# New Complexity Results for the $k$-Covers Problem

Costas S. Iliopoulos * Manal Mohamed † W.F. Smyth ‡

**Abstract**

The $k$-covers problem ($k$CP) asks us to compute a minimum cardinality set of strings of given length $k > 1$ that covers a given string. It was shown in a recent paper, by reduction to 3-SAT, that the $k$-covers problem is NP-complete. In this paper we introduce a new problem, that we call the $k$-Bounded Relaxed Vertex Cover Problem ($\text{RVCP}_k$), which we show is equivalent to $k$-Bounded Set Cover ($\text{SCP}_k$). We show further that $k$CP is a special case of $\text{RVCP}_k$ restricted to certain classes $G_{\boldsymbol{x},k}$ of graphs that represent all strings $\boldsymbol{x}$. Thus a minimum $k$-cover can be approximated to within a factor $k$ in polynomial time. We discuss approximate solutions of $k$CP, and we state a number of conjectures and open problems related to $k$CP and $G_{\boldsymbol{x},k}$.

**Keywords**: string, cover, regularity, complexity, NP-complete.

## 1 Introduction

The computation of various kinds of "regularities" in given strings $\boldsymbol{x} = \boldsymbol{x}[1..n]$ has been of interest for a quarter-century, signalled by the publication in the early 1980s of several $O(n \log n)$-time algorithms for computing all ***repetitions*** (adjacent identical substrings) [7, 3, 16], work that has more recently been refined to $O(n)$-time algorithms [15, 13]. In response to applications arising in data compression and molecular biology, the computation of repetitions was generalized to computation of ***repeats*** (adjacency condition dropped), for which also $O(n)$-time algorithms have been found [5, 8]; then still further to computation of ***approximate repeats*** [17].

In [2] the idea of a ***quasiperiod*** or ***cover*** was introduced; that is, a proper substring $\boldsymbol{u}$ of the given string $\boldsymbol{x}$ such that every position of $\boldsymbol{x}$ is contained in an occurrence of $\boldsymbol{u}$. Several algorithms to compute covers of $\boldsymbol{x}$ were published in the 1990s, culminating in an algorithm [14] that in $O(n)$ time computes a ***cover array*** specifying all the covers (quasiperiods) of every prefix of $\boldsymbol{x}$; this algorithm thus directly generalizes the border array ("failure function") algorithm [1] that specifies all the borders, hence all the periods, of every prefix of $\boldsymbol{x}$.

In [12] a further extension, the ***$k$-covers problem***, was introduced: compute a minimum set $U_\nu = \{\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_\nu}\}$ of strings of given length $k > 1$ such that every position of $\boldsymbol{x}$ is contained in an occurrence of some element of $U_\nu$. A polynomial-time algorithm was given

*Department of Computer Science, King's College London, London WC2R 2LS, England; csi@dcs.kcl.ac.uk.

†Department of Computer Science, King's College London, London WC2R 2LS, England; manal@dcs.kcl.ac.uk.

‡Algorithms Research Group, Department of Computing & Software, McMaster University, Hamilton, Ontario, Canada L8S 4K1; Department of Computing, Curtin University, Perth WA 6845, Australia; smyth@computing.edu.au.

for this problem, later discovered to be incorrect [18]; just recently the problem itself has been shown to be NP-complete, based on a reduction to 3-SAT [6]. In this latter paper, two $O(n \log n)$ algorithms were described that yielded an approximation to a minimum $k$-cover of $\boldsymbol{x}$; it was conjectured that these algorithms would yield a $k$-cover of cardinality at most $\log n$ times the minimum.

In Section 2 of this paper we introduce a new NP-complete problem which we call the relaxed vertex cover problem. We show that a special case of this problem is equivalent to the $k$-bounded set cover problem. We call this subproblem $k$-bounded relaxed vertex cover ($\text{RVCP}_k$).

In Section 3 we show that the $k$-covers problem is a subproblem of $\text{RVCP}_k$. Thus the existence of an approximation algorithm that achieves at least a ratio of $k$ times the minimum is assured. The new reduction of $k$-covers raises the possibility that in fact $k$-covers can also be approximated to within a lower factor.

In Section 4 we discuss conjectures and open problems derived from the complexity analysis of the $k$-covers problem, both here and in [6].

