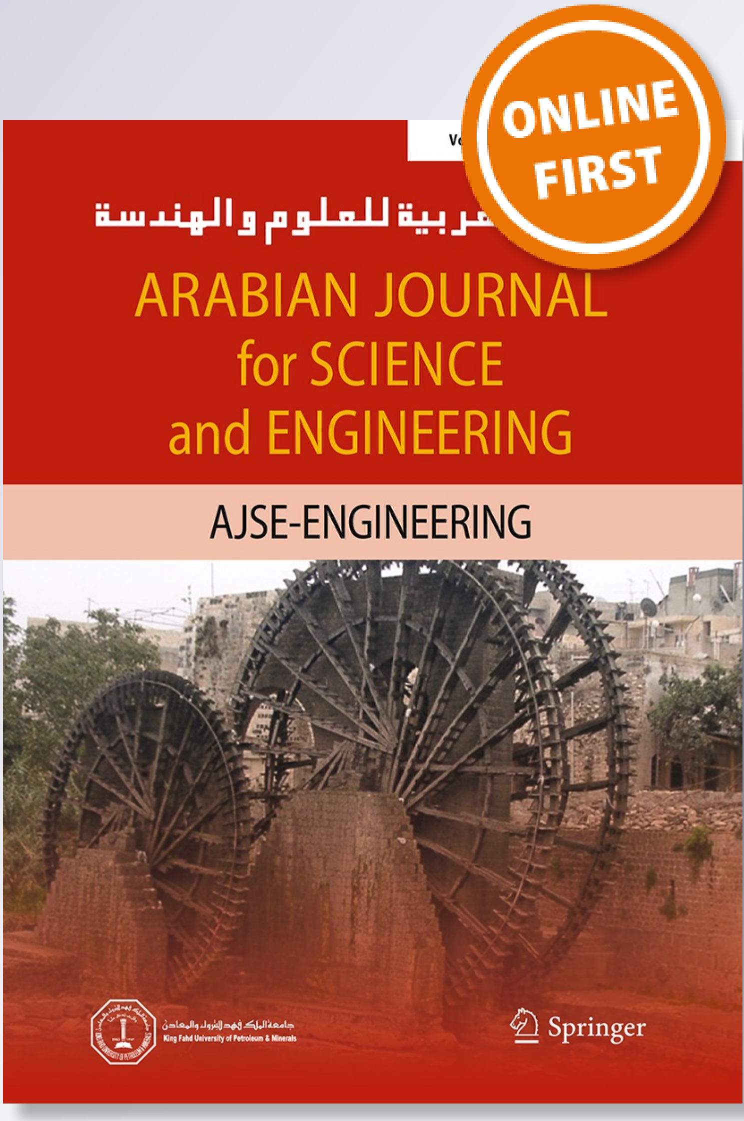
A New Approximation Algorithm for k-Set Cover Problem

Hanaa A. E. Essa, Yasser M. Abd El-Latif, Salwa M. Ali & Soheir M. Khamis

Arabian Journal for Science and Engineering

ISSN 1319-8025

Arab J Sci Eng
DOI 10.1007/s13369-015-1895-3



Springer

Arab J Sci Eng
DOI 10.1007/s13369-015-1895-3

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

A New Approximation Algorithm for *k*-Set Cover Problem

Hanaa A. E. Essa[1] · **Yasser M. Abd El-Latif**[2] · **Salwa M. Ali**[2] · **Soheir M. Khamis**[2]

