

Deckert, Dominic; Reichelt, Dirk; Holland-Moritz, Peter

**Article — Published Version**

## Introducing Covering Problems for the Electronic Wafer Test

Operations Research Forum

**Provided in Cooperation with:**

Springer Nature

*Suggested Citation:* Deckert, Dominic; Reichelt, Dirk; Holland-Moritz, Peter (2025) : Introducing Covering Problems for the Electronic Wafer Test, Operations Research Forum, ISSN 2662-2556, Springer International Publishing, Cham, Vol. 6, Iss. 3, <https://doi.org/10.1007/s43069-025-00504-2>

This Version is available at:

<https://hdl.handle.net/10419/330538>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/4.0/>



# Introducing Covering Problems for the Electronic Wafer Test

Dominic Deckert<sup>1</sup> · Dirk Reichelt<sup>1</sup> · Peter Holland-Moritz<sup>1</sup>

Received: 27 March 2025 / Accepted: 2 July 2025  
© The Author(s) 2025

## Abstract

Semiconductor manufacturing is an essential area of modern industry. Its yield is influenced by a wide variety of different factors, among them machining precision as well as scheduling and processing concerns. To minimize the amount of faulty products, the semiconductors are examined at several steps in the manufacturing process. This paper presents the wafer covering problem, an optimization problem that arises during the multi-site electrical wafer test. Foundational research in this area is minimal, which this paper aims to amend. Mathematical definitions are introduced for the underlying decision problem. An objective function is presented to minimize probe mark damage and increase manufacturing yield while considering test efficiency as well. The problem is analyzed and references to similar problems in the literature are drawn. It is shown that even the decision version is NP-complete, using a reduction from the well-known set covering problem. The paper concludes by presenting a suite of practically relevant test problems for use in further research. Optimal solutions for many of these test instances are calculated using a constraint satisfaction solver.

**Keywords** Operations research · Set covering · Electrical wafer testing · Multi-site wafer probing · NP completeness · Constraint satisfaction programming

## 1 Introduction

Modern life would be unthinkable without the advances made in the field of microelectronics. Cars, mobile phones, the rapid improvement of computers, and the Internet

---

✉ Dominic Deckert  
dominic.deckert@htw-dresden.de

Dirk Reichelt  
dirk.reichelt@htw-dresden.de

Peter Holland-Moritz  
p.hollandmoritz.paper@gmail.com

<sup>1</sup> Hochschule für Technik und Wirtschaft Dresden – University of Applied Sciences, Dresden, Germany

rely heavily on technology. Without the production of integrated circuits and the semiconductor manufacturing industry, all these products would be unthinkable.

This industry presents one of the most challenging manufacturing scenarios, fabricating structures of a size of only a few thousandths of a centimeter while utilizing high temperatures and several highly reactive chemicals. Precision is paramount in every step of the production process, and any error may lead to significant losses in production time or material. At the same time, the profit margins rely on high tool utilization, which might inadvertently introduce variability into the manufacturing system and reduce overall output [1]. Dealing with these and many other competing objectives in the semiconductor manufacturing process means that any small misstep may have significantly larger consequences.

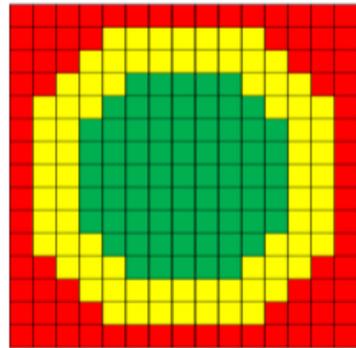
The *dies*, i.e., structurally separate circuits formed on the wafer, are tested at multiple steps in the manufacturing process in order to detect any error as early as possible. This is done so that product quality improves and production targets are met. One of these tests is the *electrical wafer test*, also commonly known as *wafer probing*, during which the conductance and other electrical properties of the die are confirmed using an array of contact needles, arranged as a so-called *probe card*. In *multi-site* probing, several wafers are tested at the same time to reduce test time and improve the efficiency of the probing process [2, 3]. To accommodate different wafer sizes and other constraints of the production—for example, to avoid test structures or contaminants—differently shaped probe cards are utilized. In multi-site wafer tests, the focus of this paper, only one probe card shape can be used during each test process. In a *touchdown*, all dies in the shape of the probe card are tested at the same time. Multiple overlapping touchdowns on the same die may be necessary to achieve full coverage of all the dies on the wafer. However, repeated tests of the same die cause probe mark damage (e.g., [4]), and several dies might become unusable in inefficient testing procedures.

