Proc. 10th SWAT-06, Vol. 4059 in LNCS, pp. 339-350, Springer, 2006

Minimum Membership Set Covering and the Consecutive Ones Property

Michael Dom, Jiong Guo*, Rolf Niedermeier, and Sebastian Wernicke**

Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Fed. Rep. of Germany.
{dom,guo,niedermr,wernicke}@minet.uni-jena.de

Abstract. The Minimum Membership Set Cover problem has recently been introduced and studied in the context of interference reduction in cellular networks. It has been proven to be notoriously hard in several aspects. Here, we investigate how natural generalizations and variations of this problem behave in terms of the consecutive ones property: While it is well-known that classical set covering problems become polynomial-time solvable when restricted to instances obeying the consecutive ones property, we experience a significantly more intricate complexity behavior in the case of Minimum Membership Set Cover. We provide polynomial-time solvability, NP-completeness, and approximability results for various cases here. In addition, a number of interesting challenges for future research is exhibited.

1 Introduction

Set Cover (and, equivalently, Hitting Set [1]) is a core problem of algorithmics and combinatorial optimization [2, 3]. The basic task is, given a collection $\mathcal{C}$ of subsets of a base set S, to select as few sets in $\mathcal{C}$ as possible such that their union is the base set. This models many resource allocation problems and generalizes fundamental graph problems such as Vertex Cover and Dominating Set. Set Cover is NP-complete and only allows for a logarithmic-factor polynomial-time approximation [7]. It is parameterized intractable (that is, W[2]-complete) with respect to the parameter "solution size" [5, 14].

Numerous variants of set covering are known and have been studied [2, 4, 8, 9, 11, 16]. Motivated by applications concerning interference reduction in cellular networks, Kuhn et al. [10] very recently introduced and investigated the Minimum Membership Set Cover problem.

Minimum Membership Set Cover (MMSC)
Input: A set S, a collection $\mathcal{C}$ of subsets of S, and a nonnegative integer k.
Task: Determine if there exists a subset $\mathcal{C}' \subseteq \mathcal{C}$ such that $\bigcup_{C\in\mathcal{C}'} = S$ and $\max_{s\in S} |\{C \in \mathcal{C}' \mid s \in C\}| \leq k$.

* Supported by the Deutsche Forschungsgemeinschaft, Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4.
** Supported by the Deutsche Telekom Stiftung.

In this natural variant again a base set S has to be covered with sets from a collection $\mathcal{C}$. By way of contrast to the classical Set Cover problem, however, the goal is not to minimize the number of sets from $\mathcal{C}$ required to do this, but the maximum number of occurrences each element from S has in the cover. Kuhn et al. [10] showed that MMSC is NP-complete and has similar approximation properties as the classical Set Cover problem.

A well-known line of attack against the hardness of Set Cover is to study special cases of practical interest. Perhaps the most famous one of these cases is Set Cover obeying the *consecutive ones property (c.o.p.)* [11–13, 16, 17]. Herein, the elements of S have the property that they can be ordered in a linear arrangement such that each set in the collection $\mathcal{C}$ contains only whole "chunks" of that arrangement, that is, without any gaps.[1] Set Cover instances with c.o.p. are solvable in polynomial time, a fact which is made use of in many practical applications [11, 13, 16, 17]. Thus, the question naturally arises whether such results can be transferred to MMSC. This is what we study here, arriving at a much more colorful scenario than in the classical case.

In order to thoroughly study MMSC, in particular with respect to the c.o.p., it is fruitful to consider the following generalization.

Red-Blue Hitting Set (RBHS)
Input: An n-element set S, two collections $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ of subsets of S, and a nonnegative integer k.
Task: Determine if there exists a subset $S' \subseteq S$ such that each set in $\mathcal{C}_{red}$ contains *at least one* element from S' and each set in $\mathcal{C}_{blue}$ contains *at most* k elements from S'.

MMSC is the same as RBHS for the case $\mathcal{C}_{red} = \mathcal{C}_{blue}$. However, the RBHS formulation now opens a wide field of natural investigations concerning the c.o.p., the point being that the c.o.p. may apply to either $\mathcal{C}_{red}$, $\mathcal{C}_{blue}$, both, or none of them. The c.o.p. in connection with RBHS leads to a number of different results concerning the computational complexity. This is what we explore here, Table 1 providing a general overview of known and new results.

The main messages from Table 1 (and this work) are:

- In case of only "partial" or even no consecutive ones properties (first three columns), the problem mostly remains NP-complete.
- In the case that both $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ obey the c.o.p., only simple cases are known to be polynomial-time solvable but the general case remains open.
- The case that both $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ obey the c.o.p. allows for a simple and efficient approximation which is only by additive term one worse than an optimal solution. Surprisingly, an optimal solution seems harder to achieve.