Received: 5 May 2015 / Accepted: 6 October 2015

Abstract In the set cover problem (SCP), a set of elements $X = \{x_1, x_2, \ldots, x_n\}$ and a collection $F = \{S_1, S_2, \ldots, S_m\}$ of subsets of X, for some integers $n, m \geq 1$, are given. In addition, each element of X belongs to at least one member of F. The problem is to find a sub-collection $C \subseteq F$ such that $\bigcup_{S \in C} S = X$ and C has the minimum cardinality. When $|S| \leq k$ for all $S \in F$, the k-set cover problem (k-SCP) is obtained. For all $k \geq 3$, the k-SCP is an NP-complete optimization problem (Karp in Complexity of computer computations. Plenum Press, New-York, pp 85–103, 1972). It is well known that a greedy algorithm for the k-SCP is a h_k-approximation algorithm, where $h_k = \sum_{i=1}^{k} \frac{1}{i}$ is the k^{th} harmonic number. Since the SCP is a fundamental problem, so there is a research effort to improve this approximation ratio. In this paper, the authors propose an approximation algorithm which accepts any instance of the k-SCP problem as an input. This approximation algorithm is a $(1 + \frac{1}{k})$-algorithm with a polynomial running time for $k \geq 6$ that improves the previous best approximation ratio $h_k - 0.5902$ for all values of $k \geq 6$.

Keywords Set cover problem (SCP) · An approximation algorithm · A greedy algorithm · An NP-complete optimization problem

✉ Yasser M. Abd El-Latif
Y.AbdEllatif@sci.asu.edu.eg

Hanaa A. E. Essa
essahanaa@yahoo.com

Salwa M. Ali
Sal.M.Ali65@gmail.com

Soheir M. Khamis
soheir_khamis@hotmail.com

1 Department of Math., Faculty of Science, Tanta University, Tanta, Egypt

2 Division of Computer Science, Department of Math., Faculty of Science, Ain Shams University, Cairo, Egypt

1 Introduction

The set cover problem (SCP) is an *NP*-complete combinatorial optimization problem. This problem and its special cases are very important since they have several applications in many fields such as efficient testing, statistical design of experiments, airline crew scheduling, and facility placement problems, see for details [2,3].

In the weighted set cover problem (WSCP), a set of elements $X = \{x_1, x_2, \ldots, x_n\}$ and a collection F of subsets of X are given with a positive weight $w(S)$ for each $S \in F$, where $\bigcup_{S \in F} S = X$. The goal is to find a sub-collection $C \subseteq F$ such that $\bigcup_{s \in C} S = X$ and its weight $\sum_{S \in C} w(S)$ is minimized.

Considering instances of the WSCP such that each $S \in F$ has at most k elements, i.e., $|S| \leq k$, it leads to a new version of the WSCP called weighted k-set cover problem (k-WSCP). The WSCP and the k-WSCP in which all sets in F have uniform weights are treated as the un-weighted versions.

A greedy algorithm [4] for the k-WSCP is a h_k-approximation algorithm, where $h_k = \sum_{i=1}^{k} \frac{1}{i}$ is the kth harmonic number. Furthermore, this bound is tight even for the k-SCP [5].

For unbounded values of k, Slavík [6] has shown that the approximation ratio of the greedy algorithm for the SCP is $\ln n - \ln \ln n + \Theta(1)$, where $|X| = n$. Feige [7] has proven that unless $NP \subseteq DTIME(n^{poly \log n})$, the SCP cannot be approximated within a factor $(1 - \varepsilon) \ln n$, for any $\varepsilon > 0$. In [8], it has been proven that if $P \neq NP$,

Published online: 03 November 2015

then the SCP cannot be approximated within a factor c log n, for some constant c. This result showed that the greedy algorithm is an asymptotically best possible approximation algorithm for weighted and un-weighted versions of the problem [unless $NP \subseteq DTIME(n^{poly \log n})$]. Goldschmidt and Hochbaum [9] have modified the greedy algorithm for the SCP and shown that the resulting algorithm has a ratio $h_k - \frac{1}{6}$.

For all $k \geq 3$, the k-SCP is known to be NP-complete [1] and approximating it within $1+\delta$ is NP-hard, for some fixed $\delta > 0$. When $k = 2$, this problem becomes a graph problem known as edge cover, which can be solved optimally via a straight forward transformation to maximum matching [10]. Halldórsson has presented an algorithm based on local search that has an approximation ratio $h_k - \frac{1}{3}$ for the k-SCP and a $(1.4 + \varepsilon)$-approximation algorithm for the 3-SCP [10]. In [11], Duh and Fürer have improved this result and presented a $(h_k - \frac{1}{2})$-approximation algorithm for the k-SCP. Also, Levin has proposed a $(h_k - \frac{196}{390})$-approximation algorithm for the k-SCP for all values of $k \geq 4$ [12]. Athanassopoulos et al. have introduced an improved algorithm with approximation ratio $h_k - 0.5902$ for $k \geq 6$ [13].

