Originally published in *Journal of Discrete Algorithms*, 6(3):393–407. Elsevier B. V, 2008.

Red-Blue Covering Problems and the Consecutive Ones Property [1]

Michael Dom, Jiong Guo [2], Rolf Niedermeier, Sebastian Wernicke [3]

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

Abstract

Set Cover problems are of core importance in many applications. In recent research, the "red-blue variants" where blue elements all need to be covered whereas red elements add further constraints on the optimality of a covering have received considerable interest. Application scenarios range from data mining to interference reduction in cellular networks. As a rule, these problem variants are computationally at least as hard as the original set cover problem. In this work we investigate whether and how the well-known consecutive ones property, restricting the structure of the input sets, makes the red-blue covering problems feasible. We explore a sharp border between polynomial-time solvability and NP-hardness for these problems.

Key words: Set Cover, Hitting Set, Minimum Membership Set Cover, Minimum Degree Hypergraph, NP-completeness

Originally published in *Journal of Discrete Algorithms*, 6(3):393–407. Elsevier B. V, 2008.

1 Introduction

Motivation and Definitions. Covering problems are of central importance in algorithm theory and combinatorial optimization. Two of the most prominent examples for this type of problem are Set Cover and Hitting Set. In both problems, the input consists of a set S and a collection $\mathcal{C}$ of subsets of S. For Set Cover, one tries to find a minimum-size subcollection $\mathcal{C}' \subseteq \mathcal{C}$ that *covers* S, that is, it satisfies $\bigcup_{C \in \mathcal{C}'} C = S$. For Hitting Set, one tries to find a minimum-size subset $S' \subseteq S$ that *covers* $\mathcal{C}$, that is, each set in $\mathcal{C}$ contains at least one element from S'. It is well-known that both problems are equivalent in this general setting [3]. Due to their practical importance, there is a lot of literature on Set Cover and Hitting Set [6,8]. Set Cover is NP-complete and only allows for a logarithmic-factor polynomial-time approximation [14]. It is parameterized intractable (that is, W[2]-complete) with respect to the parameter "solution size" [12]. Due to the equivalence between Set Cover and Hitting Set, these results also apply to Hitting Set. [4]

Generalizations as well as restrictions of Set Cover and Hitting Set played a prominent role in algorithmics. In this work, we are going to study two covering problems with an important generalization called "red-blue" together with an important restriction called "consecutive ones property" which we apply to both problems.

The first covering problem is called Minimum Degree Hypergraph (MDH) and is defined as follows:

---

[1] A preliminary version of this paper appeared under the title "Minimum Membership Set Covering and the Consecutive Ones Property" in the proceedings of the 10th Scandinavian Workshop on Algorithm Theory (SWAT 2006), held in Riga, Latvia, July 2006 [10]. Note that we changed the title due to significant changes in comparison with the conference version. First, a dynamic programming algorithm for Minimum Degree Hypergraph (we called this problem Red-Blue Hitting Set in the preliminary version) with consecutive ones property has been replaced by a solution based on integer linear programming. Moreover, a dynamic programming algorithm for Red-Blue Set Cover with consecutive ones property has been added. Second, we added further NP-completeness results concerning Red-Blue Set Cover with consecutive ones property. Finally, in accordance with previous literature, the meanings of red and blue sets in instances of Minimum Degree Hypergraph have been interchanged.

[2] Supported by the Deutsche Forschungsgemeinschaft, Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4.

[3] Supported by the Deutsche Telekom Stiftung.

[4] Generally, a set cover problem, where elements have to be covered by sets, can be equivalently formulated as a hitting set problem, where sets have to be covered by elements, by simply exchanging elements and sets.

MINIMUM DEGREE HYPERGRAPH (MDH)
Input: A set S, two collections $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$ of subsets of S, and a nonnegative integer k.
Task: Determine if there exists a subset $S' \subseteq S$ such that

$$\forall C \in \mathcal{C}_{blue} : |S' \cap C| \geq 1, \text{ and } \forall C \in \mathcal{C}_{red} : |S' \cap C| \leq k.$$

Feder et al. [13] introduced this problem and gave a factor-$O(\log |S|)$ polynomial-time approximation algorithm for it. Motivated by applications concerning interference reduction in cellular networks, Kuhn et al. [22] introduced the MINIMUM MEMBERSHIP SET COVER problem, a special case of MDH. Here, given a set S and a collection $\mathcal{C}$ of subsets of S, one wants to determine a subcollection $\mathcal{C}' \subseteq \mathcal{C}$ that covers S but where the maximum number of occurrences of each element from S in the subsets in $\mathcal{C}'$ shall be minimized. MMSC is the special case of MDH where $\mathcal{C}_{blue} = \mathcal{C}_{red}$.

Our second covering problem within the "red-blue setting", the so-called RED-BLUE SET COVER (RBSC) problem, has been introduced by Carr et al. [7] and is defined as follows.

Input: Two disjoint sets B (blue elements) and R (red elements), a collection $\mathcal{C}$ of subsets of $B \cup R$, and a nonnegative integer k.
Task: Determine if there exists a subcollection $\mathcal{C}' \subseteq \mathcal{C}$ such that

$$\forall b \in B\ \exists C \in \mathcal{C}' : b \in C, \text{ and } |(\bigcup_{C \in \mathcal{C}'} C) \cap R| \leq k.$$

SET COVER is the special case of RBSC where each set in $\mathcal{C}$ contains exactly one red element and no red element is contained in more than one set. Carr et al. provided several natural application scenarios such as data mining for RBSC and several positive and negative results concerning the polynomial-time approximability of RBSC. A further problem connected to RBSC is the GENERALIZED VENETIAN ROUTING problem dealing with wavelength routing in optical networks [5].

To emphasize the close relationship between RBSC and MDH, we present the following, equivalent definition of RBSC [5]. This definition will be made use of in the remainder of this paper.

RED-BLUE SET COVER (RBSC)
Input: A set S, two collections $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$ of subsets of S, and a nonnegative integer k.

[5] This equivalence can be seen, similar to the equivalence between SET COVER and HITTING SET, by exchanging elements and sets. The sets B and R in the original definition correspond to the collections $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$ in our equivalent formulation.

$$\begin{array}{cccc} A & B & C & D \\ \hline 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ \hline \end{array} \qquad \begin{array}{cccc} C & A & D & B \\ \hline 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ \hline \end{array} \qquad \begin{array}{|cccc|} \hline 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline \end{array}$$

Fig. 1. Example for the C1P: The matrix on the left has the C1P, because by permuting its columns (labeled with A–D) one can obtain the matrix shown in the middle where the ones in each row appear consecutively. The matrix on the right, in contrast, does not have the C1P [33].

Task: Determine if there exists a subset $S' \subseteq S$ such that

$$\forall C \in \mathcal{C}_{blue} : |S' \cap C| \geq 1, \text{ and } |\{C \in \mathcal{C}_{red} \mid S' \cap C \neq \emptyset\}| \leq k.$$

The difference between RBSC and MDH is that in the case of RBSC the number of red sets containing elements of the solution set is restricted, whereas in the case of MDH the maximum number of elements of a red set being contained in the solution set is restricted.

As to the consecutive ones property (C1P), there is a long history of research [35,33,25,26,24,30,23,32,9,11]. Applied to instances of the problems MDH and RBSC, the C1P means that the elements of S can be ordered in a linear arrangement such that each set in $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$ contains only a whole "chunk" of that arrangement, that is, without any gaps. The name "consecutive ones" refers to the fact that one may think of an MDH or RBSC instance as a coefficient matrix M where the elements in the ground set correspond to columns and the sets in the subset collection correspond to rows; an entry is 1 if the respective element is contained in the respective set, and 0, otherwise. If an instance has the C1P, then the columns of M can be permuted in such a way that the ones in each row appear consecutively as Figure 1 illustrates. Set Cover instances with the C1P are solvable in polynomial time, a fact which is made use of in many practical applications [24,23,26,30,35]. In applications of MDH or RBSC with geographic background (such as the interference reduction considered by Kuhn et al. [22]), the problem instances may have the C1P or be "close" to the C1P [23,24]. Katz et al. [21] recently considered geometric Set Cover problems that are also related to covering problems under the C1P restriction.