[1] The name "consecutive ones" refers to the fact that one may think of a Set Cover instance as a coefficient matrix M where the elements of S correspond to columns and the sets in $\mathcal{C}$ correspond to rows; An entry is 1 if the respective element is contained in the respective set and 0 otherwise. If the Set Cover instance has the c.o.p., then the columns of M can be permuted in such a way that the ones in each row appear consecutively.

Table 1. An overview of previously known results, new results presented in this paper, and open questions for future research regarding the computational complexity of the Red-Blue Hitting Set problem.

<table>
<tr><th></th><th>no c.o.p. requirement</th><th>$\mathcal{C}_{red}$ has c.o.p.</th><th>$\mathcal{C}_{blue}$ has c.o.p.</th><th>$\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ have c.o.p.</th></tr>
<tr><td>No restrictions</td><td>NP-c [10]</td><td rowspan="4">NP-c (Thm. 5)</td><td rowspan="4">NP-c (Thm. 7)</td><td>+1-approx. (Thm. 9)</td></tr>
<tr><td>Fixed max. overlap k</td><td rowspan="3">NP-c (Thm. 5)</td><td rowspan="2">poly.-time (Thm. 11)</td></tr>
<tr><td>Fixed elem. occ. in $\mathcal{C}_{blue}$</td></tr>
<tr><td>Fixed elem. occ. in $\mathcal{C}_{red}$</td><td>?</td></tr>
<tr><td>Card.-2 sets $\mathcal{C}_{red}$ or $\mathcal{C}_{blue}$</td><td colspan="3">NP-c (Thms. 5, 6, 7, and 8)</td><td rowspan="2">trivial</td></tr>
<tr><td>Card.-2 sets $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$</td><td colspan="3">linear-time (Cor. 4)</td></tr>
<tr><td>Fix. card. sets $\mathcal{C}_{red}$</td><td rowspan="2">NP-c [10]</td><td colspan="2" rowspan="2">NP-c (Thms. 5, 7, and 8)</td><td>?</td></tr>
<tr><td>Fix. card. sets $\mathcal{C}_{blue}$</td><td>poly.-time (Thm. 11)</td></tr>
<tr><td>$\mathcal{C}_{blue}$ contains one set</td><td rowspan="2">NP-c (Thm. 2)</td><td>poly.-time (Cor. 12)</td><td rowspan="2">NP-c (Thm. 2)</td><td rowspan="2">poly.-time (Cor. 12)</td></tr>
<tr><td>Fix. num. sets in $\mathcal{C}_{blue}$</td><td>?</td></tr>
</table>

Preliminaries. Formally, the consecutive ones property is defined as follows.

Definition 1. *Given a set $S = \{s_1, \ldots, s_n\}$ and a collection $\mathcal{C}$ of subsets of S, the collection $\mathcal{C}$ is said to have the* consecutive ones property (c.o.p.) *if there exists an order $\prec$ on S such that for every set $C \in \mathcal{C}$ and $s_i \prec s_k \prec s_j$, it holds that $s_i \in C \wedge s_j \in C \Rightarrow s_k \in C$.*

The following simple observation is useful for our NP-completeness proofs.

Observation 1 *Given a set $S = \{s_1, \ldots, s_n\}$ and a collection $\mathcal{C}$ of subsets of S such that all sets in $\mathcal{C}$ are mutually disjoint, the collection $\mathcal{C}$ has the c.o.p..*

In order to simplify the study of Red-Blue Hitting Set, for a given instance $(S, \mathcal{C}_{red}, \mathcal{C}_{blue}, k)$ we will call k the *maximum overlap* and say that a set S' has the *minimum overlap property* if each set in $\mathcal{C}_{red}$ contains at least one element from S'. The set S' has the *maximum overlap property* if each set in $\mathcal{C}_{blue}$ contains at most k elements from S'. Thus, a set S' that has both the minimum and maximum overlap property constitutes a valid solution to the given instance of Red-Blue Hitting Set.

2 Red-Blue Hitting Set Without C.O.P.

This section deals with the general RBHS problem, meaning that we make no requirement for $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ concerning the c.o.p.. Being a generalization of MMSC, RBHS is of course NP-complete in general. This even holds for some rather strongly restricted variants, as the next theorem shows.