In this paper, the authors demonstrate an approximation algorithm for solving the k-SCP. The idea of this algorithm is to efficiently separate the collection F of any instance (X, F) of the k-SCP into smaller sub-collections for which this problem is easier to tackle. To apply this idea, we introduce a graph representation of the k-SCP.

The remaining part of this paper is organized as follows. Section 2 consists of the formulization of the SCP, the set cover, and two representations of any instance (X, F) of the SCP, namely the matrix and the graph representations. Moreover, this section contains two lemmas that are used in proving the correctness of the suggested idea of our algorithm. In Sect. 3, the authors describe a new algorithm for solving the k-SCP. Finally, a conclusion of the paper is given in Sect. 4.

2 Basic Definitions and Concepts

The *set cover problem* (SCP) is a fundamental problem in the class of covering problems. Any instance of this problem consists of a set of elements $X = \{x_1, x_2, \ldots, x_n\}$ and a collection $F = \{S_1, S_2, \ldots, S_m\}$ of subsets of X, for $m, n \in \mathbb{N}$. The objective of this problem is to select the minimum number of subsets of F so that every element in X is contained in at least one of these selected subsets. In the SCP, the set $C \subseteq F$ is called a *cover* of the set X, if $\bigcup_{S \in C} S = X$. The solution of this problem is a cover of minimum cardinality. For any two sets $A, B \subseteq F$, A covers the set $\bigcup_{S \in B} S$, if $\bigcup_{S' \in A} S' = \bigcup_{S \in B} S$ and $A \subseteq B$.

	S_1	S_2	S_3	S_4	S_5	S_6
x_1	1	0	0	1	1	0
x_2	0	0	0	1	0	1
x_3	1	1	0	0	1	0
x_4	1	0	0	0	1	0
x_5	1	1	1	0	0	0
x_6	1	0	0	0	0	0

Fig. 1 Matrix A that represents the given instance (X, F)

2.1 The Representations of the SCP

Every instance (X, F) of the SCP can be represented by (0, 1)- matrix $A = [a_{ij}]$, for $1 \leq i \leq |X|$ and $1 \leq j \leq |F|$. The rows (resp. the columns) of A correspond to the elements of X (resp. F). The entries of A are determined by a_{ij}, where $a_{ij} = 1$, if $x_i \in S_j$ and zero, otherwise. In the following example, we show how to construct the matrix A that corresponds to a specified instance (X, F).

Example 1 Given $X = \{x_1, x_2, \ldots, x_6\}$, $F = \{S_1, S_2, \ldots, S_6\}$, where $S_1 = \{x_1, x_3, \ldots, x_6\}$, $S_2 = \{x_3, x_5\}$, $S_3 = \{x_5\}$, $S_4 = \{x_1, x_2\}$, $S_5 = \{x_1, x_3, x_4\}$, and $S_6 = \{x_2\}$. Figure 1 illustrates the matrix A of the given instance (X, F). The entry a_{ij} in A (the entry at the intersection of the ith row and the jth column) is one if an element x_i is in the subset S_j, otherwise a_{ij} is zero. For example, the entry $a_{11} = 1$ because the element x_1 is in S_1. Also, $a_{23} = 0$ because $x_2 \notin S_3$.

In the following, we construct a graph $G = (V, E)$ as a new representation of any instance (X, F) of the SCP. As usual, V is the finite set of vertices and E is the set of edges of G. V and E are obtained as follows.

1. Each vertex of V corresponds to one member S of F and has the same labeled of S. Hence, $V \equiv F$.
2. For each $S_i, S_j \in F, 1 \leq i, j \leq |F|, j \neq i$, check whether $|S_i \cap S_j| \geq 1$ or not. If true, join the vertex S_i and the vertex S_j by an edge.