Contributions. Seemingly for the first time, this work brings together the concepts of "red-blue" and the C1P, that is, we investigate the time complexity of the two red-blue covering problems with the C1P. The formulations of MDH and RBSC open a wide field of natural investigations concerning the C1P, the point being that the C1P may apply to either $\mathcal{C}_{blue}$, $\mathcal{C}_{red}$, $\mathcal{C}_{blue} \cup \mathcal{C}_{red}$, or none of $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$.

On the positive side, we show polynomial-time solvability for MDH and RBSC in the case that $\mathcal{C}_{blue} \cup \mathcal{C}_{red}$ possesses the C1P. In addition, we provide a simple greedy algorithm that approximates RBSC with $\mathcal{C}_{blue} \cup \mathcal{C}_{red}$ having the C1P to an additive term of one. On the negative side, we prove several NP-completeness results in case that at most one of $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ has the C1P. More specifically, we indicate several sharp borders between polynomial-time solvability and NP-completeness of MDH depending on the subset sizes (the main point being, roughly speaking, a distinction between subset sizes two and three, see Corollary 8). Moreover, we show that if at most one of $\mathcal{C}_{red}$ and $\mathcal{C}_{blue}$ has the C1P, then also RBSC becomes NP-complete.

2 Preliminaries and Basic Observations

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 (C1P) *if there exists a linear 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 $s_i \in C \wedge s_j \in C \Rightarrow s_k \in C$.*

Given a subset system $(S, \mathcal{C})$, the linear order $\prec$ in Definition 1 can be found in $O(|S| + |\mathcal{C}| + \sum_{C \in \mathcal{C}} |C|)$ time [4,19]. Therefore, in all our algorithmic results except Theorem 6 we can without loss of generality assume that the elements of the set S in the input are already sorted according to the order $\prec$.

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 C1P.*

We say that a set $S' \subseteq S$ has the *minimum overlap property* if each set in $\mathcal{C}_{blue}$ contains at least one element from S'. Moreover, for a given instance $(S, \mathcal{C}_{blue}, \mathcal{C}_{red}, k)$ of MDH and RBSC, we will call k the *maximum overlap* and the *maximum containment*, respectively. A set S' has the *maximum overlap property* if each set in $\mathcal{C}_{red}$ contains at most k elements from S'. Analogously, a set S' has the *maximum containment property* if at most k sets in $\mathcal{C}_{red}$ contain elements from S'.

As it is easy to see that the problems considered in this paper are contained in NP, all our NP-completeness proofs will only show the NP-hardness of the corresponding problems.

We continue with two observations concerning MDH without C1P. Being a generalization of Set Cover, MDH is of course NP-complete in general. This even holds for a rather strongly restricted variant:

Observation 2 MDH *is* NP-*complete even if* $|\mathcal{C}_{red}| = 1$ *and* $\forall C \in \mathcal{C}_{blue} : |C| = 2$.

The observation can be seen by a reduction from the NP-complete Vertex Cover problem [16]: 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 MDH by setting $S := V$, $\mathcal{C}_{blue} := E$, $\mathcal{C}_{red} := \{V\}$ (that is, the collection $\mathcal{C}_{red}$ consists of one set containing all elements of S), and setting the maximum overlap equal to k. The correctness of this construction is straightforward.

Polynomial-time solvable instances of MDH arise when the cardinalities of all sets in the collection $\mathcal{C}_{blue}$ are restricted to 2 and the maximum overlap $k = 1$:

Observation 3 MDH *can be solved in polynomial time if* $k = 1$ *and* $\forall C \in \mathcal{C}_{blue} : |C| \leq 2$.

This observation can be shown by stating the restricted MDH instance equivalently as a 2-Sat problem; 2-Sat is well-known to be solvable in linear time [1]. For the reduction, construct the following instance F of 2-Sat for a given instance $(S, \mathcal{C}_{blue}, \mathcal{C}_{red}, 1)$ of MDH:

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

Corollary 1 MDH *can be solved in polynomial time if* $\forall C \in \mathcal{C}_{blue} \cup \mathcal{C}_{red} : |C| \leq 2$.

To see this, first note that if $k \geq 2$ then the corresponding MDH instance is trivially solvable by setting $S' := S$, because then no set in $\mathcal{C}_{red}$ has more than k elements in common with the solution set S'. Hence, we only need to deal with the case $k = 1$, for which the claim is true by Observation 3.

Note that the restrictions imposed by Observation 3 and Corollary 1 are “tight.” If we allow $\mathcal{C}_{blue}$ to contain cardinality-3 subsets, then MDH becomes NP-complete (Theorems 9 and 11). If $\mathcal{C}_{red}$ contains cardinality-3 subsets and the maximum overlap is 2, then we can also prove the NP-completeness (Theorems 10 and 12).

3 Minimum Degree Hypergraph and Red-Blue Set Cover with C1P

In this section, we make the requirement that $\mathcal{C} := \mathcal{C}_{blue} \cup \mathcal{C}_{red}$ in a given instance $(S, \mathcal{C}_{blue}, \mathcal{C}_{red}, k)$ of MDH and RBSC obeys the C1P and call the resulting problems "MDH with C1P" and "RBSC with C1P."

By using known linear programming techniques, MDH with C1P can be solved in polynomial time; we will describe this approach in Sect. 3.1, followed by a much simpler greedy approximation algorithm in Sect. 3.2. The polynomial time solvability of RBSC with C1P is more difficult to see; for this problem we will present an exact polynomial time dynamic programming algorithm in Sect. 3.3.

To simplify our subsequent considerations, we assume that the elements in $S = \{s_1, \ldots, s_n\}$ are sorted such that all subsets in $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$ have the C1P. This sorting can be done in $O(|S| + |\mathcal{C}| + \sum_{C \in \mathcal{C}} |C|)$ time [4,19]. For each subset $C \in \mathcal{C}_{red} \cup \mathcal{C}_{blue}$, its *left index* $l(C)$ is defined as $\min\{i \mid s_i \in C\}$ and its *right index* $r(C)$ is defined as $\max\{i \mid s_i \in C\}$.

3.1 Linear Programming for Minimum Degree Hypergraph

Here we will first give a formulation of MDH with C1P as an integer linear program (ILP) and then explain two ways to solve this ILP in polynomial time. Refer to Schrijver [31] for basics about (integer) linear programming as we will need them here.

Given an instance of MDH with C1P, we introduce for each element $s_i \in S$ a variable x_i which, if set to 1, expresses that s_i has to be part of an optimal solution. Every integral feasible solution for the following integer linear program (ILP) then obviously yields a solution for MDH with C1P:

$$\begin{aligned} -x_{l(C)} - x_{l(C)+1} - \cdots - x_{r(C)} &\leq -1 \qquad \forall C \in \mathcal{C}_{blue} \\ x_{l(C)} + x_{l(C)+1} + \cdots + x_{r(C)} &\leq k \qquad \forall C \in \mathcal{C}_{red} \\ x_i &\in \{0,1\} \quad \forall i \in \{1, \ldots, |S|\} \end{aligned}$$

Note that the coefficient matrix of this ILP has the C1P, that is, every row of the matrix contains only either 0's and 1's or 0's and −1's, and in every row the non-zero entries appear consecutively. Now consider the relaxation of the ILP, that is, replace the constraints $x_i \in \{0, 1\}$ $\forall i \in \{1, \ldots, |S|\}$ by $-x_i \leq 0$ $\forall i \in \{1, \ldots, |S|\}$ and $x_i \leq 1$ $\forall i \in \{1, \ldots, |S|\}$.

As we will see, the resulting system of constraints has the property that its coefficient matrix is *totally unimodular*, which means that every square submatrix has determinant 0, 1, or -1. The following theorem of Hoffman and Kruskal [18] shows that if the relaxed linear program has a feasible solution, then it has also an integral feasible solution. Moreover, such an integral feasible solution can easily be found in polynomial time, because every corner of the polyhedron given by the inequality system is integral.