Probe mark damage is one of the less-researched problems of the wafer testing process. While a full literature review is not within the scope of this paper, in Section 2.1, the state of the research is assessed by querying the database IEEEExplore. While many results of the query are weakly related to the multi-site wafer test, none focuses on the problem of minimizing probe mark damage and improving yield directly.

This paper attempts to contribute to the scientific discourse by providing a theoretically founded basis for this problem. The electrical wafer test is analyzed as an abstract mathematical optimization problem, referred to as the *wafer covering problem* (WCP). A formal definition is introduced in Section 2. This definition is used to investigate the theoretical properties of the WCP. Section 3 proves a discouraging result: the wafer covering problem is NP-complete and thus very difficult to solve efficiently. In Section 4, the presented WCP is enhanced with a practically inspired objective function.

Furthermore, this paper introduces a suite of standard wafer covering problems in Section 4 so that further research can be compared on a common set of problems. For all problem instances, constraint satisfaction programming is used to provide the optimal solution or a good lower bound on it.

**Fig. 1** Sketch of a small wafer with required (green), optional (yellow), and prohibited (red) dies



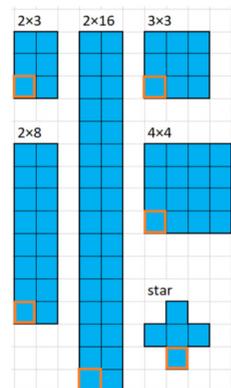
## 2 Wafer Covering

The goal of the wafer covering problem is to find a plan for the electrical wafer test that causes the least probe mark damage to the dies on the wafer. In the multi-site wafer test, multiple *dies* are checked using a probe card in a so-called *touchdown*. Inefficient wafer testing can affect the yield of the semiconductor manufacturing process.

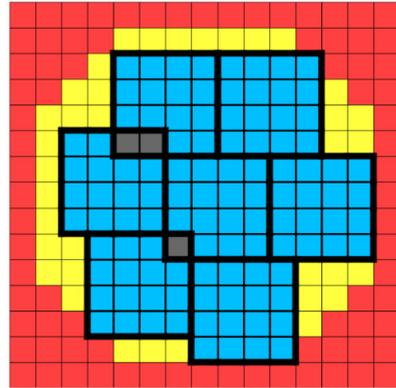
Multiple touchdowns have to be performed in different positions to cover every die of the wafer, avoiding regions that are protected or could contaminate the contact needle. All positions that need to be tested will be referred to as *required*, all positions that can be touched without problem as *optional* and all other positions as *prohibited*. The dies on a wafer are commonly arranged in a grid pattern to allow easy separation by cutting at the end of the manufacturing process. Accordingly, the positions of dies and touchdowns are represented as 2-dimensional integer coordinates. In Fig. 1, a small wafer is sketched. Required dies are marked as green, optional as yellow, and prohibited as red.

Similar to the wafers, the probe cards are arranged according to the orthogonal grid on which the tests are performed. They are represented as collections of 2-dimensional integer vectors that originate from a single position, which in this paper is referred to as the site-0. In Fig. 2, different typical probe cards are displayed. The site-0 is marked.

**Fig. 2** Different probe card shape, with the site-0 marked in orange



**Fig. 3** Wafer, covered using a  $4 \times 4$  probe card



Due to the shape of the probe card and the wafer, a die might be tested multiple times during the same test. Figure 3 illustrates this problem on a small wafer with a large probe card. There, the three dies marked in gray are tested twice. This overlap produces no new information and is in fact problematic for the probe card, since every test requires a *scrub mark* that damages the wafer [5]. The aim of the wafer covering problem is to minimize the probe mark damage and subsequent yield loss due to overlap.