## 2 The Relaxed Vertex Cover Problem

In this section, we introduce a new problem which we call the ***relaxed vertex cover problem***. Given a directed graph $G = (V, E)$, where $V_o \subseteq V$ is the set of all vertices in $V$ with out-degree $> 0$, find the smallest subset $V' \subseteq V_o$ such that if $(u, v) \in E$, then one of the following conditions holds:

**(C1)** $u \in V'$;

**(C2)** $v \in V'$;

**(C3)** there exist $w_u, w_v \in V'$ such that $(w_u, u) \in E$ **and** $(w_v, v) \in E$.

We say that $V'$ is a ***vertex semi-cover*** of $G$.

The decision form of the relaxed vertex cover problem asks for given $G$ and $\nu$, whether there exists a vertex semi-cover $V' \subseteq V_o$ of $G$ such that $|V'| = \nu$. We call this problem ***RVCP***. If the in-degree of all vertices in $V - V_o$ is no more than $k$ we call this problem the $k$-bounded relaxed vertex cover problem (***RVCP$_k$***) and we show that it is equivalent to the $k$-bounded set cover problem ($\text{SCP}_k$).

$\text{SCP}_k$ is a special case of the set cover problem and is defined as follows: given a collection $U$ of subsets of a finite set $S$ where the number of occurrences in $U$ of any element is bounded by a constant $k$, find a minimum size subset $U' \subseteq U$ such that every element in $S$ belongs to at least one member in $U'$. This problem is well-known to be NP-complete [9]. Bar-Yehuda and Even [4], and Hochbaum [11] presented polynomial time $k$-approximation algorithms for this problem. Halperin [10] described the most effective such algorithm which yields a subset whose cardinality is $k - \frac{(k-1)\cdot \ln \ln n}{\ln n}$ times the minimum.

**Theorem 1** *Problem $RVCP_k$ is equivalent to $SCP_k$.*

**Proof:** First, we show that $\text{RVCP}_k$ can be reduced to $\text{SCP}_k$ in polynomial time. Suppose we are given a directed graph $G = (V, E)$ together with a subset $V'$ of $V_o$, where $V_o \subseteq V$ is the set of all vertices with out-degree $> 0$, an instance of $\text{RVCP}_k$. We construct a set $S$ from $E$ and a collection $U$ of subsets of $S$, an instance of $\text{SCP}_k$. Then we show how $V'$ can be used to calculate a set cover $U'$ such that $U'$ is a cover of $S$ if and only if $V'$ is a vertex semi-cover of $G$.

Suppose the vertices of $V$ are labelled $1, 2, ..., n$ and the arcs $(u, v)$ are labelled $uv$. Let $S$ be the set of labels of arcs of $E$. The set $U$ (initially empty) is constructed as follows: for each vertex $v \in V_o$,

1. Determine $N(v) = \{i|(v, i) \in E\}$, the set of vertices adjacent to vertex $v$ (out-neighbors of $v$).
2. Form $O_v = \{vu|(v, u) \in E\}$, the set of the outgoing arcs.
3. Form $I_v = \{uv|(u, v) \in E\}$, the set of incoming arcs.
4. Form $C_v = \{uw|(u, w) \in E; u, w \in N(v)\}$. the set of arcs between the out-neighbors of $v$.
5. Form $U_v = I_v \cup O_v \cup C_v$.
6. Update $U \leftarrow U \cup \{U_v\}$.

Note that each set $U_v$ corresponds to the set of arcs that could be semi-covered by vertex $v$. The sets $C_v$ are the sets of arcs that satisfy condition **(C3)**. It is not difficult to see that each arc $(v_1, v_2)$, where $v_1, v_2 \in V_o$, appears exactly twice in $U$, while the rest of the arcs cannot appear more than $k$ times. This is because the in-degree of each vertex in $V - V_o$ is no more than $k$.

By construction, we see that $V' = \{i_1, i_2, ..., i_{|V'|}\}$ is a semi-cover of $G$ if and only if the corresponding set $U' = \{U_{i_1}, U_{i_2}, ..., U_{i_{|V'|}}\}$ is a cover of $S$.