By a similar way, it is easy to obtain a graph G that corresponds to an instance (X, F), via using the corresponding matrix A. In Fig. 2, we show the constructed graph for the given instance (X, F) in the above example.

For $V' \subseteq V$, $G[V']$ denotes the sub-graph of G induced by V'. The vertex set of $G[V']$ is V' and the edge set consists of those edges of G with both end points in V'. Let $G = (V, E)$ be a constructed graph for any instance of the k-SCP and $P(V)$ be the set of all subsets of V. Since $V \equiv F$, so each $W \in P(V)$ is corresponding to a subset of the collection F. Let $\tilde{C} : P(V) \to P(V)$ be a function defined by: for every $W \in P(V)$, $\tilde{C}(W)$ is the set of vertices corresponding to the

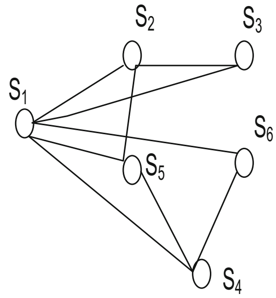


Fig. 2 Constructed graph for an example in Fig. 1

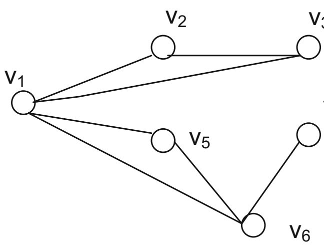


Fig. 3 Graph for Example 2

cover of the set $\bigcup_{S\in W} S$ with minimum cardinality in $G[W]$ and $\tilde{C}(W) \subseteq W$.

The *open neighborhood* of a vertex $v \in V$, $N(v)$, is given by $\{u \in V : \{u, v\} \in E\}$. The *closed neighborhood* or simply the neighborhood of v is denoted by $N[v] = N(v) \cup \{v\}$. The neighborhood of a set $S \subseteq V$ is defined as $N[S] = \bigcup_{s\in S} N[s]$. For $r \in \mathbb{N}$, the r^{th} *neighborhood* of $v \in V$ is defined recursively as $N_r[v] = N[N_{r-1}[v]]$, where $N_1[v] = N[v]$.

Example 2 In this example, we find the open and close neighborhoods for the vertices in a graph G. Let $G = (V, E)$ be a graph and $V = \{v_1, v_2, \ldots, v_6\}$. In Fig. 3, we introduce the open and the close neighborhood for the vertex v_3. The open neighborhood is $N(v_3) = \{v_2, v_1\}$, and the close neighborhood is $N[v_3] = \{v_3, v_2, v_1\}$. Also, $N_2[v_3] = \{v_1, v_2, v_3, v_5, v_6\}$.

Now, we discuss the main idea of the suggested algorithm that depends on the separation of the set of vertices in the constructed graph for any instance (X, F) of the k-SCP into subsets. The sub-graphs induced by the subsets of this separation divide the constructed graph into small parts for which it becomes easier to tackle the k-SCP. For the constructed graph $G = (V, E)$ of any instance (X, F) of the underlying problem, let $H = \{H_1, H_2, \ldots, H_L\}$ be a collection of pairwise disjoint subsets of V, i.e.,$H_i \subset V$, for $i = 1, 2, \ldots, L$. This collection leads to obtain a bound on the cardinality with respect to an optimal solution of the k-SCP.

The following two lemmas are essential for designing the algorithm that will be given in Sect. 3.

Lemma 1 *Let X be a set of n elements, F be a collection of m subsets of X, $n, m \geq 1$, and $G = (V, E)$ be a constructed graph for the instance (X, F) of the k- SCP. For a collection $H = \{H_1, H_2, \ldots, H_L\}$ of pairwise disjoint subsets of V in the graph G, we have $|\tilde{C}(V)| \geq \sum_{i=1}^{L} |\tilde{C}(H_i)|$, where $\tilde{C}$ is a function from P(V) to P(V) given above.*

