New Complexity Results for the $k$-Covers Problem

Costas S. Iliopoulos$^{1*}$, Manal Mohamed$^{1**}$, and W.F. Smyth$^{2***}$

1 Department of Computer Science, King’s College London
   London WC2R 2LS, England
csi@dcs.kcl.ac.uk
www.dcs.kcl.ac.uk/staff/csi
2 Algorithms Research Group, Department of Computing & Software
   McMaster University, Hamilton, Ontario, Canada L8S 4K1
   Department of Computing, Curtin University, Perth WA 6845, Australia
   smyth@mcmaster.ca
   www.dcss.mcmaster.ca/~bill/cv.shtml

Abstract. The $k$-covers problem ($k$CP) asks us to compute a minimum cardinality set of strings of given length $k \geq 1$ 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 Relaxed Vertex Cover Problem (RVCP), which we show is a special case of Set Cover (SCP). We show further that kCP is equivalent to RVCP restricted to certain classes $G_{x,k}$ of graphs that represent all strings $x$. We discuss approximate solutions of kCP, and we state a number of conjectures and open problems related to kCP and $G_{x,k}$.

Keywords: String, word, cover, regularity, complexity, NP-complete.

1 Introduction

The computation of various kinds of “regularities” in given strings $x = x[1..n]$ 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) [C81, AP83, ML84], work that has more recently been refined to $O(n)$-time algorithms [M89, K00]. 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 [BLPS00, FST03]; then still further to computation of approximate repeats [S98].

* Supported in part by a Marie Curie fellowship, Wellcome & Royal Society grants.
** Supported by an EPSRC studentship.
*** Supported in part by a grant from the Natural Sciences & Engineering Research Council of Canada.
In [AF91] the idea of a quasiperiod or cover was introduced; that is, a proper substring $u$ of the given string $x$ such that every position of $x$ is contained in an occurrence of $u$. Several algorithms to compute covers of $x$ were published in the 1990s, culminating in an algorithm [LS02] that in $O(n)$ time computes a cover array specifying all the covers (quasiperiods) of every prefix of $x$; this algorithm thus directly generalizes the border array ("failure function") algorithm [AHU74] that specifies all the borders, hence all the periods, of every prefix of $x$.

In [IS98] a further extension, the $k$-covers problem, was introduced: compute a minimum set $U_k = \{u_1, u_2, \ldots, u_\nu\}$ of strings of given length $k > 1$ such that every position of $x$ is contained in an occurrence of some element of $U_k$. A polynomial-time algorithm was given for this problem, later discovered to be incorrect [Y00]; just recently the problem itself has been shown to be NP-complete, based on a reduction to $3$-SAT [CIMSY03]. In this latter paper, two $O(n \log n)$ algorithms were described that yielded an approximation to a minimum $k$-cover of $x$; 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 show that the $k$-covers problem is reducible to a subproblem of the Set Cover Problem (SCP). Thus the existence of an approximation algorithm that achieves at least a logarithmic multiple of the minimum cover is assured [J74, L75]. In fact, the reduction of $k$-covers is to a new problem, a special case of SCP that we call the Relaxed Vertex Cover Problem (RVCp). Since it is well-known [GJ79, S82] that the Vertex Cover Problem (VCP) actually has a polynomial-time algorithm that achieves at most twice the minimum cover, the new reduction of $k$-covers raises the possibility that in fact $k$-covers also can also be approximated within a factor of two.

In Section 3 we discuss conjectures and open problems derived from the complexity analysis of the $k$-covers problem, both here and in [CIMSY03].

2 The Relaxed Vertex Cover Problem

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

A vertex cover of an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ such that if $(u, v) \in E$, then either $u \in V'$ or $v \in V'$. The vertex cover problem (VCP) asks us to determine, for given graph $G$ and integer $\nu$, whether there exists a vertex cover $V'$ of $G$ such that $|V'| = \nu$. VCP is well-known to be NP-complete [GJ79], but several $O(|V| + |E|)$-time greedy algorithms have been described that compute a vertex cover of $G$ whose cardinality is at most twice that of the minimum-cardinality vertex cover of $G$. The first such algorithm discovered, due to Gavril ([GJ79], p. 134), is a straightforward application of maximal matching; another [S82] is based on depth-first search.
It is convenient here to define a relaxed version of VCP. We say that a \textit{vertex semi-cover} of $G = (V, E)$ is a subset $V' \subseteq V$ 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 $(w_v, v) \in E$.