Theorem 2 ([18]) *Let A be an $m \times n$ integral matrix. Then the polyhedron defined by $Ax \leq b, x \geq 0$ is integral for every integral vector $b \in \mathbb{N}^m$ if and only if A is totally unimodular.*

In order to see that the coefficient matrix is always totally unimodular, consider the following characterization of totally unimodular matrices by Ghouila-Houri [17].

Theorem 3 ([17]) *An $m \times n$ matrix A with entries $0, 1, -1$ is totally unimodular if and only if each collection of columns from A can be split into two partitions such that in each row the sum of the entries of the first partition and the sum of the entries of the second partition differ by at most 1.*

The coefficient matrix of our system of constraints clearly fulfills the conditions of Theorem 3: Take an arbitrary collection of columns from the coefficient matrix and order them according to the C1P. Splitting the columns by putting every second column, starting with the first, into one partition and every second column, starting with the second, into the other partition, leads to a splitting as required in Theorem 3. Solving MDH with C1P in this way needs $O(|S|^5 \log(k))$ arithmetic operations on numbers that can be encoded with $O(|S|^2 \log k)$ bits [20].

So far, only the fact that the coefficient matrix is totally unimodular was used. However, it is known that an ILP whose coefficient matrix has the C1P can be solved even faster by transforming it into an edge-weighted graph and solving a shortest path problem on this graph. To this end, replace the n variables $x_1, \ldots, x_n$ by $n+1$ variables $y_0, \ldots, y_n$ such that $x_i = y_i - y_{i-1}$ for all $i \in \{1, \ldots, n\}$, which yields the following inequation system.

$$\begin{aligned} y_{l(C)-1} - y_{r(C)} &\leq -1 && \forall C \in \mathcal{C}_{blue} \\ -y_{l(C)-1} + y_{r(C)} &\leq k && \forall C \in \mathcal{C}_{red} \\ -y_i + y_{i-1} &\leq 0 && \forall i \in \{1, \ldots, |S|\} \\ y_i - y_{i-1} &\leq 1 && \forall i \in \{1, \ldots, |S|\} \end{aligned}$$

In this coefficient matrix every row contains exactly one 1 and one -1 and, hence, can be interpreted as a directed edge in a graph G whose vertices correspond to the variables $y_0, \ldots, y_n$. More precisely, let $G = (V, E)$ be the directed edge-weighted graph with

$$V = \{v_i \mid \text{ the ILP contains a variable } y_i\},$$
$$E = \{(v_i, v_j) \mid \text{ the ILP contains an inequation whose left side is } -y_i + y_j\},$$

where every edge e has a weight that is equal to the right side of the inequation corresponding to e in the ILP.

Now consider the following statement known as *Farkas' Lemma* (see Schrijver [31]).

Lemma 4 *Let $A \in \mathbb{R}^{m \times n}$ be a matrix and $b \in \mathbb{R}^m$ be a vector. Then the inequation system $Ay \leq b$ has a solution $y \in \mathbb{R}^n$ if and only if the inequation system $z^{\mathrm{T}}A = (0^n)^{\mathrm{T}}, z^{\mathrm{T}}b < 0, z \geq 0^m$ has* no *solution $z \in \mathbb{R}^m$.*

Interpreting A as the incidence matrix of the graph G defined above, Farkas' Lemma says that the given MDH instance is a *yes*-instance iff G contains no directed cycle whose edge weight sum is negative. To see this, observe that every positive component of the solution vector z corresponds to an edge of such a cycle: the constraint $z^{\mathrm{T}}A = (0^n)^{\mathrm{T}}$ enforces that for every vertex in G the same number of ingoing and outgoing edges have to be selected. By using the Bellmann-Ford-Moore-Algorithm [8], it can be decided in $O(|V| \cdot |E|)$ time if G contains such a negative cycle, and, hence, MDH with C1P can be decided in $O(|S| \cdot (|\mathcal{C}_{blue}| + |\mathcal{C}_{red}| + 2 \cdot |S|)) = O(|S|^3)$ time.

If G contains no cycle with negative edge weight sum and a solution for the ILP shall be constructed (that is, the values of the y_i shall be computed), then just set y_0 to 0 and y_i, $i \in \{1, \ldots, n\}$, to the length of the shortest path in G from v_0 to v_i. Because G contains no negative cycle, these shortest paths are all well-defined. It is easy to see that this solution satisfies all inequalities of the ILP. The shortest paths can be computed by the Bellmann-Ford-Moore-Algorithm in $O(|S|^3)$ time.

Altogether, we summarize our observations in the following theorem.

Theorem 5 MDH *can be solved in* $O(|S| \cdot (|\mathcal{C}_{blue}| + |\mathcal{C}_{red}| + |S|)) = O(|S|^3)$ *time if $\mathcal{C}_{blue} \cup \mathcal{C}_{red}$ has the C1P.*

3.2 Greedy Algorithm for Minimum Degree Hypergraph

As we have seen in the previous section, MDH with C1P can be solved in polynomial time with an ILP approach. By way of contrast, here we describe a simple greedy algorithm for MDH with C1P that has an absolute approx-

Originally published in *Journal of Discrete Algorithms*, 6(3):393–407. Elsevier B. V, 2008.

imation guarantee of additive term "+1." To this end, we consider the optimization version of MDH: Given S, $\mathcal{C}_{blue}$, and $\mathcal{C}_{red}$, find a subset $S' \subseteq S$ with $S' \cap C \neq \emptyset$ for all $C \in \mathcal{C}_{blue}$ which minimizes $\max_{C' \in \mathcal{C}_{red}}\{|C' \cap S'|\}$.

The idea of the greedy algorithm is to search in each step for the set $C \in \mathcal{C}_{blue}$ with the leftmost right index $r(C)$ such that no element of C is contained in the current solution set, and to add the rightmost column of C to the solution:

01 $S' \leftarrow \emptyset$, $\mathcal{C}'_{blue} \leftarrow \mathcal{C}_{blue}$
02 **while** $\mathcal{C}'_{blue} \neq \emptyset$
03 $C \leftarrow$ set from $\mathcal{C}'_{blue}$ with minimum right index
04 $S' \leftarrow S' \cup \{s_{r(C)}\}$
05 $\mathcal{C}'_{blue} \leftarrow \mathcal{C}'_{blue} \setminus \{C \in \mathcal{C}'_{blue} : C \cap S' \neq \emptyset\}$
06 **return** S'

Theorem 6 *For* MDH *with C1P, the greedy algorithm approximates an optimal solution within an additive term of one in $O(|S| \cdot |\mathcal{C}_{blue}|)$ time, provided that the elements in S are sorted such that all subsets in $\mathcal{C}_{blue}$ have the C1P.*

PROOF. Obviously, the output S' of the greedy algorithm has the minimum overlap property. It is also clear that all steps of the algorithm can be done in $O(|S| \cdot |\mathcal{C}_{blue}|)$ time altogether. It remains to determine $\max_{C' \in \mathcal{C}_{red}}\{|C' \cap S'|\}$.

Let $C_{\max}$ denote one subset in $\mathcal{C}_{red}$ with $|C_{\max} \cap S'| = \max_{C' \in \mathcal{C}_{red}}\{|C' \cap S'|\}$. Due to the C1P, all sets C chosen in step *03* are pairwise disjoint, and, hence, the set $C_{\max}$ contains at least $|C_{\max} \cap S'| - 1$ pairwise disjoint sets from $\mathcal{C}_{blue}$ as subsets, implying that *any* solution for this instance has to contain at least $|C_{\max} \cap S'| - 1$ elements from $C_{\max}$ in order to satisfy the minimum overlap property for these pairwise disjoint $\mathcal{C}_{blue}$-sets. Therefore, $|C_{\max} \cap S'_{\text{opt}}| \geq |C_{\max} \cap S'| - 1$ for any optimal solution S'_{opt}. □

3.3 Dynamic Programming for Red-Blue Set Cover

In the case of RBSC with C1P, we do not know an ILP formulation whose coefficient matrix is totally unimodular. We now present a polynomial-time dynamic programming algorithm that solves the optimization version of Red-Blue Set Cover with C1P: Given S, $\mathcal{C}_{blue}$, and $\mathcal{C}_{red}$, find a subset $S' \subseteq S$ with $S' \cap C \neq \emptyset$ for all $C \in \mathcal{C}_{blue}$ which minimizes $|\{C \in \mathcal{C}_{red} \mid S' \cap C \neq \emptyset\}|$.