Theorem 2. RBHS *is* NP*-complete even if the following restrictions apply:*

1. *The collection $\mathcal{C}_{blue}$ contains exactly one set, and*
2. *each set in $\mathcal{C}_{red}$ has cardinality 2.*

Proof. We show the theorem by a reduction from the NP-complete VERTEX COVER problem. Given a graph $G = (V, E)$ and a nonnegative integer k, this problem asks to find a size-k subset $V' \subseteq V$ such that for every edge in E, at least one of its endpoints is in V'. Given an instance (G, k) of VERTEX COVER, construct an instance of RBHS by setting $S := V$, $\mathcal{C}_{red} := E$, $\mathcal{C}_{blue} := \{V\}$ (that is, the collection $\mathcal{C}_{blue}$ consists of one set containing all elements of S), and setting the maximum overlap equal to k. It is easy to see that this instance of RBHS directly corresponds to the original vertex cover instance: We may choose at most k elements from S to be in the solution set S' such that at least one element from every set in $\mathcal{C}_{red}$ is contained in S'. □

As shown in the next theorem, polynomial-time solvable instances of RBHS arise when the cardinalities of all sets in the collection $\mathcal{C}_{red}$ are restricted to 2 and the maximum overlap $k = 1$.

Theorem 3. RBHS *can be solved in polynomial time if the maximum overlap $k = 1$ and all sets in $\mathcal{C}_{red}$ have cardinality at most 2.*

Proof. We prove the theorem by showing how the restricted RBHS instance can equivalently be stated as a 2-SAT problem; 2-SAT is well-known to be solvable in linear time [6].

For our reduction, we construct the following instance F of 2-SAT for a given instance $(S, \mathcal{C}_{red}, \mathcal{C}_{blue}, 1)$ of RBHS:

- For each element $s_i \in S$, where $1 \leq i \leq n$, F contains the variable x_i.
- For each set $\{s_{i_1}, s_{i_2}\} \in \mathcal{C}_{red}$, F contains the clause $(x_{i_1} \vee x_{i_2})$.
- For each set $\{s_{i_1}, \ldots, s_{i_d}\} \in \mathcal{C}_{blue}$, F contains $d(d-1)/2$ clauses $(\neg x_{i_a} \vee \neg x_{i_b})$ with $1 \leq a < b \leq d$.

If the resulting Boolean formula F has a satisfying truth assignment T, then $S' := \{s_i \in S : T(x_i) = \mathit{true}\}$ is a solution to the RBHS instance $(S, \mathcal{C}_{red}, \mathcal{C}_{blue}, 1)$: By construction, for each set in $\mathcal{C}_{red}$ at least one element must have been chosen in order to satisfy the corresponding clause. Hence, S' has the minimum overlap property. Also, no two elements s_{i_a}, s_{i_b} from a set in $\mathcal{C}_{blue}$ can have been chosen because this would imply that the corresponding clause

$(\neg x_{i_a} \vee \neg x_{i_b})$ in F is not satisfied by T. Hence, S' also has the maximum overlap property and thus is a valid solution to the RBHS instance.

Omitting a formal proof here, it is easy to see that a solution set S' for $(S, \mathcal{C}_{red}, \mathcal{C}_{blue}, 1)$ can be used to construct a satisfying truth assignment T for F: For all $1 \leq i \leq n$, set $T(x_i) = \mathit{true}$ if $s_i \in S'$ and $T(x_i) = \mathit{false}$ otherwise. □

Corollary 4. RBHS *can be solved in linear time if all sets in* $\mathcal{C}_{red}$ *and* $\mathcal{C}_{blue}$ *have cardinality at most 2.* □

3 Red-Blue Hitting Set With Partial C.O.P.

In this section, we prove that RBHS remains NP-complete even under the requirement that either $\mathcal{C}_{red}$ or $\mathcal{C}_{blue}$ is to have the c.o.p.. To this end, we give reductions from the following restricted variant of the SATISFIABILITY problem:

> RESTRICTED 3-SAT (R3-SAT)
> **Input:** An n-variable Boolean formula F in conjunctive normal form where each variable x_i, $1 \leq i \leq n$, appears at most three times, each literal appears at most twice, and each clause contains at most three literals.
> **Task:** Determine if there exists a satisfying truth assignment T for F.

It is well-known that R3-SAT is NP-complete (e.g., see [15, p. 183]).[2]

3.1 Consecutive Ones Property for $\mathcal{C}_{red}$

The following two theorems (Theorems 5 and 6) show that the requirement of $\mathcal{C}_{red}$ obeying the c.o.p. does not make RBHS tractable. The theorems complement each other in the sense that they impose different restrictions on the cardinalities of the sets $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$; Theorem 5 allows for size-3 sets in $\mathcal{C}_{red}$ and size-2 sets in $\mathcal{C}_{blue}$ (the reduction encodes clauses of a given R3-SAT instance in $\mathcal{C}_{red}$) while the converse holds true for Theorem 6 (the reduction encodes variables in $\mathcal{C}_{red}$).