The \textit{relaxed VCP} (RVCP) asks, for given $G$ and $\nu$, whether there exists a vertex semi-cover of $G$ such that $|V'| = \nu$. Of course every vertex cover of $G$ is also a vertex semi-cover of $G$; thus, if $\nu_{VC}^*$ (respectively, $\nu_{RVCP}^*$) is the minimum cardinality of a vertex cover (respectively, semi-cover) $V'$ of $G$, then

$$\nu_{RVCP}^* \leq \nu_{VC}^*.$$  \hfill (1)

Since as just remarked an efficient approximation algorithm can be found for VCP that computes

$$\nu_{VC}^{EST} \in \nu_{VC}^* \cdots 2\nu_{VC}^*,$$

it follows from (1) that the same algorithm will also provide an upper bound for RVCP:

$$\nu_{RVCP}^* \leq \nu_{VC}^{EST} \leq 2\nu_{VC}^*.$$  \hfill (2)

Suppose now that a string $x = x[1\ldots n]$ and an integer $k$ are given. We initialize a graph $G_{x,k} = (V, E)$, where $V = \{1, 2, \ldots, n\}$ and $E = \emptyset$. At every position $i = 1, 2, \ldots, n-k+1$ of $x$, determine the leftmost position $i' \leq i$ in $x$ at which the substring $u = x[i, \ldots, i+k-1]$ occurs: if $i' = i$, compute

$$E \leftarrow E \cup \{(i', i+1), (i', i+2), \ldots, (i', i+k-1)\};$$

while if $i' < i$, compute

$$E \leftarrow E \cup \{(i', i), (i', i+1), \ldots, (i', i+k-1)\}.$$  \hfill (3)

Thus the only edges in $|E|$ are those that connect the leftmost position of each distinct $k$-string $u$ with every other position in $x$ covered by $u$; hence

$$(k-1)(n-k+1) \leq |E| \leq k(n-k+1) - 1.$$  \hfill (4)

For example, given $x = abaab$ and $k = 2$, the corresponding graph $G_{x,k}$ will be given by

$$V = \{1, 2, 3, 4, 5\}, \ E = \{(1, 2), (1, 4), (1, 5), (2, 3), (3, 4)\}.$$  \hfill (5)

Thus the position of the leftmost occurrence of every distinct $k$-string $u$ of $x$ will be adjacent to every other position covered by $u$. If the alphabet of $x$ is ordered, an algorithm to compute $G_{x,k}$ from $x$ can be implemented in $O(kn \log n)$ time using a straightforward approach, somewhat faster using a suffix tree to sort the $k$-strings. Let us designate this algorithm $A(x, k)$. 
Consider a set $U_\nu = \{u_1, u_2, \ldots, u_\nu\}$ of distinct substrings of $x$, each of length $k$, and let $j_i$ denote the leftmost position in $x$ at which $u_i$ occurs, $i = 1, 2, \ldots, \nu$. Let $J_\nu = \{j_1, j_2, \ldots, j_\nu\}$. Now observe that if in fact $U_\nu$ is a $k$-cover of $x$, a vertex semi-cover of $G_{x,k}$ is given by $V' = J_\nu$: for every $(u, v) \in E$, if neither (C1) nor (C2) is true, then by the construction of $E$, (C3) must be true. Conversely, if $V' = \{j_1, j_2, \ldots, j_\nu\}$ is a vertex semi-cover of a graph $G_{x,k}$ formed using $A(x, k)$ for some string $x$ and some $k \geq 2$, then

$$U'_\nu = \{x[j_1..j_1+k-1], x[j_2..j_2+k-1], \ldots, x[j_\nu..j_\nu+k-1]\}$$

must be a $k$-cover of $x$. Thus:

**Theorem 1.** Problem $kCP$ is equivalent to RVCP restricted to graphs $G_{x,k}$ formed according to Algorithm $A(x, k)$. \(\square\)

Since as proved in [CIMS03], $kCP$ is NP-complete over strings $x$ for every fixed choice of $k \geq 2$, we have immediately:

**Theorem 2.** RVCP restricted to graphs computed using Algorithm $A(x, k)$, $k \geq 2$, is NP-complete. \(\square\)

It is not hard to show that RVCP, like VCP, is a special case of SCP:

**Theorem 3.** Every instance of RVCP is equivalent to an instance of SCP.

**Proof.** Suppose we are given a graph $G = (V, E)$ together with a subset $V'$ of $V$. We construct a set $S$ from $E$ and a collection $U'$ of subsets of $S$ from $V'$; we then show that $V'$ is a vertex semi-cover of $G$ (solution of RVCP) if and only if $U'$ is a cover of $S$ (solution of SCP).

Suppose the vertices of $V$ are labelled $1, 2, \ldots, n$ and the edges $(u, v), u < v,$ of $E$ are labelled $uv$. Let $S$ be the set of all labels of edges of $E$. Let $V' = \{i_1, i_2, \ldots, i_\nu\}$. We construct $U'$ (initially empty) according to the following steps, each performed for every $j \in 1..r$:

1. Determine $N(i_j)$, the set of vertices adjacent to (in the neighbourhood of) vertex $i_j$.
2. Form $S_j$, the set of edge labels $i_jq$ $(i_j < q)$ or $qi_j$ $(q < i_j)$ identified by the vertices $q \in N(i_j)$.
3. For every $\ell \neq j$, form $S_{j\ell}$, the set of edge labels $pq$ $(p < q)$ or $qp$ $(q < p)$ determined by all vertices $p \in N(i_j)$ that are adjacent to vertices $q \in N(i_\ell)$.
4. Form $U' \leftarrow U' \cup S_j \cup \bigcup_{\ell \neq j} S_{j\ell}$.