We assume that the sets in $\mathcal{C}_{blue}$ are ordered according to their left indices and denote them with $B_1, \ldots, B_{|\mathcal{C}_{blue}|}$; the sets of $\mathcal{C}_{red}$ are ordered analogously and denoted with $R_1, \ldots, R_{|\mathcal{C}_{red}|}$. If a set in $\mathcal{C}_{blue}$ is a superset of another set in $\mathcal{C}_{blue}$, it can be removed. Therefore, for any two sets $B_i, B_j \in \mathcal{C}_{blue}$ it holds that

$$l(B_i) < l(B_j) \Leftrightarrow r(B_i) < r(B_j).$$

Given a subset $S' \subseteq S$, we denote with $w(S')$ the number of sets from $\mathcal{C}_{red}$ that are covered by S'.

The idea of the dynamic programming algorithm is to compute so-called *optimal partial solutions* $S_{\text{opt}}(i_1, i_2, j)$. Each optimal partial solution $S_{\text{opt}}(i_1, i_2, j)$ has the following properties:

(1) $S_{\text{opt}}(i_1, i_2, j) \subseteq \{s_1, \ldots, s_{i_1}\}$,
(2) $S_{\text{opt}}(i_1, i_2, j)$ covers all sets $B_1, \ldots, B_j$,
(3) if $i_2 > 0$, then $S_{\text{opt}}(i_1, i_2, j)$ contains at least one element from $\{s_{i_2}, \ldots, s_n\}$ (where $n := |S|$), and
(4) the cost $w(S_{\text{opt}}(i_1, i_2, j))$ is minimum under all subsets of S that have the first three properties.

A subset of S that has the first three properties is called a *feasible partial solution.*

The algorithm uses a three-dimensional table $S_{\text{opt}}(i_1, i_2, j)$ with $1 \leq i_1 \leq n$, $0 \leq i_2 \leq n$, and $1 \leq j \leq |\mathcal{C}_{blue}|$ for storing optimal partial solutions, and a table $W_{\text{opt}}(i_1, i_2, j)$ of the same size where the cost of every optimal partial solution is stored. Then, the entry $S_{\text{opt}}(n, 0, |\mathcal{C}_{blue}|)$ contains an optimal solution for the RBSC instance.

The two tables are filled with three nested loops, iterating over i_1, i_2, and j. To compute table entries $S_{\text{opt}}(i_1, i_2, j), W_{\text{opt}}(i_1, i_2, j)$ with $i_1 = 1$ is simple. All other entries are computed as follows: If $l(B_j) > i_1$ or $i_2 > i_1$, then there is no partial solution $S_{\text{opt}}(i_1, i_2, j)$, and $W_{\text{opt}}(i_1, i_2, j)$ is set to ∞. Otherwise, we consider two cases: the optimal partial solution contains s_{i_1} or not. (Note that if $i_2 = i_1$, then all feasible partial solutions have to contain s_{i_1}.) In the first case, the optimal partial solution $S_{\text{opt}}(i_1, i_2, j)$ can only contain elements from $\{s_1, \ldots, s_{i_1-1}\}$, and, hence, $S_{\text{opt}}(i_1, i_2, j) = S_{\text{opt}}(i_1-1, i_2, j)$. In the second case, the optimal partial solution $S_{\text{opt}}(i_1, i_2, j)$ is computed as follows: By choosing s_{i_1}, property (3) is clearly obtained, because we can assume that $i_2 \leq i_1$. Moreover, all sets in $\mathcal{C}_{blue}$ that contain s_{i_1} are covered by s_{i_1}. Therefore, in order to obtain property (2), it remains to cover those sets $B_p \in \{B_1, \ldots, B_j\}$ that have $r(B_p) < i_1$. Hence, adding s_{i_1} to an optimal partial solution $S_{\text{opt}}(i_1 - 1, i_2', j')$, where i_2' is chosen from $\{0, \ldots, i_2\}$ and j' is the maximum possible index such that $r(B_{j'}) < i_1$, yields an optimal partial solution $S_{\text{opt}}(i_1, i_2, j)$. The value for i_2' has to be chosen such that $W(i_1, i_2, j) = W(i_1 - 1, i_2', j') + |\mathcal{C}_{red}(i_1)| - |X|$ is minimum, where $\mathcal{C}_{red}(i_1)$ denotes the sets from $\mathcal{C}_{red}$ that are covered by s_{i_1} and X denotes the sets from $\mathcal{C}_{red}$ that are covered by both s_{i_1} and $S_{\text{opt}}(i_1 - 1, i_2', j')$.

Originally published in *Journal of Discrete Algorithms*, 6(3):393–407. Elsevier B. V, 2008.

Before showing the details of our algorithm and proving its correctness, we introduce some more notations:

$$\mathcal{C}_{red}(i) := \{C \in \mathcal{C}_{red} \mid s_i \in C\}, 1 \leq i \leq n, \text{ and}$$

$$\mathcal{C}^{\leftarrow}_{red}(i) := \{C \in \mathcal{C}_{red} \mid s_i \in C \wedge s_{i-1} \in C\}, 1 < i \leq n.$$

With $R^{\leftarrow}(i,k)$ we denote the kth set from $\mathcal{C}^{\leftarrow}_{red}(i)$, where we assume that the sets $C \in \mathcal{C}^{\leftarrow}_{red}(i)$ are ordered according to $l(C)$. The following pseudo code shows how an optimal partial solution $S_{\text{opt}}(i_1,i_2,j)$ together with its cost $W_{\text{opt}}(i_1,i_2,j)$ is computed for $i_1 > 1$.

01 **if** $(l(B_j) > i_1) \vee (i_2 > i_1)$ **then**
02 $\quad W_{\text{opt}}(i_1,i_2,j) \leftarrow \infty$, $S_{\text{opt}}(i_1,i_2,j) \leftarrow \emptyset$, **break**
03 $W_{\text{opt}}(i_1,i_2,j) \leftarrow W_{\text{opt}}(i_1-1,i_2,j)$ // Lines 3–4: partial solution not containing s_{i_1}
04 $S_{\text{opt}}(i_1,i_2,j) \leftarrow S_{\text{opt}}(i_1-1,i_2,j)$
05 $j' \leftarrow \max\{p \in \{1,\ldots,j\} \mid r(B_p) < i_1\}$ // Lines 5–14: partial sol. containing s_{i_1}
06 $x \leftarrow W_{\text{opt}}(i_1-1,0,j') + |\mathcal{C}_{red}(i_1)|$
07 **if** $x < W_{\text{opt}}(i_1,i_2,j)$ **then**
08 $\quad W_{\text{opt}}(i_1,i_2,j) \leftarrow x$
09 $\quad S_{\text{opt}}(i_1,i_2,j) \leftarrow S_{\text{opt}}(i_1-1,0,j') \cup \{s_{i_1}\}$
10 **for** $k = 1$ **to** $|\mathcal{C}^{\leftarrow}_{red}(i_1)|$ **do**
11 $\quad x \leftarrow W_{\text{opt}}(i_1-1, l(R^{\leftarrow}(i_1,k)), j') + |\mathcal{C}_{red}(i_1)| - k$
12 $\quad$ **if** $x < W_{\text{opt}}(i_1,i_2,j)$ **then**
13 $\quad\quad W_{\text{opt}}(i_1,i_2,j) \leftarrow x$
14 $\quad\quad S_{\text{opt}}(i_1,i_2,j) \leftarrow S_{\text{opt}}(i_1-1, l(R^{\leftarrow}(i_1,k)), j') \cup \{s_{i_1}\}$

Theorem 7 RBSC *can be solved in* $O(|\mathcal{C}_{blue}| \cdot |\mathcal{C}_{red}| \cdot |S|^2)$ *time if* $\mathcal{C}_{blue} \cup \mathcal{C}_{red}$ *has the C1P.*

PROOF. We show the correctness of the pseudo code shown above. In lines 3–4 the algorithm searches for an optimal partial solution $S_{\text{opt}}(i_1,i_2,j)$ that does not contain s_{i_1}. Lines 5–14 handle the case that the optimal partial solution $S_{\text{opt}}(i_1,i_2,j)$ contains s_{i_1}. Clearly the procedure outputs a feasible partial solution, and it is easy to verify that the value of $W_{\text{opt}}(i_1,i_2,j)$ computed by the procedure upper-bounds the cost of the partial solution $S_{\text{opt}}(i_1,i_2,j)$ computed by the procedure. It remains to show that the value of $W_{\text{opt}}(i_1,i_2,j)$ computed by the procedure equals the actual cost of an optimal partial solution in the case that the optimal partial solution contains s_{i_1}.