Second, we show that $\text{SCP}_k$ can also be reduced to $\text{RVCP}_k$ in polynomial time. Let $S = \{e_1, e_2, ...e_{|S|}\}$ and $U = \{U_1, U_2, ..., U_{|U|}\}$ be a given instance of $\text{SCP}_k$. We construct a graph $G = (V, E)$ such that $|V| = |S| + |U|$, where $|S|$ vertices are associated with the elements in $S$ (*element-vertices*) and $|U|$ vertices are associated with the distinct subsets in $U$ (subset-vertices). The set of arcs $E$ is constructed by adding an arc $(u, v)$ from each subset-vertex $u$ to each element-vertex $v$ that belongs to the subset represented by $u$. Additional arcs are added between the subset-vertices if the two subsets share one or more elements. More formally $E$ is constructed according to the following steps, each performed for every element $U_i \in U$:

1. Let $u$ be the subset-vertex associated with $U_i = \{e_{i_1}, e_{i_2}, ..., e_{i_{|U_i|}}\}$.
2. Determine $E(u)$, the set of element-vertices associated with $e_{i_j}, j \in 1..|U_i|$.
3. Form $E \leftarrow E \cup \{(u, v)|v \in E(u)\}$.
4. Determine $I(u)$, the set of subset-vertices associated with the subset elements in $U$ that intersect with $U_i$.
5. Form $E \leftarrow E \cup \{(u, w)|w \in I(u), w \neq u\}$.

Note that the only vertices in $V$ that have out-degree $> 0$ are the subset-vertices. Additionally, the in-degree of each position-vertex is no more than $k$. Clearly, any set $U' \in U$ is a set cover of $S$ if and only if the set $V'$ is a semi-cover of $G$, where $V'$ is the set of subset-vertices associated with the subsets in $U'$. $\square$

**Corollary 1** *For the $k$-bounded relaxed vertex cover problem ($RVCP_k$) there is a polynomial time algorithm with an approximation ratio $k - \frac{(k-1)\cdot\ln\ln n}{\ln n}$, where $n = |E|$.*

This follows directly from Theorem 1 and the results obtained in [10].

## 3 $\mathrm{RVCP}_k$ and the $k$-Covers Problem

Here we consider the decision form of the $k$-covers problem: given a string $\boldsymbol{x}$ and integers $k > 1$ and $\nu$, decide whether there exists a $k$-cover of $\boldsymbol{x}$ of cardinality $\nu$. We call this problem $k\boldsymbol{CP}$ and we show that it is a special case of $\mathrm{RVCP}_k$.

**Theorem 2** *Every instance of $kCP$ can be reduced to an instance of $RVCP_k$ in polynomial time.*

**Proof:** Suppose now that a string $\boldsymbol{x} = \boldsymbol{x}[1..n]$ and an integer $k$ are given. Let $n$ be the length of the string $\boldsymbol{x}$ and $n'$ be the number of distinct $k$-substrings (substrings of length $k$) in $\boldsymbol{x}$. We initialize a directed graph $G_{\boldsymbol{x},k} = (V, E)$, where $|V| = n' + n$ and $E = \emptyset$. We called the first $n'$ vertices in $V$ the *$k$-substring-vertices* and the remaining $n$ vertices the *position-vertices*. For every distinct $k$-substring $u_i$ where $i = 1, ..., n'$, compute

1. The set $P(u_i)$ of position-vertices that correspond to the positions in $\boldsymbol{x}$ that can be covered by $u_i$, where a position $i$ can be covered by $u_i$ if and only if $u_i$ occurs at some position $j \in i - k + 1..i$.

2. The set $O(u_i)$ of $k$-substring-vertices that correspond to all $k$-substrings of $\boldsymbol{x}$ that overlap with $u_i$, where two strings overlap if and only if there is a non empty prefix of one of them which equals a suffix of the other.