## 2.1 Research Methodology

Although there has been significant innovation of the technology involved in the electrical wafer test, there remains a dearth of literature attempting to optimize the wafer covering problem. This problem is focused on minimizing probe mark damage during the multi-site wafer test. To assess the current state of the literature, the well-known literature database IEEExplore was queried. The query (following below) included all literature on multi-site wafer tests where the probe mark damage is minimized.

```
("All Metadata": "Wafer Prob*" OR "All Metadata": "Wafer
Test*")
AND ("All Metadata": "Multi" OR "All Metadata":
"Parallel")
AND ("All Metadata": "Cover*" OR (("All Metadata":
"Mark" OR "All Metadata": "Damage" OR "All Metadata":
"Overlap") AND ("All Metadata": "Minimization" OR "All
Metadata": "Optimization" OR "All Metadata":
"Reduction")))
```

This query revealed 14 research articles, of which none focused on the wafer covering problem. Only six paper considered multi-site wafer tests, with four considering different improvements to the test technology [6–9] and two presenting the state of the art in 2004 and 2024, respectively [3, 10]. Altogether, this indicates a lack of research on wafer covering.

To fully assess the current research, pertinent citations were followed to and from the discovered papers. The discovered wafer probing research focuses on the improvement of probe card technology, such as the structure of the probe card (e.g., [11] or the wafer prober infrastructure [12]). Other improvements regarded means of virtual defect detection such as data analysis to increase test accuracy [13], defect clustering [14], or improving reprobing decisions and avoiding falsely rejected dies [15].

## 2.2 Formal Definition

The wafer covering is an optimization problem that has to consider both efficiency of the test and probe mark damage, as is discussed in greater detail in Section 4.1. This section presents a simplified formulation of the WCP as a decision problem. This is necessary for the formal underpinning of its complexity analysis.

In the wafer covering decision problem, all valid coverings are limited in the total number of touchdowns they can perform on the wafer. Formally, it can be defined as follows:

**Definition 1** (Wafer Covering Decision Problem) An instance of the wafer covering decision problem can be represented by a tuple  $(D_{req}, D_{opt}, \text{and } PC, k)$ , where

- The die position sets  $D_{req}, D_{opt} \subset \mathbb{Z}^2$  are finite with  $D_{req} \cap D_{opt} = \emptyset$ .
- The probe card  $PC \subset \mathbb{Z}^2$  is finite, with  $(0, 0) \in PC$ .
- And  $k \in \mathbb{N}$  is the number of allowed touchdowns.

Notably, requiring  $(0, 0) \in PC$  can be done without loss of generality. This is true since every problem instance where the probe card  $PC$  does not contain  $(0, 0)$  can be transformed into an equivalent problem that does by choosing any  $p \in PC$  and subtracting  $p$  from all vectors in  $PC$ .  $(0, 0)$  is the site-0 position. Later, arguments are simplified by this requirement.

To define the notion of solutions, a more precise definition of touchdowns is required. Every touchdown can be represented by its site-0 position.

**Definition 2** For an instance  $P = (D_{req}, D_{opt}, PC, k)$  of the WCP as defined in Definition 1, a touchdown is represented by a position  $t \in \mathbb{Z}^2$ . Then, the *touchdown coverage*  $cvg(t)$  can be defined as  $cvg(t) = \{t + p \mid p \in PC\}$  (where “+” is the standard vector addition). The touchdown  $t$  is called *valid* iff  $cvg(t) \subseteq D_{req} \cup D_{opt}$ . The set of all valid touchdowns for  $P$  is  $val(P)$ .

With this, it is straightforward to define a solution of the WCP:

**Definition 3** (WCP Solution) For an instance  $P$  of the WCP as defined in Definition 1, let  $T \subset val(P)$  be a set of valid touchdowns. The associated *covering*  $Cover(T)$  is defined as  $Cover(T) = \bigcup_{t \in T} cvg(t)$ .  $T$  is called a *solution* for  $P$  if

- $D_{req} \subseteq Cover(T)$
- $|T| \leq k$

If there is a solution for  $P$ , then the instance is called *satisfiable*.

### 3 NP-Completeness Proof

Here, a proof is presented using the definition presented in Section 2 that even the decision version of WCP is NP-complete. Consequently, optimal solutions can not be calculated efficiently. This proof is based on the proof method of polynomial-time reduction from set covering to wafer covering. Both proof method and set covering are introduced in below. Section 3.3 presents the proof of NP-completeness for the WCP.