To this end, let $S_{\text{opt}}(i_1,i_2,j)$ be an optimal partial solution where $s_{i_1} \in S_{\text{opt}}(i_1,i_2,j)$. Moreover, let $S' := S_{\text{opt}}(i_1,i_2,j) \setminus \{s_{i_1}\}$, let $j' := \max\{p \in \{1,\ldots,j\} \mid r(B_p) < i_1\}$, and let $i' := \max\{q \in \{1,\ldots,n\} \mid s_q \in S'\}$. We distinguish two cases.

Case 1: For all $C \in \mathcal{C}_{red}(i_1)$ it holds that $s_{i'} \notin C$. Then $W_{\text{opt}}(i_1, i_2, j) = w(S') + |\mathcal{C}_{red}(i_1)|$. The set S' must have the following properties: S' consists of elements from $\{s_1, \ldots, s_{i_1-1}\}$, and S' covers all sets $B_1, \ldots, B_{j'}$. Under all subsets of S having these two properties, the set $S_{\text{opt}}(i_1 - 1, 0, j')$ is, by definition, the one with minimum cost, and, hence, choosing $S_{\text{opt}}(i_1, i_2, j) = S_{\text{opt}}(i_1 - 1, 0, j') \cup \{s_{i_1}\}$ is optimal. In this case, the procedure finds the correct value of $W_{\text{opt}}(i_1, i_2, j)$ in lines 6–9.

Case 2: There exists a $k \in \{1, \ldots, |\mathcal{C}^{\leftarrow}_{red}(i_1)|\}$ such that $R^{\leftarrow}(i_1, k) \cap S' \neq \emptyset$. We assume that k is maximum under this property. Due to the order of the sets in $\mathcal{C}_{red}$, we have $R^{\leftarrow}(i_1, k') \cap S' \neq \emptyset$ for all $k' < k$, and, hence, $W_{\text{opt}}(i_1, i_2, j) = w(S') + |\mathcal{C}_{red}(i_1)| - k$. The set S' must have the following properties: S' consists of elements from $\{s_1, \ldots, s_{i_1-1}\}$, and S' covers all sets $B_1, \ldots, B_{j'}$. Moreover, the maximum index i' of an element in S' has to satisfy $i' \geq l(R^{\leftarrow}(i_1, k))$, because otherwise $R^{\leftarrow}(i_1, k)$ would not be covered by S'. Under all subsets of S having these three properties, the set $S_{\text{opt}}(i_1 - 1, l(R^{\leftarrow}(i_1, k)), j')$ is the one with minimum cost, and, hence, choosing $S_{\text{opt}}(i_1, i_2, j) := S_{\text{opt}}(i_1 - 1, l(R^{\leftarrow}(i_1, k)), j') \cup \{s_{i_1}\}$ is optimal. In this case, the procedure finds the correct value of $W_{\text{opt}}(i_1, i_2, j)$ in lines 10–14.

It remains to show the running time. The table size is $O(|S|^2 \cdot |\mathcal{C}_{blue}|)$, and to compute an entry of the table, at most $O(|\mathcal{C}_{red}|)$ other entries have to be considered in lines 10–14. Line 5 can be executed in constant time if, in a preprocessing step (which can be implemented similar to bucket sort and needs $O(|\mathcal{C}_{blue}| + |S|)$ time), for every possible value of i_1 the corresponding value of j' is computed and stored in an extra table. This yields the claimed running time. □

4 Minimum Degree Hypergraph and Red-Blue Set Cover with Partial C1P

Whereas the C1P case always leads to polynomial-time solvability, in case of only partially holding C1Ps we typically face NP-hardness as shown in this section.

4.1 Minimum Degree Hypergraph with Partial C1P

In this section we prove that MDH remains NP-complete even under the requirement that either $\mathcal{C}_{blue}$ or $\mathcal{C}_{red}$ is to have the C1P. To this end, we give reductions from the following restricted variant of the Satisfiability problem:

Originally published in *Journal of Discrete Algorithms*, 6(3):393–407. Elsevier B. V, 2008.

Restricted 3-Sat (R3-Sat)
Input: An n-variable, m-clause 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 for F.

It is well-known that R3-Sat is NP-complete (e.g., see [28, p. 183]).[6] Without loss of generality, we assume that no variable appears in F solely positively or negatively, and F contains no singleton clause.

Our reductions show the NP-completeness of Minimum Degree Hypergraph variants that have, apart from the C1P for $\mathcal{C}_{blue}$ or $\mathcal{C}_{red}$, several further restrictions. In order to emphasize the correlation between the hardness of the problem and the value of k and the subset sizes in $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$, we summarize some of the results in the following statement, which is a corollary of Observations 2 and 3, Corollary 1, and Theorems 9, 10, 11, and 12.

Corollary 8 MDH *is* NP*-complete even if all of the following restrictions apply:*

(1) One of the collections $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$ has the consecutive ones property,
(2) $k = 1$, and
(3) $\forall C \in \mathcal{C}_{blue} : |C| \leq 3$ and $\forall C \in \mathcal{C}_{red} : |C| \leq 2$.

However, replacing restriction (2) by $k = 0$, replacing restriction (3) by $\forall C \in \mathcal{C}_{blue} : |C| \leq 2$, or replacing restriction (3) by $\forall C \in \mathcal{C}_{red} : |C| \leq 1$ leads to polynomial-time solvability.

MDH *is* NP*-complete even if all of the following restrictions apply:*

(1) One of the collections $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$ has the consecutive ones property,
(2) $k = 2$, and
(3) $\forall C \in \mathcal{C}_{blue} : |C| \leq 2$ and $\forall C \in \mathcal{C}_{red} : |C| \leq 3$.

However, replacing restriction (2) by $k \leq 1$, replacing restriction (3) by $\forall C \in \mathcal{C}_{blue} : |C| \leq 1$, or replacing restriction (3) by $\forall C \in \mathcal{C}_{red} : |C| \leq 2$ leads to polynomial-time solvability.

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

The following two theorems (Theorems 9 and 10) show that the requirement of $\mathcal{C}_{blue}$ obeying the C1P does not make MDH tractable. The theorems com-

[6] 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 solvable in polynomial time [28].

plement each other in the sense that they impose different restrictions on the cardinalities of the sets $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$; Theorem 9 needs size-3 sets in $\mathcal{C}_{blue}$ and size-2 sets in $\mathcal{C}_{red}$ (the reduction encodes clauses of a given R3-SAT instance in $\mathcal{C}_{blue}$) while the converse holds true for Theorem 10 (the reduction encodes clauses in $\mathcal{C}_{red}$).

