j \in J} c_j$ there exists at least one $t \in \{0, \ldots, T\}$ such that $B_t \geq \sum_{j \in J} c_j x_j^*$, where $x^* = (x_j^*)$ is an optimal MCRP solution. Let $\hat{t}$ denote the smallest index $t$ verifying the above condition, i.e.

$$\hat{t} = \arg \min_{t \in \{0, \ldots, T\}} \{ t : (1 + \epsilon)^t c_{\text{min}} \geq \sum_{j \in J} c_j x_j^* \}$$

By definition of $\hat{t}$, we have:

$$\sum_{j \in J} c_j x_j^* \leq (1 + \epsilon)^\hat{t} c_{\text{min}} \leq (1 + \epsilon) \sum_{j \in J} c_j x_j^*$$

(5)

Now, consider the following algorithm.

**Begin / Algorithm APPROX-MCRP /**

**For** $t = 0, 1, \ldots, T$

$B_t \leftarrow (1 + \epsilon)^t c_{\text{min}}$

$x^t \leftarrow$ APPROX-BMCP($B_t$)

**EndFor**

Return $\bar{x} \leftarrow \min_{t=0, \ldots, T} \frac{f + \sum_{j \in J} c_j x_j^t}{\sum_{i \in I} \text{cov}_i(x^t)}$

**End**

We obtain from the fact that APPROX-BMCP($B_t$) is a $(1 - \frac{1}{e})$-approximation for BMCP ([10]):

$$\sum_{i \in I} \text{cov}_i(x^\hat{t}) \geq (1 - \frac{1}{e}) \sum_{i \in I} \text{cov}_i(x^*)$$

$$\Rightarrow \frac{f + c_x^\hat{t}}{\sum_{i \in I} \text{cov}_i(x^\hat{t})} \leq \left( \frac{e}{e - 1} \right) \frac{f + (1 + \epsilon)^\hat{t} c_{\text{min}}}{\sum_{i \in I} \text{cov}_i(x^*)}$$

$$\Rightarrow r(\bar{x}) \leq r(x^\hat{t}) \leq (1 + \epsilon) \frac{e}{e - 1} r(x^*)$$

using (5)
which concludes the proof of Theorem 5. \(\square\)

Based on the above approximation result for MCRP, we can now introduce the main result for Clustered-SCP in next section.

## 4 Approximation of Clustered-SCP

The approximation algorithm for Clustered-SCP is based on an iterative approximate solving of the Minimum Cover-Ratio Problem (MCRP) introduced in section 3. The main principle of the algorithm is to solve at each iteration \(K\) subproblems: for each cluster \(k\), we find a collection of subsets within \(\mathcal{F}_k\) minimizing the ratio of the total cost of that collection (including fixed cost) by the number of new elements covered by that collection. Each subproblem is obviously a Minimum Cover-Ratio Problem and can be approximated as shown in section 2. Then the best collection with minimum ratio among all clusters \(k\) is added to the solution in a greedy way. The fixed cost of the chosen cluster is set to zero and the process is iterated until all elements of \(C\) are covered.

\begin{verbatim}
Begin / Algorithm APPROX-CLUSTERED-SCP /
U ← C (subset of elements that remain to cover)
S^h ← ∅ (current solution)
Repeat
  For k = 1, . . . , K do
    S_k ← collection of subsets returned by APPROX-MCRP for the MCRP subproblem:
    \[
    \min \{ r_k(S') = \frac{f_k + c(S')}{|(\bigcup_{S_j \in S'} S_j) \cap U|} : S' \subseteq \mathcal{F}_k \} 
    \]
  Endfor
  l ← arg min\(1 \leq k \leq K\) \(r_k(S_k)\)
  S^h ← S^h ∪ \{S_l\}
  f_l ← 0
  U ← U \ (\bigcup_{S_i \in S, S_j} S_j)
Until U = ∅
Output S^h
End
\end{verbatim}

An example of application of the above algorithm when the minimum-ratio subproblem is solved to optimality, is given in figure 1.

Before analyzing the approximation ratio of the above algorithm, we introduce the following lemma. For every element \(i \in C\), let \(\rho_i = r_l(S_l)\) where \(S_l\) is the collection of subsets selected by the Clustered-SCP algorithm that covers element \(i\) for the first time at some iteration \(l\) (in other words, element \(i\) has not been covered during the \(l - 1\) first iterations).

**Lemma 6.** Let \(\rho_i = r_l(S_l)\) where \(S_l\) is the first collection of subsets selected by the Clustered-SCP algorithm that covers element \(i\), over all iterations. Then \(\bar{c}(S^h) = \sum_{i=1}^{n} \rho_i\).