#### 3.1 Polynomial-Time Reduction $\leq_P$

Complexity theory is a challenging and broad field. Within the scope of this paper, only roughly outlines can be provided of the necessary concepts for the proof below. For a more in-depth introduction to complexity theory, ref. [16] is of particular use.

The field of complexity theory delineates different *complexity classes*, groups of problems that require “similar” resources to be solved exactly. Of particular importance are the classes P and NP, which express problems that have a deterministic or nondeterministic algorithm with polynomial runtime respectively. To this point, no efficient algorithms has been found for the most difficult problems in the class NP, the so-called NP-complete problems.

NP-completeness for a problem class  $\mathcal{Q}$  indicates that no problem in NP can be harder than  $\mathcal{Q}$ . There are problem classes for which this property has been shown explicitly, among them set covering [17]. Polynomial-time reducibility can be used to show that another problem is at least as hard to solve. This can be used to show NP-completeness.

**Definition 4** (Polynomial-Time Reducibility) A problem class  $\mathcal{A}$  is *polynomial-time reducible* to a problem class  $\mathcal{B}$  iff there is a polynomial-time function  $f : \mathcal{A} \rightarrow \mathcal{B}$  such that an instance  $A \in \mathcal{A}$  is satisfiable iff  $f(A) \in \mathcal{B}$  is satisfiable. This is written as  $\mathcal{A} \leq_P \mathcal{B}$  and call  $f$  a *P-Time reduction* from  $\mathcal{A}$  to  $\mathcal{B}$ .

If a problem class is reducible to another, then it cannot be categorically more difficult—otherwise, the problem could be solved using the P-time reduction, causing a contradiction. Accordingly, polynomial-time reducibility can be used to show NP-completeness.

**Lemma 1** *If  $\mathcal{A}$  is NP-complete,  $\mathcal{A} \leq_P \mathcal{B}$  and  $\mathcal{B} \in NP$ , then  $\mathcal{B}$  is NP-complete.*

There can be a difference between the complexity of a decision problem and the corresponding optimization problems. For example, while the decision problem of Set Covering (i.e., “Is there a covering with at most  $k$  subsets?”) is NP-complete, the associated optimization problem (i.e., “What is the smallest  $k$  such that there is a covering with  $k$  subsets?”) is considered NP-hard. In this proof, the focus is on the decision problems underpinning set covering and wafer covering.

### 3.2 Set Covering

The set covering problem (SCP) is one of the most prominent and challenging optimization problems in complexity theory [17]. It describes any problem where an assortment of items has to be covered efficiently using some specified combinations and can be found in a wide array of topics (for an overview, see ref. [18]). Problems of operations research such as resource allocation, including train scheduling [19] or flight crew scheduling [20] as well problems from the field of location optimization problems, such as cell tower placement [21] or hospital ambulance coverage [22], can be modeled as set covering problems.

A wide array of different solution approaches have been proposed for the set covering problem, which may serve as a basis for a solution to the wafer covering problem in further research. Meta-heuristical approaches such as genetic algorithms (e.g., [23, 24]) and ant colony optimization (e.g., [25, 26]) have shown good results. Furthermore, mathematical optimization has proven equally effective, with the Lagrangian relaxation approach presented in ref. [19] consistently achieving very good results.

An SCP instance can be described by the base set  $\mathcal{U}$ , sometimes called the *universe*, of all its elements and the specified subsets  $S$  that can be utilized in the covering. The additional parameter  $k$  describes the maximal number of subsets that can be utilized in a solution. Formally, a set covering problem is defined as follows:

**Definition 5** (Set Covering Decision Problem) An instance of the set covering decision problem can be represented by a tuple  $(\mathcal{U}, S, k)$  such that

- The set  $\mathcal{U}$  is finite.
- $S \subseteq 2^{\mathcal{U}}$  is the set of allowed subsets.
- And  $k \in \mathbb{N}$  is the number of allowed subsets.

A solution to the set covering problem has to cover the universe, using at most  $k$  of the specified subsets. Formally, this can be defined as follows:

**Definition 6** (SCP Solution) Let  $P = (\mathcal{U}, S, k)$  be an instance of the set covering problem as defined in Definition 5. Then, the set  $C \subseteq S$  is called a solution of  $P$  iff

- $\bigcup_{s \in C} s = \mathcal{U}$ .
- And  $|C| \leq k$ <sup>2</sup>

If a solution for  $P$  exists, then  $P$  is called *satisfiable*.