Proof For each subset $H_i \in H$, consider the neighborhood $N[H_i]$. The set $\tilde{C}(V) \cap N[H_i]$ has to correspond to a cover of the set $H_i \subseteq F$, where $\tilde{C}(V)$ is the set of vertices corresponding to the cover of the set X with minimum cardinality. On the other hand, also $\tilde{C}(H_i) \subset N[H_i]$ is corresponding to the cover of the set $\bigcup_{S\in H_i} S$, $H_i \subseteq F$, using a minimum number of vertices in G. Therefore, we get $|\tilde{C}(V) \cap N[H_i]| \geq |\tilde{C}(H_i)|$. Combining this for all subsets of the collection H, we get

$$|\tilde{C}(V)| \geq |\bigcup_{i=1}^{L} \tilde{C}(V) \cap N[H_i]| \geq |\bigcup_{i=1}^{L} \tilde{C}(H_i)| = \sum_{i=1}^{L} |\tilde{C}(H_i)|$$

as claimed. □

Lemma 1 states that the collection H leads to a lower bound on the minimum cardinality of a cover of the set X. Moreover, such a collection leads to find an approximation of this cardinality. The next lemma gives an approximation of the k-SCP.

Lemma 2 *Let X be a set of n elements, F be a collection of m subsets of X, $n, m \geq 1$, and G = (V, E) be a constructed graph for the instance (X, F) of* the *k-SCP. Also, let $H = \{H_1, H_2, \ldots, H_L\}$ be a collection of pairwise disjoint subsets of V in the graph G and $T_1, \ldots, T_L$ be subsets of V with $H_i \subseteq T_i$, for all $i = 1, \ldots, L$. If there exists a real number $\rho > 1$ such that $|\tilde{C}(T_i)| \leq \rho.|\tilde{C}(H_i)|$ holds for all $i = 1, \ldots, L$ and $\bigcup_{i=1}^{L} \tilde{C}(T_i)$ is corresponding to the cover of the set X, then the set $\bigcup_{i=1}^{L} \tilde{C}(T_i)$ is a ρ- approximation of the k-SCP.*

Proof It is clear that

$$|\bigcup_{i=1}^{L} \tilde{C}(T_i)| = \sum_{i=1}^{L} |\tilde{C}(T_i)| \leq \rho. \sum_{i=1}^{L} |\tilde{C}(H_i)| \leq \rho.|\tilde{C}(V)|.$$

□

In the next section, we use the condition of Lemma 2 at $\rho = 1 + \frac{1}{k}$. We focus on constructing suitable subsets T_i such that $H_i \subseteq T_i \subseteq V$, for all $i = 1, \ldots, L$. These T_i's assist to obtain a ρ-approximation of the k-SCP.

3 The Approximation Algorithm for the k-SCP

In this section, we describe a new algorithm to construct an approximation solution for the k-SCP, $k \geq 6$. If we apply this algorithm on any instance of the k-SCP, then the algorithm returns a $(1 + \frac{1}{k})$-approximate set cover. Let $G = (V, E)$ be the constructed graph for any instance of the k-SCP. An approximation solution is constructed by computing optimal solutions for small parts of the given instance and then taking the union of them. Each part corresponds to a sub-instance of the given input instance of the introduced algorithm. The input of this algorithm is the matrix A_{nxm} that represents the instance (X, F), where X is a set of n elements and F is a collection of m subsets of X. The algorithm works as follows. Initially, let a set $C = \varphi$, the algorithm creates the graph representation of the input instance (X, F) with a set of vertices V, when executing steps 3–13 of the next algorithm, where $V \equiv F$. After that, this algorithm chooses an arbitrary vertex $S \in V$ and finds the rth neighborhood of S, for $r \geq 0$, where $N_0[S] = \{S\}$. Then, it computes sets cover of minimum cardinalities corresponding to these neighborhoods as long as the following Inequality (1) holds for all r.

$$\left|\tilde{C}(N_{r+1}[S])\right| > \left(1 + \frac{1}{k}\right) . \left|\tilde{C}(N_r[S])\right|. \qquad (1)$$