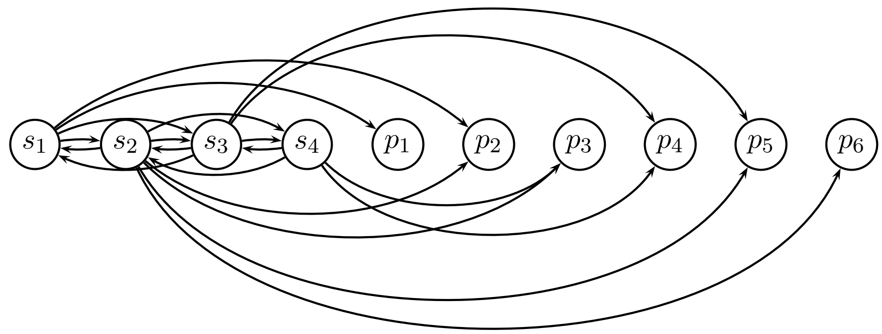
3. If $u$ is the $k$-substring-vertex related to $u_i$ then $E$ is updated as follows:
$$E \leftarrow E \cup \{(u, v)|v \in P(u_i)\} \cup \{(u, w)|w \in Q(u_i), w \neq u\}.$$

Clearly, the $k$-substring-vertices are the only vertices with out-degree $> 0$. Accordingly, any vertex semi-cover of $G_{\boldsymbol{x},k}$ is a set of $k$-substring-vertices. Note that each position in $\boldsymbol{x}$ cannot be covered with more than $k$ distinct $k$-substrings. Thus, the in-degree of all position-vertices is no more than $k$.

Consider a vertex semi-cover $V'$ of $G_{\boldsymbol{x},k}$. Let vertex $s$ be one of the vertices in $V'$ and let $u_s$ be the $k$-substring corresponding to $s$. Then in addition to the outgoing and incoming arcs of $s$, all the arcs pointed to each position-vertex $v \in P(u_s)$ will be semi-covered according to condition **(C3)**. This is because the sources of these arcs are $k$-substring-vertices $\in O(u_s)$

If the alphabet of $\boldsymbol{x}$ is ordered, an algorithm to compute $G_{\boldsymbol{x},k}$ from $\boldsymbol{x}$ can be implemented in $O(n \log n)$ time using a straightforward approach, somewhat faster using a suffix tree to sort the $k$-strings. □

For example, if $\boldsymbol{x} = aabbab$ and $k = 2$, then the only four distinct $k$-substrings are $aa, ab, ba$, and $bb$. Let $s_1, s_2, s_3, s_4$ be the $k$-substring-vertices associated with them. The corresponding graph $G_{aabbab,2}$ is:




where each position-vertex $p_i$ represents position $i$ in $\boldsymbol{x}$. The sets $V_1' = \{s_1, s_2, s_3\}$ and $V_2' = \{s_1, s_2, s_4\}$ are semi-covers of $G_{aabbab,2}$. The semi-cover $V_1'$ corresponds to the minimum $k$-cover $U_1 = \{aa, ab, ba\}$ while $V_2'$ corresponds to $U_2 = \{aa, ab, bb\}$.

Theorem 2 and Corollary 1 show that there is an approximation algorithm that calculates a minimum $k$-cover of a given string $\boldsymbol{x}$ whose cardinality is at most $k - \frac{(k-1)\cdot \ln\ln 2kn}{\ln 2kn}$ times the minimum. This is because the number of arcs in graph $G_{\boldsymbol{x},k}$ formed from $\boldsymbol{x} = \boldsymbol{x}[1..n]$ is at most $2kn$.

## 4 Open Problems

We have shown that for $k \geq 2$, the $k$-covers problem $k$CP is equivalent to RVCP$_k$, hence that efficient algorithms can be used to approximate a minimum $k$-cover as specified in Section 3. Interesting questions remain:

**(Q1)** The set $\mathcal{G}$ of graphs $G_{\boldsymbol{x},k}$ in some sense describes the structure of all strings. To our knowledge these graphs have not previously been reported in the literature. Can the graphs of $\mathcal{G}$ be characterized in another way? What are their defining properties?