### 3.3 Proof

Here, a proof is presented that even the simple wafer covering decision problem is NP-complete. Following the structure of Lemma 1, it is first proven in Section 3.3.1 that the WCP is in NP by roughly sketching an algorithm. Then, a proof is presented in Section 3.3.2 that set covering is polynomial-time reducible to the wafer covering problem.

<sup>1</sup> Here,  $2^{\mathcal{U}}$  denotes the power set construction of  $\mathcal{U}$ .

<sup>2</sup>  $|\cdot|$  here denotes the cardinality of a set.

### 3.3.1 Wafer Covering is in NP

It is easy to see that wafer covering can be solved in NP. For a WCP instance, a naive algorithm can non-deterministically choose  $k$  touchdowns from  $D_{req} \cup D_{opt}$ . This restriction is possible since every valid touchdown must be on a position from  $D_{req} \cup D_{opt}$  (recall that  $(0, 0)$  is an element of every probe card). Afterwards, the algorithm can check that this touchdown set is a solution by checking whether every required die is tested at least once. This test can be done in at most  $k \cdot |PC|$  steps. If the instance is satisfiable, then this algorithm has an accepting run with polynomial runtime.

### 3.3.2 Set Covering $\leq_P$ Wafer Covering

Here, it is shown that the set covering problem can be reduced to the wafer covering problem. Intuitively, this proof is rather straightforward: the subsets will be transformed into a probe card in such a way that every row represents a subset and every column an element. The required dies are arranged in a single row so that every satisfying wafer covering must consist of touchdowns that in total cover the row (i.e., all elements of the universe). Then, since the relevant row on the required dies represents a subset for every touchdown, this wafer covering must represent a set covering.

**Proof** Let  $P_{SCP} = (\mathcal{U}, S, k)$  be a set covering problem instance as described in Definition 5. Without loss of generality, let  $u_0, \dots, u_{m-1}$  be an ordering of all elements in  $\mathcal{U}$  and  $s_0, \dots, s_{n-1}$  be a sequence of elements in  $S$  such that  $u_0 \in s_0$ . This can be done for every satisfiable set covering problem, since otherwise no combination of subsets from  $S$  could contain  $u_0$  (i.e., the problem would be unsatisfiable). Then,  $P_{SCP}$  can be reduced to a WCP instance  $P_{WCP} = (R, O, PC, k)$  such that

- The probe card is  $PC := \{(i, j) \mid i, j \in \mathbb{N}, i < m, j < n, u_i \in s_j\}$ .
- The required dies are  $R := \{(i, 0) \mid i \in \mathbb{N}, i < m\}$ .
- The optional dies are  $O := \{(i, j) \mid i \in \mathbb{N}, i < m, j \in \mathbb{Z}, |j| \leq n - 1, j \neq 0\}$ .<sup>3</sup>

Notably, with this definition,  $(0, 0) \in PC$  since  $u_0 \in s_0$ .

Now, it must be shown that  $P_{SCP}$  is satisfiable iff  $P_{WCP}$  is satisfiable. Within this paper, one direction is shown. The other follows analogously.

$\Leftarrow$ : Let  $T$  be a solution for  $P_{WCP}$ . Notably, every element in  $T$  must be of the form  $(0, y)$ , since otherwise the touchdown would not be valid: there must be an element  $(m - 1, p_y) \in PC$  for some  $p_y \in \mathbb{N}$ , and therefore,  $(x, y) + (m - 1, p_y) \notin R \cup O$  for any  $x > 0$  (since  $x + (m - 1) \not\leq m$ ).

Now, it is easy to construct a solution candidate  $C$  for the set covering problem from  $T$ : let  $C$  be the set such that  $s_j \in C$  iff  $(0, -j) \in T$ . To show that  $C$  is a solution for the set covering, it must be shown that  $\bigcup_{s \in C} s = \mathcal{U}$  and that  $|C| \leq k$ .