Theorem 5. RBHS *is* NP*-complete even if all the following restrictions apply:*

1. *The collection* $\mathcal{C}_{red}$ *has the consecutive ones property.*
2. *The maximum overlap* k *is equal to one.*
3. *Each set in* $\mathcal{C}_{red}$ *has cardinality 3, and each set in* $\mathcal{C}_{blue}$ *has cardinality 2.*
4. *Each element from* S *occurs in exactly one set in* $\mathcal{C}_{red}$*, and each element from* S *occurs in at most two sets in* $\mathcal{C}_{blue}$*.*

Proof. We prove the theorem by a reduction from R3-SAT. Given an m-clause Boolean formula F that is an instance of R3-SAT, construct the following instance $(S, \mathcal{C}_{red}, \mathcal{C}_{blue}, k)$ of RBHS:

[2] Note that it is essential for the NP-completeness of R3-SAT that the Boolean formula F may contain size-2 clauses, otherwise, the problem is in P [15].

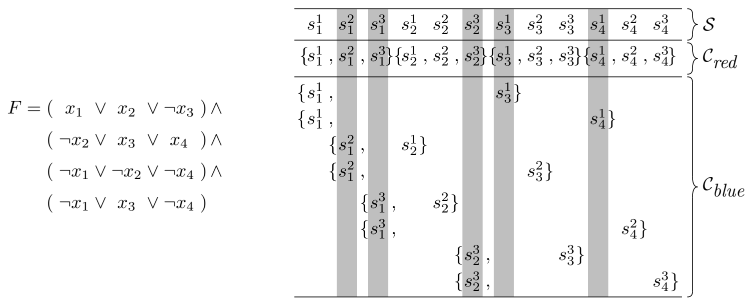

Fig. 1. Example of encoding an instance of R3-Sat into an instance of RBHS (proof of Theorem 5). Each clause of the Boolean formula F is represented by a three-element set in $\mathcal{C}_{red}$. The sets in $\mathcal{C}_{blue}$ and the maximum overlap $k = 1$ ensure that no two elements from S can be chosen into a solution that correspond to conflicting truth assignments of the same variable. Observe how $S' = \{s_1^2, s_1^3, s_2^3, s_3^1, s_4^1\}$ (grey columns) constitutes a valid solution to the RBHS instance; accordingly, a truth assignment T which makes all the corresponding literals evaluate to *true* satisfies F.

- The set S consists of elements $s_1^1, s_1^2, s_1^3, \ldots, s_m^1, s_m^2, s_m^3$. The element s_j^i corresponds to the i-th literal in the j-th clause of F. If the j-th clause has only two literals, then S contains only s_j^1 and s_j^2.
- Each set in $\mathcal{C}_{red}$ corresponds to a clause in F, that is, for the i-th clause in F, we add $\{s_i^1, s_i^2, s_i^3\}$ to $\mathcal{C}_{red}$ if it contains three literals and $\{s_i^1, s_i^2\}$ if it contains two literals.
- For each variable x in F and for all pairs of literals $l_1 = x, l_2 = \neg x$ in F: If l_1 is the i-th literal in the j-th clause and l_2 is the p-th literal in the q-th clause of F, $\mathcal{C}_{blue}$ contains the set $\{s_j^i, s_q^p\}$.
- The maximum overlap k is set to one.

The construction is illustrated in Figure 1. It is easy to see that, by the definition of R3-Sat, the constructed instance satisfies the restrictions claimed in the theorem; note that $\mathcal{C}_{red}$ has the consecutive ones property due to Observation 1. It remains to show that the constructed instance of RBHS has a solution iff F has a satisfying truth assignment T.

"$\Rightarrow$" Assume that the constructed instance of RBHS has a solution set S'. Let T be a truth assignment such that, for every $s_j^i \in S'$, the variable represented by s_j^i is set to *true* if the literal represented by s_j^i is positive, and *false* otherwise. This truth assignment is well defined because S' must have the maximum overlap property—it therefore cannot happen that two elements $s_j^i, s_q^p \in S'$ correspond to different literals of the same variable.

To show that T constitutes a satisfying truth assignment for F, observe that, for each clause of F, at least one element from S' must correspond to a literal in this clause because S' has the minimum overlap property. On the one hand, if this element corresponds to a positive literal x_i, then $T(x_i) = \mathit{true}$, satisfying the clause. On the other hand, if the element corresponds to a negative literal $\neg x_i$, then $T(x_i) = \mathit{false}$, satisfying the clause.

"$\Leftarrow$" Let T be a satisfying truth assignment for F. Let S' be the set of elements in S that correspond to literals that evaluate to *true* under T. Then, S' has the minimum overlap property because at least one literal in every clause of F must evaluate to true under T and each set in $\mathcal{C}_{red}$ represents exactly one clause of F. Also, S' has the maximum overlap property because T is uniquely defined for every variable that occurs in F. Since S' has both the minimum and maximum overlap property, it is a valid solution to the RBHS instance. □

The following theorem can be proven in a similar way as Theorem 5.