Denote by $\hat{r}$ the smallest r at which the Inequality (1) is violated or the condition in step 18 of the introduced algorithm is satisfied. The value of r is increased until we find a corresponding $\hat{r}$. In this case, add $\tilde{C}(N_{\hat{r}+1}[S])$ to the current solution C. After that, remove all vertices in $N_{\hat{r}+1}[S]$ and edges incident to them from the graph G. For each vertex, S belongs to the remaining set of vertices; if the corresponding $S \in F$ is a subset of $\bigcup_{S' \in C} S'$, then remove the vertex S and all edges incident to this vertex from the graph G. Let V' be the set of remaining vertices. The process repeats again for the remaining part of G by choosing a new vertex $S \in V'$ and finding a new $\hat{r}$.The algorithm terminates when $V' = \Phi$ or the set C is a set cover of the input instance (X, F). The algorithm is described formally in the following ASC algorithm. The ASC is the name of our algorithm, and it is the abbreviation of approximating set cover.

Algorithm ASC: compute a $(1 + \frac{1}{k})$-approximate of any instance of k-SCP, $k \geq 6$.

Input: the matrix A_{nxm} that represents the instance (X, F), where X is a set of n elements and F is a collection of m subsets.

Output: $C \subseteq F$ such that $\bigcup_{S \in C} S = X$.

Begin
1: $C \leftarrow \varphi$, $L \leftarrow 1$;
2: $V \leftarrow F$, $V' \leftarrow V$;
3: **for** i =1 to m **do**
4: **for** j =i+1 to m **do**
5: $Z \leftarrow 0$;
6: **for** m =1 to n **do**
7: **if** $a_{mi} x a_{mj} \neq 0$ **then**
8: $Z \leftarrow Z + 1$;
9: **od**
10: **if** $Z>0$ **then**
11: put an edge between S_i and S_j;
12: **od**
13: **od**
14: **while** $V' \neq \varphi$ **do**
15: choose randomly $S \in V'$;
16: $r \leftarrow 0$;
17: **while** $|\tilde{C}(N_{r+1}[S])| > (1 + \frac{1}{k}).|\tilde{C}(N_r[S])|$ **do**
18: **if** $|\tilde{C}(N_{r+1}[S])| > (r + k)$ **then**
19: $\tilde{C}(N_{r+1}[S]) \leftarrow \tilde{C}(N_r[S])$;
20: $N_{r+1}[S] \leftarrow N_r[S]$
21: **else**
22: $r \leftarrow r + 1$;
23: **od**
24: $C \leftarrow C \cup \tilde{C}(N_{r+1}[S])$;
25: $V' \leftarrow V' \backslash N_{r+1}[S]$;
26: $H_L \leftarrow N_r[S]$;
27: $T_L \leftarrow N_{r+1}[S]$;
28: **if** C is a cover of the set X **then**
29: **return** C;
30: $L \leftarrow L+1$;
31: **for** each $S \in V'$ **do**
32: **if** the corresponding $S \in F$ is a subset of $\bigcup_{S' \in C} S'$ **then**
33: $V' \leftarrow V' \backslash \{S\}$;
34: **od**
35: **od**
36: **return** C;
End

It is clear that the ASC algorithm is terminated since the number of iterations, L, of the outer while-loop (steps 14–35) is bounded by the min $\{|X|, |F|\} \leq (|X|.|F|)^{1/2}$. Since $V \equiv F$ and F is a cover of X, the subsets of F that correspond to the removed elements of V which are not covered previously, are added to C. Hence, when $V' = \varphi$ at the end of the algorithm, the output C is corresponding to a set cover of the input instance. The input of the ASC algorithm is the matrix A_{nxm} that represents the instance (X, F), where X is a set of n elements and F is a collection of m subsets. Thus, the size of memory which we need is $O(|X||F|)$. Also, we need $O\left(|F|^2\right)$ for the graph representation of the instance (X, F) of the SCP.

According to Lemma 2 and the fact that the output C of the ASC algorithm is corresponding to a cover for the input set X, we obtain the following lemma:

Lemma 3 *The set C which is computed by the ASC- algorithm is corresponding to a cover of the input set X and is a* $(1+\frac{1}{k})$*-approximation of an optimal solution for an instance (X, F) of the k-SCP,* $k \geq 6$.