- Let us assume that  $\bigcup_{s \in C} s \subsetneq \mathcal{U}$ . Then, there must be an  $\hat{i}$  with  $u_{\hat{i}} \in \mathcal{U}$  such that  $u_{\hat{i}} \notin \bigcup_{s \in C} s$ . Accordingly, it is true for all  $j$  with  $s_j \in C$  that  $u_{\hat{i}} \notin s_j$ , and consequently,

<sup>3</sup> The bounds of  $O$  had to be chosen in such a way that  $PC$  can perform a valid touchdown with every of its  $n$  rows.

$(\hat{i}, j) \notin PC$ . However, according to the construction, for all those  $j$ , it also holds that  $(0, -j) \in T$ , and thus,  $(0, -j) + (\hat{i}, j) = (\hat{i}, 0) \notin cvg((0, -j))$ . This would entail  $(\hat{i}, 0) \notin Cover(T)$ , and thus,  $T$  would not be a solution of the wafer covering problem.

Consequently,  $\bigcup_{s \in C} s \supseteq \mathcal{U}$  must be true. And since all subsets of  $C$  are subsets of  $\mathcal{U}$ , the conjunction cannot be a proper superset of  $\mathcal{U}$ , meaning that  $\bigcup_{s \in C} s = \mathcal{U}$  must hold.

- Since  $T$  is a solution of the wafer covering problem, it must hold that  $|T| \leq k$ . Furthermore, according to the construction of  $C$ , the sets  $C$  and  $T$  have the same number of elements. Thus,  $|C| \leq k$ .

Therefore, the set covering problem  $P_{SCP}$  is satisfiable if the wafer covering problem  $P_{WCP}$  is satisfiable. Furthermore, it is easy to see that the reduction can be performed in polynomial time:  $R$  can be constructed in  $|\mathcal{U}|$  steps,  $O$  can be constructed in  $|\mathcal{U}| \cdot (2 \cdot |S| - 1)$  steps, and  $PC$  in less than  $|\mathcal{U}| \cdot |S|$  steps. Thus, set covering is reducible to wafer covering, and with the result from Section 3.3.1, the WCP is NP-complete. □

## 4 WCP Benchmarks

Realistically, occurring problems require more complex constraints and objectives than the simple touchdown restriction presented in Section 2. Below, a more practically relevant objective function is presented for the wafer covering optimization problem. Furthermore, this paper introduces a suite of abstracted problem instances so that interested researcher can compare their results on these problems. Constraint satisfaction programming is used to provide optimal solutions for these problem instances where feasible and solution bounds for all other cases.

### 4.1 Wafer Covering Objective Function

One goal of the wafer covering should be to minimize the potential damage done to the required dies during the electrical wafer test. To this end, every die should be tested as few times as possible. Especially in the context of the semiconductor manufacturing process as a whole—during which the same wafer is fully tested multiple times at different steps to recognize defects as soon as possible and avoid unnecessary manufacturing costs—it is important that every individual test avoids repeated tests on the same die.

The wafer covering optimization problem (WCOP) is defined so that it minimizes the strain put on every single die. Its formal definition can be formulated very similarly to the definitions presented in Section 2.

**Definition 7** (Wafer Covering Optimization Problem) A wafer covering optimization problem is a tuple  $P_{WCOP} = (D_{req}, D_{opt}, PC, b)$  where:

- The die position sets  $D_{req}, D_{opt} \subset \mathbb{Z}^2$  are finite with  $D_{req} \cap D_{opt} = \emptyset$ .

- The probe card  $PC \subset \mathbb{Z}^2$  is finite.
- The objective function parameter  $b \in \mathbb{R}$  satisfies  $b > 1$ .

For the WCOP, let  $t \in \mathbb{Z}^2$  be a touchdown analogous to the definition presented in Definition 2. Accordingly, the *touchdown coverage*  $cvg(t)$  is defined as  $cvg(t) = \{t + p \mid p \in PC\}$ . The touchdown  $t$  is called *valid* iff  $cvg(t) \subseteq D_{req} \cup D_{opt}$ . The set of all valid touchdowns for  $P_{WCOP}$  is  $val(P_{WCOP})$ .

The objective function of the WCOP minimizes the probe mark damage on the required dies by assigning every required die a touchdown penalty based on its number of touchdowns and minimizing the sum of these penalties. To avoid a situation where a die is tested several times early on in the manufacturing process, resulting in a significant risk of defectiveness by the end of production, the touchdown penalty is chosen to grow exponentially with the number of touchdowns in the individual test.