By construction, we see that $V'$ is a semi-cover of $G$ if and only if $U'$ is a cover of $S$. \(\square\)

As noted in the introduction, there exists an efficient algorithm that computes solutions of SCP to within a logarithmic factor of problem size [J74,L75]. By
Theorem 3 the same is true for RVCP, and so we can compute, in addition to (2),
\[ \nu_{RVC}^{EST} \in O(\log n \times \nu_{RVC}^*). \tag{4} \]
Of course we would really like to have the analogue of (2):
\[ \nu_{RVC}^{EST} \in \nu_{RVC}^* \cdot 2\nu_{RVC}^* ??? \tag{5} \]
In the next section we discuss this open question further.

We conclude this section with the following observations:
- Suppose that the substrings \( u_i \) of \( U \) are no longer constrained to have length \( k \) but may instead have any positive length \( \leq k \). This change leads to the idea of a \( k \)-cover, say \( U' \), and a corresponding problem \((P1')\). The graph \( G_{x,k} \) can be formed in the same way as before by executing Algorithm \( A(x,k') \) for every \( k' \in 2..k \), while eliminating any duplicate edges that result during the construction process. If we call the new algorithm \( A'(x,k) \), then an analogue to Theorem 1 holds: \((P1')\) is equivalent to RVCP restricted to graphs formed according to \( A'(x,k) \).
- Observe that, by virtue of the way in which edges are formed in \( G_{x,k} \), all the edges must be expressed in the form \((u,v), u < v \). Thus if we interpret the edges as arcs in a directed graph \( G_{x,k} \), the set \( \mathcal{G} \) of all such digraphs \( G_{x,k} \) is therefore a subset of the set of all acyclic digraphs. \( \mathcal{G} \) is moreover a proper subset since none of its members can be a tournament.

3 Open Problems

We have shown that for \( k \geq 2 \), the \( k \)-cover problem kCP is equivalent to a restriction of RVCP, hence that efficient algorithms can be used to approximate a minimum \( k \)-cover as specified by (3) and (4). Interesting questions remain:

(Q1) The main outstanding question is whether or not (5) holds for RVCP, hence by Theorem 1 for kCP. For the closely-related problem VCP, one standard algorithm [882] for computing \( \nu_{VC}^{EST} \leq 2\nu_{VC}^* \) is based on depth first search (DFS):

\[ \text{compute a DFS tree } T_G \text{ of } G; \]
\[ V' \leftarrow \text{the set of internal nodes of } T_G. \]

For RVCP an analogous algorithm would need to include in \( V' \) only every other level of the internal nodes; that is,
- parents \( P_1 \) of leaf nodes \( L \) in \( T_G \);
- parents \( P_3 \) of parents of \( P_1 \) in \( T_G \);
- etc.
But such an algorithm can yield a very bad estimate of \( \nu_{RVC}^* \); for example, DFS from source vertex 1 of the graph
\[ V = \{1,2,\ldots,2r+2\}; \]
\[ E = \{(1,2); (2,i), 1 \leq i \leq 2r+2; (i+2, i+r+2), 1 \leq i \leq r\} \]
can lead to a covering set $V' = \{1, 3, 4, \ldots, r+2\}$ of cardinality $r+1$, while in fact $\nu_{RVC}^* = 1$ for this graph. A similar example exists for breadth first search (BFS).

It appears that in order to compute a good estimate of $\nu_{RVC}^*$, the vertices of maximum degree should be given special consideration. The following greedy algorithm for graphs $G$ is a generalization of one of the approximation algorithms proposed for strings $x$ in [Y00]:

$$V' \leftarrow \emptyset;$$

while $E \neq \emptyset$ do

- find a vertex $v$ of maximum degree in $G$;
- $V' \leftarrow V' \cup \{v\}$;
- remove $v$ and all its incident edges from $G$.

This algorithm can be implemented in $O(n \log n)$ time, where $|V| = n$, and we conjecture that it computes $|V'| \leq 2\nu_{RVC}^*$, in accordance with (5).

(Q2) The set $\mathcal{G}$ of graphs $G_{x,k}$ formed by Algorithm $A(x, k)$ 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?

(Q3) The NP-completeness proof given in [CIMSY03] 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

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

Let us call the ratio $\gamma_k = n/(\nu k)$ the $k$-coverability of the string $x[1..n]$; observe that $\gamma_k$ has as an upper bound the average number of occurrences in $x$ 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 [CIMSY03], $\gamma_k > 6/5$; in other words, the strings in the $k$-cover occur on average somewhat frequently in $x$.

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.

References