Theorem 6. RBHS *is* NP*-complete even if all the following restrictions apply:*

1. *The collection $\mathcal{C}_{red}$ has the consecutive ones property.*
2. *The maximum overlap k is equal to two.*
3. *Each set in $\mathcal{C}_{red}$ has cardinality 2, and each set in $\mathcal{C}_{blue}$ has cardinality 3.*
4. *Each element from S occurs in exactly one set in $\mathcal{C}_{red}$, and each element from S occurs in at most two sets in $\mathcal{C}_{blue}$.* □

3.2 Consecutive Ones Property for $\mathcal{C}_{blue}$

Note that by the proof of Theorem 2, RBHS is NP-complete already if $\mathcal{C}_{blue}$ contains just a single set and, hence, has the c.o.p.. However, this requires a non-fixed maximum overlap k and unrestricted cardinality of the sets contained in $\mathcal{C}_{blue}$. Therefore, if we want to show the NP-hardness of RBHS with the additional restriction that the maximum overlap k is fixed and the sets in $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ have small cardinality, another reduction is needed. Analogously to Theorems 5 and 6, the following two theorems impose different restrictions on the cardinalities of the sets in $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$.

Theorem 7. RBHS *is* NP*-complete even if all the following restrictions apply:*

1. *The collection $\mathcal{C}_{blue}$ has the consecutive ones property.*
2. *The maximum overlap k is equal to one.*
3. *Each set in $\mathcal{C}_{red}$ has cardinality 3, and each set in $\mathcal{C}_{blue}$ has cardinality 2.*
4. *Each element from S occurs in at most two sets in $\mathcal{C}_{red}$, and each element from S occurs in exactly one set in $\mathcal{C}_{blue}$.*

Proof. Again, we give a reduction from R3-Sat. For a given n-variable Boolean formula F that is an instance of R3-Sat, construct the following instance $(S, \mathcal{C}_{red}, \mathcal{C}_{blue}, k)$ of RBHS:

- The set S consists of $2n$ elements $s_1, \bar{s}_1, \ldots, s_n, \bar{s}_n$, that is, for each variable x_i in F, S contains an element s_i representing the literal x and an element $\bar{s}_i$ representing the literal $\neg x$.
- For each clause in F, $\mathcal{C}_{red}$ contains a set of those elements from S that represent the literals of that clause.

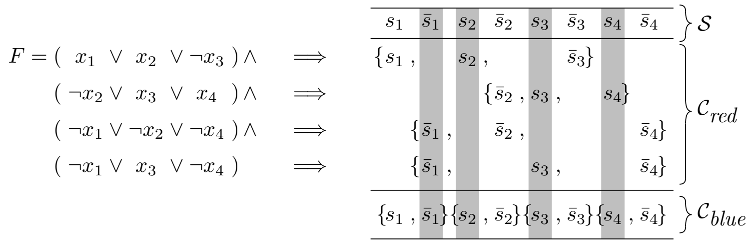

Fig. 2. Example of encoding an instance of R3-Sat into an instance of RBHS (proof of Theorem 7). Each clause of the Boolean Formula F is encoded into one set of $\mathcal{C}_{red}$. The sets in $\mathcal{C}_{blue}$ and the maximum overlap $k = 1$ ensure that no two elements from S can be chosen into a solution that correspond to conflicting truth assignments of the same variable. Observe how $S' = \{\bar{s}_1, s_2, s_3, s_4\}$ (grey columns) constitutes a valid solution to the RBHS instance; accordingly, a truth assignment T with $T(x_i) = \mathit{true}$ iff $s_i \in S'$ satisfies F.

- $\mathcal{C}_{blue} = \bigcup_{1 \leq i \leq n} \{\{s_i, \bar{s}_i\}\}$.
- The maximum overlap k is one.

Observe that this instance satisfies all restrictions claimed in the theorem; $\mathcal{C}_{blue}$ has the c.o.p. due to Observation 1. The reduction is illustrated by an example in Figure 2. It remains to show that the constructed instance has a solution iff F has a satisfying truth assignment T. We omit the details. □

The proof of the following theorem is similar to the one of Theorem 7.

Theorem 8. RBHS *is* NP*-complete even if all the following restrictions apply:*

1. *The collection $\mathcal{C}_{blue}$ has the consecutive ones property.*
2. *The maximum overlap k is equal to two.*
3. *Each set in $\mathcal{C}_{red}$ has cardinality 2, and each set in $\mathcal{C}_{blue}$ has cardinality 3.*
4. *Each element from S occurs in at most two sets in $\mathcal{C}_{red}$, and each element from S occurs in at exactly one set in $\mathcal{C}_{blue}$.* □

Note that if k is restricted to $k = 1$ instead of $k = 2$ in the instances discussed in the above theorem, they become polynomial-time solvable according to Theorem 3.