**(Q2)** The NP-completeness proof given in [6] is based upon strings whose length $n$ is a function of three parameters: $k$ (the length of the covering substrings), $r$ (the number of variables in the corresponding 3-SAT problem), and $s$ (the number of clauses in the corresponding 3-SAT problem). A short calculation shows that in fact

$$n = (18k+7)r + (42k-3)s + (2k-1),$$

while at the same time the minimum cover size

$$\nu = 9r + 6r' + 8s + 1,\ r' \leq r.$$

Let us call the ratio $\gamma_k = n/(\nu k)$ the ***k-coverability*** of the string $\boldsymbol{x}[1..n]$; observe that $\gamma_k$ has as an upper bound the average number of occurrences in $\boldsymbol{x}$ of the strings in the minimum $k$-cover. Since $\nu \leq 15r+8s+1$, we see then that for the class of strings constructed in [6], $\gamma_k > 6/5$; in other words, the strings in the $k$-cover occur on average somewhat frequently in $\boldsymbol{x}$. What happens when $\gamma_k \leq 6/5$? Can we find a polynomial-time algorithm to compute a minimum $k$-cover given that $\gamma_k$ falls below a certain threshold? For "most" strings and some sufficiently large $k$, we expect that $\nu = \lceil n/k \rceil$, so that $\gamma_k \approx 1$; thus such an algorithm would in fact handle most of the cases that arise.

## Acknowledgements

Costas S. Iliopoulos was supported in part by a Marie Curie fellowship, Wellcome & Royal Society grants. Manal Mohamed was supported by an EPSRC studentship. W.F. Smyth was supported in part by a grant from the Natural Sciences & Engineering Research Council of Canada.

## References

[1] Alfred V. Aho, John E. Hopcroft and Jeffey D. Ullman, *The Design & Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[2] A. Apostolico, M. Farach and C. S. Iliopoulos, *Optimal Superprimitivity Testing for Strings*, Inform. Process. Lett. 39, 1991, 17–20.

[3] Alberto Apostolico and Franco P. Preparata, *Optimal Off-Line Detection of Repetitions in a String*, Theoret. Comput. Sci. 22, 1983, 297–315.

[4] Bar-Yehuda and S. Even, *A Linear Time Approximation Algorithm for the Weighted Vertex Cover Problem*, Journal of Algorithms 2, 1981, 198–203.

[5] Gerth S. Brodal, Rune B. Lyngsø, Christian N. S. Pedersen and Jens Stoye, *Finding Maximal Pairs with Bounded Gap*, J. Discrete Algs. 1, 2000, 77–103.

[6] Richard Cole, Costas S. Iliopoulos, Manal Mohamed, W. F. Smyth and Lu Yang, *Computing the Minimum $k$-Cover of a String*, Proc. Prague Stringology Conf.'03, 2003, 51–62.

[7] Maxime Crochemore, *An Optimal Algorithm for Computing All the Repetitions in a Word*, Inform. Process. Lett. 12–5, 1981, 244–248.

[8] František Franěk, W. F. Smyth and Yudong Tang, *Computing all Repeats Using Suffix Arrays*, J. Automata, Languages & Combinatorics 8–4, 2003, 579–591.

[9] Michael R. Garey and David S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, 1979.

[10] E. Halperin, *Improved Approximation Algorithms for the Vertex Cover Problem in Graphs and Hypergraph*, SIAM Journal on Computing 31(5), 2002, 1608–1623.

[11] D.S. Hochbaum, *Approximation Algorithms for the Set Covering and Vertex Cover Problems*, SIAM Journal on Computing 11(3), 1982, 555–556.

[12] Costas S. Iliopoulos and W. F. Smyth, *On-Line Algorithms for $k$-Covering*, Proc. Ninth Australasian Workshop on Combinatorial Algorithms, 1998, 107–116.

[13] Roman Kolpakov and Gregory Kucherov, *On Maximal Repetitions in Words*, J. Discrete Algs. 1, 2000, 159–186.

[14] Yin Li and W. F. Smyth, *Computing the Cover Array in Linear Time*, Algorithmica 32–1, 2002, 95–106.

[15] Michael G. Main, *Detecting Leftmost Maximal Periodicities*, Discrete Applied Maths. 25, 1989, 145–153.

[16] Michael G. Main and Richard J. Lorentz, *An $O(n \log n)$ Algorithm for Finding All Repetitions in a String*, J. Algs. 5, 1984, 422–432.

[17] Jeanette P. Schmidt, *All Highest Scoring Paths in Weighted Grid Graphs and their Application to Finding All Approximate Repeats in Strings*, SIAM J. Comput. 27–4, 1998, 972–992.

[18] Lu Yang, *Computing the Minimum $k$-Cover of a String*, M. Sc. thesis, McMaster University, 2000, 86 pp.