Theorem 9 MDH *is* NP-*complete even if all of the following restrictions apply:*

(1) The collection $\mathcal{C}_{blue}$ has the consecutive ones property,
(2) $k = 1$,
(3) $\forall C \in \mathcal{C}_{blue} : |C| \leq 3$ and $\forall C \in \mathcal{C}_{red} : |C| \leq 2$, and
(4) $\forall s \in S : |\{C \in \mathcal{C}_{blue} \mid s \in C\}| = 1$ and $|\{C \in \mathcal{C}_{red} \mid s \in C\}| \leq 2$.

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 $(S, \mathcal{C}_{blue}, \mathcal{C}_{red}, k)$ of MDH:

- The set S consists of the 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 $\mathcal{C}_{blue}$ 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 $\mathcal{C}_{blue}$ if it contains three literals and $\{s_i^1, s_i^2\}$ if it contains two literals.
- For all variables x 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, then $\mathcal{C}_{red}$ contains the set $\{s_j^i, s_q^p\}$.
- The maximum overlap k is set to one.

The construction is illustrated in Figure 2. It is easy to see that, by the definition of R3-SAT, the constructed instance satisfies the restrictions claimed in the theorem; note that $\mathcal{C}_{blue}$ has the consecutive ones property due to Observation 1. It remains to be shown that the constructed instance of MDH has a solution iff F has a satisfying truth assignment T.

"$\Rightarrow$" Assume that the constructed instance of MDH 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 with $k = 1$—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' corresponds to a literal in this clause because S' has the minimum overlap property. On the one hand, if this

Originally published in *Journal of Discrete Algorithms*, 6(3):393–407. Elsevier B. V, 2008.

$$\begin{aligned} F &= (x_1 \vee x_2 \vee \neg x_3) \\ &\wedge\ (x_3 \vee x_4) \\ &\wedge\ (\neg x_1 \vee \neg x_2) \\ &\wedge\ (\neg x_1 \vee \neg x_3 \vee \neg x_4) \end{aligned}$$

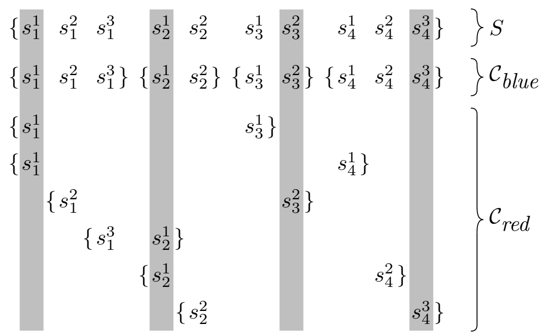

Fig. 2. Example of encoding an instance of R3-Sat into an instance of MDH (proof of Theorem 9). Each clause of the Boolean formula F is represented by a set in $\mathcal{C}_{blue}$. The sets in $\mathcal{C}_{red}$ and the maximum overlap $k = 1$ ensure that no two elements from S that correspond to conflicting truth assignments of the same variable can be chosen into a solution. Observe how $S' = \{s_1^1, s_2^1, s_3^2, s_4^3\}$ (grey columns) constitutes a valid solution to the MDH instance; accordingly, a truth assignment T which makes all the corresponding literals evaluate to *true* satisfies F.

element corresponds to a positive literal x_i, then $T(x_i) = true$, satisfying the clause. On the other hand, if the element corresponds to a negative literal $\neg x_i$, then $T(x_i) = false$, 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 $\mathcal{C}_{blue}$ represents exactly one clause of F. Also, S' has the maximum overlap property with $k = 1$ because T is well-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 MDH instance. $\square$

Theorem 10 MDH *is* NP-*complete even if all of the following restrictions apply:*

(1) The collection $\mathcal{C}_{blue}$ has the consecutive ones property,
(2) $k = 2$,
(3) $\forall C \in \mathcal{C}_{blue} : |C| \leq 2$ and $\forall C \in \mathcal{C}_{red} : |C| \leq 3$, and
(4) $\forall s \in S : |\{C \in \mathcal{C}_{blue} \mid s \in C\}| = 1$ and $|\{C \in \mathcal{C}_{red} \mid s \in C\}| \leq 2$.

PROOF. We prove the theorem by a reduction from R3-Sat. The reduction is similar to the one used in the proof of Theorem 9, but this time we use the sets of $\mathcal{C}_{red}$ instead of those of $\mathcal{C}_{blue}$ to model the clauses of F, and we use the sets of $\mathcal{C}_{blue}$ to enforce the consistency between literals representing the same variable. Moreover, in contrast to the reduction used in the proof of Theorem 9, here each element chosen into the solution set—if a solution exists—stands for a literal that is set to *false* by a satisfying truth assignment for F. Hence, not

$$\begin{aligned} F &= (x_1 \vee x_2 \vee \neg x_3) \\ &\wedge\ (x_3 \vee x_4) \\ &\wedge\ (\neg x_1 \vee \neg x_2) \\ &\wedge\ (\neg x_1 \vee \neg x_3 \vee \neg x_4) \end{aligned}$$

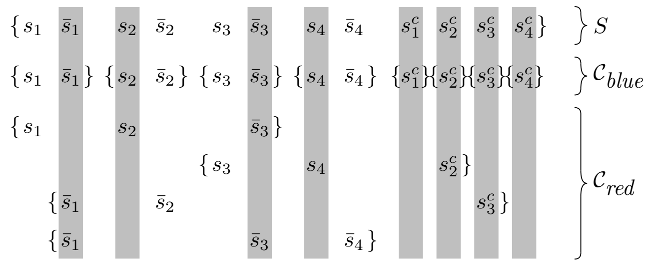

Fig. 3. Example of encoding an instance of R3-SAT into an instance of MDH (proof of Theorem 10). Each clause of the Boolean formula F is represented by a set in $\mathcal{C}_{red}$. The sets in $\mathcal{C}_{blue}$ ensure that for each variable one element of the two elements corresponding to its positive and negative literal is chosen into a solution; the maximum overlap $k = 2$ ensures that for each clause at most two elements corresponding to its literals are chosen. Observe how $S' = \{\bar{s}_1, s_2, \bar{s}_3, s_4, s_1^c, s_2^c, s_3^c, s_4^c\}$ (grey columns) constitutes a valid solution to the MDH instance; accordingly, a truth assignment T with $T(x_i) = \mathit{true}$ iff $s_i \notin S'$ satisfies F.

more than two elements per red set may be chosen into the solution set if the corresponding truth assignment for F shall be satisfying; this is expressed by setting k to two. In order to prevent both literals of a size-2 clause from being set to *false*, we add to each set in $\mathcal{C}_{red}$ corresponding to a size-2 clause a dummy element which has to be part of every solution.

The instance $(S, \mathcal{C}_{blue}, \mathcal{C}_{red}, k)$ of MDH is constructed as follows:

- We set $S := \{s_1, \bar{s}_1, \ldots, s_n, \bar{s}_n\} \cup \{s_1^c, \ldots, s_m^c\}$. Herein, n denotes the number of variables in F and m denotes the number of clauses in F. For a variable x_i in F, s_i represents the literal x_i and $\bar{s}_i$ represents the literal $\neg x_i$. We use the elements s_i^c to ensure that each set in $\mathcal{C}_{red}$ has size three.
- $\mathcal{C}_{blue} := \left(\bigcup_{1 \le i \le n} \{\{s_i, \bar{s}_i\}\}\right) \cup \{\{s_1^c\}, \ldots, \{s_m^c\}\}$.
- For each clause c in F, $\mathcal{C}_{red}$ contains a set C of those elements from S that represent the literals of c: If the j-th clause in F contains only two literals, then s_j^c is added to its representing set in $\mathcal{C}_{red}$ as the third element.
- The maximum overlap k is set to two.

See Figure 3 for an illustration of the construction. Clearly, this MDH instance satisfies all restrictions as claimed by the theorem. The correspondence between the solutions of the constructed instance and the satisfying truth assignments for F follows from the following two observations.

First, if the constructed MDH instance is solvable, then it has always a solution set S' such that, for each variable x_i, exactly one of s_i and $\bar{s}_i$ is in S'. This can easily be seen because if a solution set S' contains both of s_i and $\bar{s}_i$ for a variable x_i, then S' without s_i (or S' without $\bar{s}_i$) is also a solution for the MDH instance. This observation guarantees that we can always construct a well-defined truth assignment for F from S' and vice versa as follows: $T(x_i) = \mathit{true} \Leftrightarrow s_i \notin S'$.