Proof The two collections $\{H_1, H_2, \ldots, H_L\}$ and $\{T_1, T_2, \ldots, T_L\}$ which are obtained by the ASC algorithm satisfy the following inequality: $|\tilde{C}(T_i)| \leq (1+\frac{1}{k}).|\tilde{C}(H_i)|$,for $i = 1, \ldots, L$. Moreover, $C = \bigcup_{i=1}^{L} \tilde{C}(T_i)$. So,

$$|C| = \bigcup_{i=1}^{L} \left|\tilde{C}(T_i)\right| = \sum_{i=1}^{L} \left|\tilde{C}(T_i)\right| \leq \left(1+\frac{1}{k}\right).\sum_{i=1}^{L} \left|\tilde{C}(H_i)\right|$$

By using Lemma 1, it is easy to conclude that $(1+\frac{1}{k}).\sum_{i=1}^{L} |\tilde{C}(H_i)| \leq (1+\frac{1}{k}).|OPT|$, where *OPT* refers to the optimal solution of the k-SCP. Thus, $|C| \leq (1+\frac{1}{k}).|OPT|$. □

Finally, we show that the running time of the suggested algorithm is polynomial. Since the input of this algorithm is a matrix A which represents an input instance (X, F) of the k- SCP, so one can consider $N = |X|.|F|$ being the size of this input instance (X, F). Then, the time for constructing the corresponding graph (steps 3–13) is proportional to $O(N^2)$. Evidently, the number of iterations of the outer while-loop (steps 14–35) is bounded by min $\{|X|, |F|\} \leq N^{1/2}$. Thus, it is sufficient to compute the running time, t, of an iteration of the outer while-loop (steps 14–35) since the number of the iterations of this loop is known. The running time of for-loop (steps 31–34) is $O(N^2)$. Note that the cardinality of $\tilde{C}(N_{r+1}[S])$ is bounded by $(r+k)$. If r is a fixed reasonable integer, then the time for finding the set $\tilde{C}(N_{r+1}[S])$ can be computed. Therefore, $t = O(N^q)$, where $q = O(r)$.

Table 1 Comparison on the approximation ratio of the k-set cover problem

K	[11]	[12]	[13]	ASC
3	1.3333	1.3333	1.3333	1.3333
4	1.5833	1.5808	1.5833	1.25
5	1.7833	1.7801	1.7833	1.2
6	1.9500	1.9474	1.9208	1.1667
7	2.0929	2.0903	2.0690	1.1429
8	2.2179	2.2153	2.1762	1.125
9	2.3290	2.3264	2.2917	1.11
10	2.4290	2.4264	2.3802	1.1
20	3.0977	3.0952	3.0305	1.05
21	3.1454	3.1428	3.0784	1.048
50	3.9992	3.9966	3.9187	1.02
75	4.4014	4.3988	4.3178	1.013
100	4.6874	4.6848	4.6021	1.01

The next lemma shows that there exists a bound on $\hat{r}$, the first value of r which violates Inequality (1) or satisfies the condition in step 18 of the ASC algorithm.

Lemma 4 *There exists a function b=b(k) such that* $\hat{r}$ *is bounded by b, where* $\hat{r}$ *at which the largest neighborhood is constructed during an iteration of the outer while-loop (steps 14–35) of the ASC algorithm.*

Proof It is clear that $|\tilde{C}(N_1[S])| \geq 1$, because one vertex or more corresponds to a cover of the set $\bigcup_{S' \in N_1[S]} S'$. Consider an arbitrary value of $r < \hat{r}$ and $\rho = 1+\frac{1}{k}$, we have

$$(r+k) \geq \left|\tilde{C}(N_{r+1}[S])\right| > \rho\left|\tilde{C}(N_r[S])\right| > \ldots$$
$$> \rho^r\left|\tilde{C}(N_1[S])\right| \geq \rho^r$$

Since $\rho = 1+\frac{1}{k} > 1$, so the given above inequality has to be violated for some value of r. The bound on r when this inequality is violated for the first time depends only on k and not on the size of the input instance N. □

Remark Setting $b = k^{1.8}$ yields $(b+k) < \rho^b, k \geq 6$. Thus, b is bounded by $O(k^{1.8})$.