4 Red-Blue Hitting Set With C.O.P.

In this section, we make the requirement that both $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ in a given instance $(S, \mathcal{C}_{red}, \mathcal{C}_{blue}, k)$ of RBHS obey the c.o.p. and call the resulting problem "RBHS with c.o.p.." We present an approximation algorithm (Section 4.1) and show, among others, that for fixed k RBHS with c.o.p. is solvable in polynomial time (Section 4.2). This leads to the following observation.

Observation 2 RBHS *with c.o.p. is equivalent to* MMSC *with c.o.p..*

To see this observation, note that on the one hand, MMSC is obviously the special case of RBHS with identical red and blue subset collections. On the other hand, an RBHS instance $(S, \mathcal{C}_{red}, \mathcal{C}_{blue}, k)$ with c.o.p. can be transformed into an MMSC instance with c.o.p. by observing that for an optimal solution S' of RBHS, for any set $C_b \in \mathcal{C}_{blue}$ that contains no set from $\mathcal{C}_{red}$ as a subset we have $|C_b \cap S'| \leq 2$. Thus, if $k > 2$, then we can safely remove such blue subsets from $\mathcal{C}_{blue}$. Then, solving RBHS on the resulting instance is equivalent to solving MMSC on the instance $(S, \mathcal{C}_{red} \cup \mathcal{C}_{blue}, k)$. If $k \leq 2$ then both RBHS and MMSC with c.o.p. are solvable in polynomial time as it will be shown in Section 4.2.

To simplify the discussion in this section, we assume that the elements in $S = \{s_1, \ldots, s_n\}$ are sorted such that all subsets in $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ have the c.o.p., that is, for every $1 \leq i \leq k \leq j \leq n$ and every set $C \in \mathcal{C}_{blue} \cup \mathcal{C}_{red}$ it holds that $s_i \in C \wedge s_j \in C \Rightarrow s_k \in C$. For each subset $C \subseteq S$, its *left index* $l(C)$ is defined as $\min\{i \mid s_i \in C\}$ and its *right index* $r(C)$ is defined as $\max\{i \mid s_i \in C\}$.

4.1 Approximation Algorithm

Here, we describe a polynomial-time approximation algorithm for RBHS with c.o.p. that has a guaranteed additive term of one compared to an optimal solution. To this end, we rephrase RBHS as an optimization problem:

Input: A set S and two collections $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ of subsets of S.
Task: Find a subset $S' \subseteq S$ with $S' \cap C \neq \emptyset$ for all $C \in \mathcal{C}_{red}$ which minimizes $\max_{C' \in \mathcal{C}_{blue}}\{|C' \cap S'|\}$.

Our greedy approximation algorithm works as follows:

$S' \leftarrow \emptyset$, $\mathcal{C}'_{red} \leftarrow \mathcal{C}_{red}$
while $\mathcal{C}'_{red} \neq \emptyset$
  $C \leftarrow$ set from $\mathcal{C}'_{red}$ with minimum right index
  $S' \leftarrow S' \cup \{s_{r(C)}\}$, $\mathcal{C}'_{red} \leftarrow \mathcal{C}'_{red} \setminus \{C \in \mathcal{C}'_{red} : C \cap S' \neq \emptyset\}$
return S'

Theorem 9. *For* RBHS *with c.o.p., the greedy algorithm polynomial-time approximates an optimum solution within an additive term of one.*

Proof. Obviously, the output S' of the greedy algorithm has the minimum overlap property since $S' \cap C \neq \emptyset$ for all $C \in \mathcal{C}_{red}$. It is also clear that the algorithms runs in $O(|S| \cdot |\mathcal{C}_{red}|)$ time. It remains to show the additive term.

Let C denote one subset in $\mathcal{C}_{blue}$ with $|C \cap S'| = \max_{C' \in \mathcal{C}_{blue}}\{|C' \cap S'|\}$. It is easy to observe that C contains at least $|C \cap S'| - 1$ mutually disjoint sets from $\mathcal{C}_{red}$ as subsets, implying that *any* solution for this instance has to contain at least $|C \cap S'| - 1$ elements from C in order to satisfy the minimum overlap property for these sets. Therefore, $|C \cap S'_{\text{opt}}| \geq |C \cap S'| - 1$ for any optimal solution S'_{opt}. □

4.2 Dynamic Programming

We now present a dynamic programming algorithm that solves Red-Blue Hitting Set with c.o.p. in polynomial time provided that either the maximum overlap is a fixed constant k, the maximum cardinality of the sets in $\mathcal{C}_{blue}$ is a fixed constant c_{card}, or the maximum number of occurrences of an element in $\mathcal{C}_{blue}$ is a fixed constant c_{occ}.