**Proof.** Let \(l_t\) (resp. \(U_t\)) denote selected index \(l\) (resp. set \(U\)) at iteration \(t\) of the algorithm, \(t = 1, \ldots, T\), and \(f_{lt}\) the value of the fixed cost of cluster \(l\) at iteration \(t\) (ie., \(f_l\) or 0). We
Cluster 1, \( f_1 = 10 \)

Cluster 2, \( f_2 = 12 \)

\[ \text{figures above subsets} \]

\[ = \text{subset costs} \]

\[ \text{Iteration 1:} \]

Min ratio = \((10 + 3 + 2)/8 = 1.87\)

(f1 is then set to 0)

\[ \text{Iteration 2:} \]

Min ratio = \((12 + 3 + 4)/5 = 3.80\)

(f2 is then set to 0)

STOP: all elements are covered

Figure 1: Example of application of APPROX-CLUSTERED-SCP algorithm with minimum-ratio MCRP subproblem solved to optimality.

have:

\[
\bar{c}(S^h) = \sum_{t=1}^{T} c(S_{lt}) + f_{lt} \\
= \sum_{t=1}^{T} r_{lt}(S_{lt}) \times |\cup_{S_j \in S_{lt}} S_j \cap U_t| \\
= \sum_{i=1}^{n} \{r_{lt}(S_{lt}) : t \text{ is the first iteration where } i \text{ is covered}\} \\
= \sum_{i=1}^{n} \rho_i
\]

\[ \square \]

**Theorem 7.** Algorithm APPROX-CLUSTERED-SCP approximates the Clustered SCP within ratio \((1 + \epsilon)\frac{q}{n}H(q)\), where \( q = \max_{1 \leq k \leq K} |\cup_{S_j \in F_k} S_j| \) and \( H(q) = \sum_{i=1}^{q} \frac{1}{i} \) is the q-th harmonic number (note that \( q \leq n \)).

**Proof.** Let \( I = (C, S, c, K, J, f) \) be an instance of Clustered-SCP where \( S = \{S_1, \ldots, S_m\} \), \( J = \{J_1, \ldots, J_K\} \). Let \( S^* \) denote an optimal solution for the Clustered SCP, \( K^* = \{k : \ldots \)
\( S^* \cap F_k \neq \emptyset \), \( S_k^* = S^* \cap F_k \), and \( J_k^* = \{ j : S_j \in S_k^* \} \). Hence, \( S^* = \{ S_j : j \in \cup_{k \in K^*} J_k^* \} \).

Transform every subset \( S_j \in S^* \) into \( S_j' \subset S_j \) such that no element \( i \) belongs to two different subsets \( S_j' \) and \( S_j'' \). If so, remove the element from other subsets until it appears only once in a subset. In other words, \( \{ S_j' : j \in \cup_{k \in K^*} J_k^* \} \) is a partition of \( C \). Let \( n_k = |\cup_{j \in J_k^*} S_j'| \) and re-order these \( n_k \) elements as \( i_{k,1}, i_{k,2}, \ldots, i_{k,n_k} \) so that there are picked in that order by algorithm APPROX-CLUSTERED-SCP. Recall that \( \rho_{i_{k,s}} \) is the value of \( r_k(S_k) \) when element \( i_{k,s} \) has been covered for the first time by algorithm APPROX-CLUSTERED-SCP. Since in the worst case, elements \( i_{k,1}, \ldots, i_{k,s-1} \) have been covered during some previous iterations, we have by Theorem 5:

\[
\rho_{i_{k,s}} \leq \frac{e}{e-1} \left( f_k + \sum_{j \in J_k^*} c_j \right) \frac{1}{n_k - s + 1} \quad \text{for } k \in K^*, s = 1, \ldots, n_k \tag{6}
\]

We deduce from lemma 6 that

\[
\tilde{c}(S^h) = \sum_{i=1}^{n} \rho_i = \sum_{k \in K^*} \sum_{j \in J_k^*} \sum_{i \in S_j'} \rho_i = \sum_{k \in K^*} \sum_{s=1}^{n_k} \rho_{i_{k,s}} \leq (1 + \epsilon) \frac{e}{e-1} \sum_{k \in K^*} \sum_{s=1}^{n_k} \frac{f_k + \sum_{j \in J_k^*} c_j}{n_k - s + 1} \quad \text{by } (6)
\]

\[
= (1 + \epsilon) \frac{e}{e-1} \sum_{k \in K^*} H(n_k) \times \left( f_k + \sum_{j \in J_k^*} c_j \right) \leq (1 + \epsilon) \frac{e}{e-1} H(q) \sum_{k \in K^*} \left( f_k + \sum_{j \in J_k^*} c_j \right) = (1 + \epsilon) \frac{e}{e-1} H(q) \tilde{c}(S^*)
\]

which ends the proof. \( \square \)

5 Conclusion

We have designed a polynomial-time algorithm achieving a logarithmic approximation ratio of \( (1 + \epsilon) \frac{e}{e-1} H(q) \leq (1 + \epsilon) \frac{e}{e-1} (1 + \epsilon) \ln n \), with \( q \leq n \), for the Clustered Set Covering Problem. As the Set Covering Problem is a particular case of Clustered Set Covering and SCP cannot be approximated within a better ratio than \( (1 - \epsilon) \ln n \), the achieved ratio is asymptotically tight. We believe this approach could be extended to various fixed-cost optimization problems, not only for covering problems. Finally, an open problem is to design classes of NP-hard optimization problems for which the fixed-cost variant conserves in some way the approximability properties of the original problem.
References