Following the above analysis, it is clear that the running time of the ASC algorithm is polynomial.

Now, we give comparison on the approximation ratio of the previous algorithms and the ASC algorithm for the k-SCP. The comparison is introduced in Table 1. It is clear from Table 1 that the ratio of the developed algorithm is less than the previous algorithms for solving the k-SCP for all values of $k \geq 4$, and this is the success for this algorithm.

4 Conclusion

In this paper, a new approximation algorithm for treating the k-SCP is demonstrated, where $k \geq 6$. It is well known that the k-SCP problem is an NP-hard optimization problem for $k \geq 3$. Our algorithm can be applied for any instance of the k-SCP, $k \geq 3$, but the improvement is for values $k \geq 4$. This problem gained the attention of researchers because of its important applications.

So, all previous researches have been concentrating on finding an approximation solution as mentioned in reference [12]. In this reference, a $(h_k - \frac{196}{390})$-approximation algorithm has been presented for all values of $k \geq 4$. In addition, Athanassopoulos et al. [13] have given an algorithm with approximation ratio $h_k - 0.5902$ for $k \geq 6$.

The main idea of the given algorithm is based on the separation of the collection F of an instance (X, F) of the k-SCP into smaller sub-collections for which this problem is easier to tackle. To apply this idea, we introduce a new graph representation corresponding to any instance of the k-SCP. The suggested algorithm creates the graph represented of an input instance of the underlying problem (steps 3–13). Since this algorithm computes an approximation solution of any instance of the k-SCP with an approximation ratio $(1 + \frac{1}{k})$ for $k \geq 6$. So, it improves the previous best approximation ratio $h_k - 0.5902$ for all values of $k \geq 6$.

In a future work, we improve the approximation ratio and modify our algorithm to solve the k-WSCP.

References

1. Karp, R.M: Reducibility among combinatorial problems. In: Miller, R.E.; Thatcher, J.W. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press, New-York (1972)
2. Housos, E.; Elmoth, T.: Automatic optimization of subproblems in scheduling airlines crews. Interfaces. **27**, 68–77 (1997)
3. Lourenço, H.R.; Portugal, P.; Paixão, J.P.: Multiobjective metaheuristics for the bus-driver scheduling problem. Transp. Sci. **35**, 331–343 (2001)
4. Chvátal, V.: A greedy heuristic for the set-covering problem. Math. Oper. Res. **4**, 233–235 (1979)
5. Lovász, L.: On the ratio of optimal integral and fractional covers. Discrete Math. **13**, 383–390 (1975)
6. Slavík, P.: A tight analysis of the greedy algorithm for set cover. J. Algorithms. **25**, 237–254 (1997)
7. Feige, U.: A threshold of ln n for approximating set cover. J. ACM. **45**, 634–652 (1998)
8. Raz, R.; Safra, S.: A sub-constant error probability low-degree test, and sub-constant error-probability PCP characterization of NP. Proc. STOC 1997, pp. 475–484 (1997)
9. Goldschmidt, O.; Hochbaum, D.S.; Yu, G.: A modified greedy heuristic for the set covering problem with improved worst case bound. Inf. Process. Lett. **48**, 305–310 (1993)
10. Halldórsson, M.M.: Approximating k set cover and complementary graph coloring. In: Proc. 5th Conference on Integer Programming and Combinatorial Optimization, IPCO'1996, pp. 118–131 (1996)
11. Duh, R.; Fürer, M.: Approximation of k-set cover by semi local Optimization. Proc. STOC 1997, pp. 256–264 (1997)
12. Levin, A.: Approximating the Un-weighted k-Set Cover Problem, Greedy Meets Local Search. WAOA LNCS 4368, pp. 290–301 (2006)
13. Athanassopoulos, S.; Caragiannis, I.; Kaklamanis, C.: Analysis of approximation algorithms for k-set cover using factor-revealing linear Programs. Theory of Computing Systems, in press, **45**, 555–576 (2009). doi:10.1007/s00224-008-9112-3.