We assume that the sets $C \in \mathcal{C}_{red}$ are ordered according to $l(C)$ and denote them with $R_1, \ldots, R_{|\mathcal{C}_{red}|}$; the sets of $\mathcal{C}_{blue}$ are ordered analogously and denoted with $B_1, \ldots, B_{|\mathcal{C}_{blue}|}$. If a set in $\mathcal{C}_{red}$ is a superset of another set in $\mathcal{C}_{red}$ it can be removed, and, therefore, for any two sets $R_i, R_j \in \mathcal{C}_{red}$ it holds that $l(R_i) < l(R_j) \Leftrightarrow r(R_i) < r(R_j)$. For an analogous reason we have $l(B_i) < l(B_j) \Leftrightarrow r(B_i) < r(B_j)$ for all $B_i, B_j \in \mathcal{C}_{blue}$. We call this property the *monotonicity* of the sets in $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$.

The idea of the dynamic programming algorithm is to build collections $D(i, j)$ of so-called partial solutions; each partial solution in a collection $D(i, j)$ covers all sets $R_1, \ldots, R_j$ with a minimal subset of $\{s_1, \ldots, s_i\}$ that fulfills the maximum overlap property. To this end, the algorithm uses a two-dimensional table $D(i, j)$ with $1 \leq i \leq n$ (where $n := |S|$) and $1 \leq j \leq |\mathcal{C}_{red}|$; to fill this table, two nested loops are used, the outer one iterating over i and the inner one iterating over j. Every entry of $D(i, j)$ that already has been processed contains a collection of tuples $(S_h, v_h), 1 \leq h \leq |D(i, j)|$, where each tuple (S_h, v_h) consists of a set $S_h \subseteq \{s_1, \ldots, s_i\}$ and a vector $v_h = (v_h^1, \ldots, v_h^{|\mathcal{C}_{blue}|})$. The sets S_h are called *partial solutions* and have the following properties:

1. Each S_h contains at least one element of every set $R_1, \ldots, R_j \in \mathcal{C}_{red}$.
2. No proper subset of a partial solution S_h covers all sets $R_1, \ldots, R_j$.
3. For $1 \leq q \leq |\mathcal{C}_{blue}|$, we have $v_h^q = |S_h \cap B_q| \leq k$.

It is obvious that if the entry $D(n, |\mathcal{C}_{red}|)$ is not empty, each partial solution in $D(n, |\mathcal{C}_{red}|)$ is a solution for the RBHS instance.

The first step for filling the table is to compute all entries $D(i, j)$ with $i = 1$, a trivial task. All other entries $D(i, j)$ are computed as follows: If $l(R_j) > i$ then $D(i, j)$ is empty. Otherwise, the partial solutions that have to be generated can be divided in two categories: Partial solutions not containing the element s_i and partial solutions containing s_i. The partial solutions not containing s_i can only contain elements from $\{s_1, \ldots, s_{i-1}\}$ and, therefore, are exactly the partial solutions of $D(i - 1, j)$.

The partial solutions in $D(i, j)$ that do contain s_i are computed as follows: By selecting s_i to be member of such a partial solution in $D(i, j)$, all sets in $\mathcal{C}_{red}$ that contain s_i are covered. Therefore, the other elements in the partial solution only have to cover those sets $R_p \in \{R_1, \ldots, R_j\}$ with $r(R_p) < i$. Hence, these elements form a partial solution in a collection $D(i - 1, j')$ where j' is the maximum possible index such that $r(R_{j'}) < i$. More formally, an entry $D(i, j)$ with $i > 1$ and $l(R_j) \leq i$ is computed as follows:

if $D(i-1,j) \neq \emptyset$ **then** $D(i,j) \leftarrow D(i-1,j)$.
$j' \leftarrow \max\{p \in \{1,\ldots,j\} \mid r(R_p) < i\}$
if $D(i-1,j') \neq \emptyset$ **then for each** $(S_h, v_h) \in D(i-1,j')$ **do**
    Insert a copy $(\tilde{S}_h, \tilde{v}_h)$ of (S_h, v_h) into $D(i,j)$
    $\tilde{S}_h \leftarrow \tilde{S}_h \cup \{s_i\}$
    for each $q \in \{1,\ldots,|\mathcal{C}_{blue}| : s_i \in B_q\}$ **do**
        $\tilde{v}_h^q \leftarrow \tilde{v}_h^q + 1$
        if $\tilde{v}_h^q > k$ **then** delete $(\tilde{S}_h, \tilde{v}_h)$ from $D(i,j)$
            and continue with the next tuple in $D(i-1,j')$

In order to shrink the table size by eliminating redundant tuples, we perform the following *data reduction step* directly after the computation of an entry $D(i,j)$:

$q' \leftarrow \max\{q \in \{1,\ldots,|\mathcal{C}_{blue}|\} : r(B_q) \leq i\}$
while $D(i,j)$ contains tuples (S_{h_1}, v_{h_1}) and (S_{h_2}, v_{h_2}) such that
    $r(S_{h_1}) \geq r(S_{h_2})$ and $\forall\, q > q' \ : \ v_{h_1}^q \leq v_{h_2}^q$ **do**
    Delete (S_{h_2}, v_{h_2}) from $D(i,j)$

The data reduction step step does not affect the correctness of the algorithm: If there is a solution S' for the RBHS instance with $S' = S_{h_2} \cup \hat{S}$ such that $l(\hat{S}) > i$, then $\hat{S} \cup S_{h_1}$ is obviously also a solution.

The following lemma helps us to give an upper bound for the number of tuples (S_h, v_h) in a collection $D(i,j)$. We omit the proof.

Lemma 10. *For a collection $D(i,j) \neq \emptyset$, let $D_x(i,j)$ be the tuples $(S_h, v_h) \in D(i,j)$ with $r(S_h) = x$. Then the number of tuples in $D_x(i,j)$ is bounded from above by* $\min\{(k+1)^{c_{occ}}, \binom{c_{occ}+k}{k}\}$. □

Theorem 11. RBHS *with c.o.p. can be solved in polynomial time provided that either the maximum overlap k is a fixed constant, the maximum cardinality of the sets in $\mathcal{C}_{blue}$ is a fixed constant c_{card}, or the maximum number of occurrences of an element in $\mathcal{C}_{blue}$ is a fixed constant c_{occ}. More precisely,* RBHS *is solvable either in $|S|^{O(k)}$ time or in $|S|^{O(c_{occ})}$ time or in $|S|^{O(1)} \cdot c_{card}^{O(c_{card})}$ time.*

Proof. The correctness of the algorithm follows from its above description. It remains to show the running time, which basically depends on the size of the table (that is, the number of collections $D(i,j)$) and the number of tuples that have to be compared during the data reduction step.

The table size is $|S| \cdot |\mathcal{C}_{red}| \leq |S|^2$. An upper bound for the number of tuples in each collection can be derived from Lemma 10. The claimed running times follow, because c_{card} is bounded from above by $|S|$, k is bounded from above by $c_{card} - 1$, and c_{occ} is bounded from above by $|\mathcal{C}_{blue}|$ and by c_{card}. □

Corollary 12. RBHS *with c.o.p. can be solved in polynomial time if $|\mathcal{C}_{blue}|$ is a constant.* □

5 Conclusion

In this work, we initiated a study of Minimum Membership Set Cover and, more generally, Red-Blue Hitting Set with respect to instances (partially) obeying the consecutive ones property. Many natural challenges for future work arise from our results. For instance, it is desirable to find out more about the polynomial-time approximability and the parameterized complexity [5, 14] of the variants of Red-Blue Hitting Set proven to be NP-complete (see Table 1). Moreover, in three cases Table 1 exhibits unsettled questions concerning the computational complexity of the respective problems.

References

1. G. Ausiello, A. D'Atri, and M. Protasi. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences*, 21(1):136–153, 1980.
2. A. Caprara, P. Toth, and M. Fischetti. Algorithms for the set covering problem. *Annals of Operations Research*, 98:353–371, 2000.
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.
4. E. D. Demaine, M. T. Hajiaghayi, U. Feige, and M. R. Salavatipour. Combination can be hard: approximability of the unique coverage problem. In *Proc. 17th SODA*, pages 162–171. ACM Press, 2006.
5. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.
6. S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976.
7. U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.
8. N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997.
9. J. Guo and R. Niedermeier. Exact algorithms and applications for Tree-like Weighted Set Cover. *Journal of Discrete Algorithms*, 2006. To appear.
10. F. Kuhn, P. von Rickenbach, R. Wattenhofer, E. Welzl, and A. Zollinger. Interference in cellular networks: The minimum membership set cover problem. In *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 188–198. Springer, 2005.
11. S. Mecke and D. Wagner. Solving geometric covering problems by data reduction. In *Proc. 12th ESA*, volume 3221 of *LNCS*, pages 760–771. Springer, 2004.
12. J. Meidanis, O. Porto, and G. Telles. On the consecutive ones property. *Discrete Applied Mathematics*, 88:325–354, 1998.
13. G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.
14. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
15. C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
16. N. Ruf and A. Schöbel. Set covering with almost consecutive ones property. *Discrete Optimization*, 1(2):215–228, 2004.
17. A. F. Veinott and H. M. Wagner. Optimal capacity scheduling. *Operations Research*, 10:518–532, 1962.