Second, F is satisfiable with a truth assignment T if and only if every clause of size three has at most two literals that are evaluated to *false* by T and every clause of size two has at most one literal that is evaluated to *false*. By the correspondence between T and S', it is then easy to observe that T satisfies F iff S' fulfills the maximum overlap property with $k = 2$, that is, S' meets, for each clause c, the set in $\mathcal{C}_{red}$ corresponding to c at most twice. □

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

Note that by the reduction from Vertex Cover in Section 2, MDH is NP-complete already if $\mathcal{C}_{red}$ contains just a single set and, hence, has the C1P. However, this requires a non-fixed maximum overlap k and unrestricted cardinality of the set contained in $\mathcal{C}_{red}$. Therefore, if we want to show the NP-completeness of MDH with the additional restriction that the maximum overlap k is fixed and the sets in $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$ have small cardinality, another reduction is needed. Analogously to Theorems 9 and 10, the following two theorems impose different restrictions on the cardinalities of the sets in $\mathcal{C}_{blue}$ and $\mathcal{C}_{red}$.

Theorem 11 MDH *is* NP-*complete even if all of the following restrictions apply:*

(1) The collection $\mathcal{C}_{red}$ has the consecutive ones property,
(2) $k = 1$,
(3) $\forall C \in \mathcal{C}_{blue} : |C| \leq 3$ and $\forall C \in \mathcal{C}_{red} : |C| \leq 2$, and
(4) $\forall s \in S : |\{C \in \mathcal{C}_{blue} \mid s \in C\}| \leq 2$ and $|\{C \in \mathcal{C}_{red} \mid s \in C\}| = 1$.

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 $(S, \mathcal{C}_{blue}, \mathcal{C}_{red}, k)$ of MDH:

- The set S is equal to $\{s_1, \bar{s}_1, \ldots, s_n, \bar{s}_n\}$, that is, for each variable x_i in F, S contains an element s_i representing the literal x_i and an element $\bar{s}_i$ representing the literal $\neg x_i$.
- For each clause in F, $\mathcal{C}_{blue}$ contains a set of those elements from S that represent the literals of that clause.
- $\mathcal{C}_{red} = \bigcup_{1 \leq i \leq n} \{\{s_i, \bar{s}_i\}\}$.
- The maximum overlap k is set to one.

Observe that this MDH instance satisfies all restrictions claimed in the theorem. The reduction is illustrated by an example in Figure 4. The correctness of the reduction can be proven in a similar way as in the proof of Theorem 9. □

$$\begin{aligned} F &= (x_1 \vee x_2 \vee \neg x_3) \\ &\wedge\ (x_3 \vee x_4) \\ &\wedge\ (\neg x_1 \vee \neg x_2) \\ &\wedge\ (\neg x_1 \vee \neg x_3 \vee \neg x_4) \end{aligned}$$

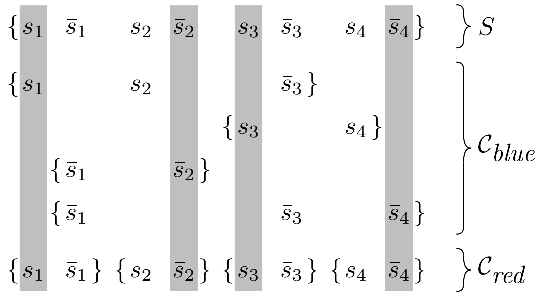

Fig. 4. Example of encoding an instance of R3-SAT into an instance of MDH (proof of Theorem 11). Each clause of the Boolean formula F is encoded into one set of $\mathcal{C}_{blue}$. Observe how $S' = \{s_1, \bar{s}_2, s_3, \bar{s}_4\}$ (grey columns) constitutes a valid solution to the MDH instance; accordingly, a truth assignment T with $T(x_i) = true$ iff $s_i \in S'$ satisfies F.

Theorem 12 MDH *is* NP*-complete even if all of the following restrictions apply:*

(1) The collection $\mathcal{C}_{red}$ has the consecutive ones property,
(2) $k = 2$,
(3) $\forall C \in \mathcal{C}_{blue} : |C| \leq 2$ and $\forall C \in \mathcal{C}_{red} : |C| \leq 3$, and
(4) $\forall s \in S : |\{C \in \mathcal{C}_{blue} \mid s \in C\}| \leq 2$ and $|\{C \in \mathcal{C}_{red} \mid s \in C\}| = 1$.

PROOF. The reduction used in this proof is a combination of the reductions used in the proofs of Theorems 9 and 10: We encode clauses and variables in a similar way as in the proof of Theorem 9. But here clauses are encoded in $\mathcal{C}_{red}$ and variables in $\mathcal{C}_{blue}$. As in the proof of Theorem 10, each element chosen into the solution set—if one exists—stands for a literal that is set to *false* by a satisfying truth assignment for F.

- We set $S := \{s_1^1, s_1^2, s_1^3, \ldots, s_m^1, s_m^2, s_m^3\} \cup \{s_1^c, \ldots, s_m^c\}$. The element s_j^i represents 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. The elements s_i^c are used to ensure that each set in $\mathcal{C}_{red}$ has size three.
- For all variables 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 $\{s_j^i, s_q^p\}$. Moreover, we add $\{s_i^c\}$ with $1 \leq i \leq m$ to $\mathcal{C}_{blue}$.
- For each clause in F, $\mathcal{C}_{red}$ contains a set of those elements from S that represent the literals of that clause. If the j-th clause in F contains only two literals, then s_j^c is added to the corresponding set in $\mathcal{C}_{red}$ as the third element.
- The maximum overlap k is set to two.

An example of the reduction is shown in Figure 5. The correctness of the reduction can be proven in a similar way as in the proof of Theorem 10. $\square$

Originally published in *Journal of Discrete Algorithms*, 6(3):393–407. Elsevier B. V, 2008.

$$
\begin{aligned}
F &= (x_1 \vee x_2 \vee \neg x_3) \\
&\wedge\ (x_3 \vee x_4) \\
&\wedge\ (\neg x_1 \vee \neg x_2) \\
&\wedge\ (\neg x_1 \vee \neg x_3 \vee \neg x_4)
\end{aligned}
$$

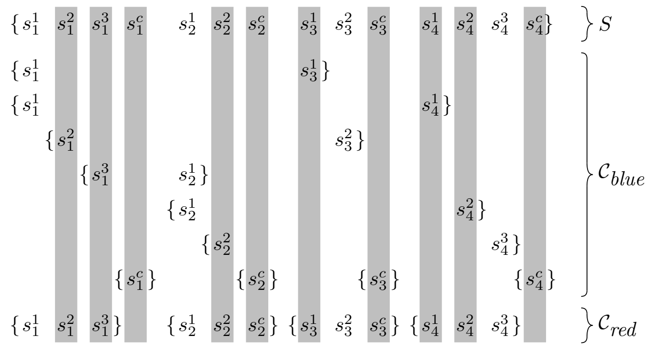

Fig. 5. Example of encoding an instance of R3-Sat into an instance of MDH (proof of Theorem 12). Each clause of the Boolean formula F is represented by a set in $\mathcal{C}_{red}$. Observe how $S' = \{s_1^2, s_1^3, s_1^c, s_2^2, s_2^c, s_3^1, s_3^c, s_4^1, s_4^2, s_4^c\}$ (grey columns) constitutes a valid solution to the MDH instance; accordingly, a truth assignment T which makes all the literals *not* corresponding to one of the chosen elements evaluate to *true* satisfies F.

4.2 Red-Blue Set Cover with Partial C1P

The problem Red-Blue Set Cover has been introduced by Carr et al. [7]; here we use the problem definition given in Section 1. We will show the NP-completeness of RBSC when restricted to instances where the sets in $\mathcal{C}_{blue}$ or the sets in $\mathcal{C}_{red}$ have the C1P.

Theorem 13 RBSC *is NP-complete even if*

(1) $|C| \leq 2$ *for all* $C \in \mathcal{C}_{blue}$*,* $|C| = 1$ *for all* $C \in \mathcal{C}_{red}$ *(which trivially implies that* $\mathcal{C}_{red}$ *has the consecutive ones property), and for all* $s \in S$*,* $|\{C \in \mathcal{C}_{blue} \mid s \in C\}| \leq 3$ *and* $|\{C \in \mathcal{C}_{red} \mid s \in C\}| = 1$*, or*
(2) *the collection* $\mathcal{C}_{blue}$ *has the consecutive ones property,* $|C| \leq 2$ *for all* $C \in \mathcal{C}_{blue}$*,* $|C| \leq 3$ *for all* $C \in \mathcal{C}_{red}$*, and for all* $s \in S$*,* $|\{C \in \mathcal{C}_{blue} \mid s \in C\}| = 1$ *and* $|\{C \in \mathcal{C}_{red} \mid s \in C\}| = 1$*.*

PROOF. We show both cases of the theorem by reductions from Vertex Cover restricted to cubic graphs. Vertex Cover restricted to cubic graphs is NP-hard [16].

To prove Case (1), let $G = (V, E)$ be a cubic graph. For the reduction, set $S := V$, $\mathcal{C}_{blue} := E$, and $\mathcal{C}_{red} := \{\{v\} \mid v \in V\}$. Clearly, the constructed instance satisfies all restrictions of this case. The one-to-one correspondence between the solutions follows directly from the construction.

To show Case (2), let $G = (V, E)$ be a cubic graph with $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. Construct the following instance $(S, \mathcal{C}_{blue}, \mathcal{C}_{red}, k)$ of RBSC:

- $S := \{s_l^i, s_l^j \mid e_l = \{v_i, v_j\} \in E\}$, that is, S contains, for every edge e_l, two elements corresponding to e_l's endpoints.
- $\mathcal{C}_{blue} := \{\{s_l^i, s_l^j\} \mid e_l = \{v_i, v_j\} \in E\}$.
- For every vertex $v_i \in V$ we add to $\mathcal{C}_{red}$ a set C_i consisting of three v_i's "occurrences." More precisely, $s_l^i \in C_i$ for an edge e_l with v_i as one endpoint.

Since the sets in $\mathcal{C}_{blue}$ are pairwise disjoint, $\mathcal{C}_{blue}$ has the consecutive ones property. The other restrictions of this case are also clearly satisfied.

It is easy to see that G has a vertex cover with at most k vertices iff the constructed RBSC-instance has a solution with maximum containment k: Given a vertex cover V' of G, the RBSC-instance has a solution $S' := \cup_{v_i \in V'} C_i$; Conversely, given a solution S' of the RBSC-instance, the set $V' := \{v_i \mid C_i \cap S' \neq \emptyset, C_i \in \mathcal{C}_{red}\}$ is clearly a size-$\leq k$ vertex cover of G. $\square$

The restriction on the cardinality of $\mathcal{C}_{blue}$-sets in Case (1) of Theorem 13 is clearly tight: For cardinality-one $\mathcal{C}_{blue}$-sets we have only one choice, that is, taking the element into the solution.

Finally, we mention in passing that our reduction also implies that RBSC as restricted above can only be approximated up to a constant factor, that is, it is MaxSNP-hard [29]. This is due to the fact that the reductions in the proof of Theorem 13 are clearly approximation-preserving reductions. Thus, the claim follows from the fact that Vertex Cover restricted to cubic graphs still is MaxSNP-hard [29].

5 Outlook

There are many natural challenges for future work. For instance, it is desirable to find out more about the polynomial-time approximability [2,34] and the parameterized complexity [12,15,27] of the variants of Minimum Degree Hypergraph and Red-Blue Set Cover proven to be NP-complete. Moreover, the connections to orthogonal segment stabbing [21] in computational geometry should be further explored.

Acknowledgement: We thank two anonymous referees for their constructive comments helping to improve the presentation of our work. In particular, we are grateful to an anonymous referee for pointing us to the shortest path approach for solving MDH with C1P (Section 3.1).

References

[1] B. Aspvall, M. F. Plass, and R. E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8:121–123, 1979. 6

[2] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation — Combinatorial Optimization Problems and Their Approximability Properties.* Springer, 1999. 21

[3] 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

[4] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976. 5, 7

[5] A. Caprara, G. F. Italiano, G. Mohan, A. Panconesi, and A. Srinivasan. Wavelength rerouting in optical networks, or the Venetian routing problem. *Journal of Algorithms*, 45(2):93–125, 2002. 3

[6] A. Caprara, P. Toth, and M. Fischetti. Algorithms for the set covering problem. *Annals of Operations Research*, 98:353–371, 2000. 2

[7] R. D. Carr, S. Doddi, G. Konjevod, and M. V. Marathe. On the red-blue set cover problem. In *Proc. 11th SODA*, pages 345–353. ACM Press, 2000. 3, 20

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, 2001. 2, 9

[9] M. Dom, J. Guo, and R. Niedermeier. Approximability and parameterized complexity of consecutive ones submatrix problems. In *Proc. 4th TAMC*, volume 4484 of *LNCS*, pages 680–691. Springer, 2007. 4

[10] M. Dom, J. Guo, R. Niedermeier, and S. Wernicke. Minimum membership set covering and the consecutive ones property. In *Proc. 10th SWAT*, volume 4059 of *LNCS*, pages 339–350. Springer, 2006.

[11] M. Dom and R. Niedermeier. The search for consecutive ones submatrices: Faster and more general. In *Proc. 3rd ACiD*, volume 9 of *Texts in Algorithmics*, pages 43–54. College Publications, 2007. 4

[12] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999. 2, 21

[13] T. Feder, R. Motwani, and A. Zhu. k-connected spanning subgraphs of low degree. *Electronic Colloquium on Computational Complexity (ECCC)*, (TR06-041), 2006. 3

[14] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998. 2

[15] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006. 21

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979. 6, 20

[17] A. Ghouila-Houri. Caractérisation des matrices totalement unimodulaires. *Comptes Rendus Hebdomadaires des Séances de l' Academie des Sciences (Paris)*, (254):1192–1194, 1962. 8

[18] A. J. Hoffman and J. B. Kruskal. Integral boundary points of convex polyhedra. In H. W. Kuhn and A. W. Tucker, editors, *Linear Inequalities and Related Systems*, pages 223–246. Princeton University Press, 1956. 8

[19] W.-L. Hsu and R. M. McConnell. PC trees and circular-ones arrangements. *Theoretical Computer Science*, 296(1):99–116, 2003. 5, 7

[20] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proc. 16th STOC*, pages 302–311. ACM Press, 1984. 8

[21] M. J. Katz, J. S. B. Mitchell, and Y. Nir. Orthogonal segment stabbing. *Computational Geometry*, 30(2):197–205, 2005. 4, 21

[22] 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. 3, 4

[23] S. Mecke, A. Schöbel, and D. Wagner. Station location - complexity and approximation. In *Proc. 5th ATMOS*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2005. 4

[24] 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. 4

[25] J. Meidanis, O. Porto, and G. Telles. On the consecutive ones property. *Discrete Applied Mathematics*, 88:325–354, 1998. 4

[26] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988. 4

[27] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. 21

[28] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994. 14

[29] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991. 21

[30] N. Ruf and A. Schöbel. Set covering with almost consecutive ones property. *Discrete Optimization*, 1(2):215–228, 2004. 4

[31] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986. 7, 9

[32] J. Tan and L. Zhang. The consecutive ones submatrix problem for sparse matrices. *Algorithmica*, 48(3):287–299, 2007. 4

[33] A. C. Tucker. A structure theorem for the consecutive 1's property. *Journal of Combinatorial Theory (B)*, 12:153–162, 1972. 4

[34] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001. 21

[35] A. F. Veinott and H. M. Wagner. Optimal capacity scheduling. *Operations Research*, 10:518–532, 1962. 4