**Definition 8** For a WCOP instance  $(D_{req}, D_{opt}, PC, b)$  as defined in Definition 7, let  $T \subset val(P_{WCOP})$  be a set of valid touchdowns.  $T$  is said to satisfy  $P_{WCOP}$  iff  $D_{req} \subseteq \bigcup_{t \in T} cvg(t)$ .

Let the related touchdowns  $rel(T, d)$  for a required die  $d \in D_{req}$  be defined as  $rel(T, d) := \{t \mid t \in T, cvg(t) \cap d \neq \emptyset\}$  and accordingly the touchdown count as  $tdc(T, d) := |rel(T, d)|$ .

Then, the objective function  $f$  can be defined as  $f(T) = \sum_{d \in D_{req}} b^{tdc(T, d)-1}$  with  $f \rightarrow min$ .

Notably, the exponent in the objective function is reduced by 1. This is done since the smallest touchdown count for a required die is 1. Accordingly, the range of objective function values is reduced without changing the behavior of the optimization problem or producing non-integer objective function values.

Notably, the properties of an optimal solution of a WCOP instance heavily depend on the value of  $b$ . The higher the value of  $b$ , the more “expensive” (in terms of objective value penalty) many touchdowns intersecting on the same die becomes. The smallest integer value for  $b$  is 2. An integer value is advantageous since it allows for exact objective value calculation and representation. Furthermore,  $b = 2$  encourages solutions that avoid many-touchdown overlaps where possible while not being so steep as to not allow them where necessary. The benchmark problem suite is therefore chosen with  $b = 2$ .

Other objective functions have also been considered. The most relevant among them were the maximum touchdown count  $mtdc$  of required dies and the number of touchdowns. Both have flaws that the presented objective function avoids: while  $mtdc$  avoids high probe mark damage on any single die, it is not sensitive to the overall probe mark damage incurred. Furthermore, it is just as insensitive to the efficiency of the covering (in terms of touchdowns required to finish the test). The number of touchdowns, on the other hand, presents a very efficient covering and might however cause yield losses by allowing very concentrated touchdown overlaps.

## 4.2 Benchmark Problem Suite

This section presents the suite of test problems for the wafer covering problem. These test problems and their underlying modeling decisions aim to cover a wide area of the testing scenarios, but can not claim to be exhaustive. The problem suite models three categories of test scenarios: complete wafer tests as well as tests on contaminated wafers and wafers with test structures. In the complete problem, all prospective dies on the wafer have to be tested, i.e., besides the optional “buffer” dies at the edge of the wafer, every die on the wafer is required.

In the contamination problem in comparison, there is an area on the die that has been contaminated and should therefore not be touched by the probe card. Dies adjacent to the confirmed contamination are considered likely defective and therefore optional. In the suite of problem instances, the contamination is represented by a 45° diagonal of forbidden dies across the entire wafer, crossing the middle. Adjacent dies to the diagonal are marked as optional.

On a wafer with test structures, there are individual dies that are designated as test structures by an engineer. These structures are considered sensitive and should not be touched, but all adjacent non-test dies remain required. In the problem instances, the test structures are located in a regular pattern of forbidden dies at the intersections of every fourth row and column.

Wafers of a wide range of different sizes are processed in semiconductor manufacturing. The smallest relevant wafers have a “diameter” of 15 required dies, while the largest exceed 200. In order to test the behavior of solution algorithms on different problem sizes, this problem suite incorporates problem instances with a range of different probe card diameter and test problem instances of diameters 15, 31, 51, 101, and 151. While wafer with diameters over 150 dies are not uncommon in the manufacturing process, the chosen solution paradigm exceeded available resources for problem diameters of 200 or more wafers. For this reason, the diameters 201 and 251 are included in the set of problems, but no benchmarks can be provided.

Similarly, during the manufacturing process, probe cards of different shapes can be utilized. Standard shapes include rectangular and square shapes of different lengths since such shapes are easy to align while maintaining small overlap. Due to technical limitations, rectangular probe cards with a width of 2 are easiest to manufacture. In the test instances, there are three rectangular probe cards of different lengths (3, 8, and 16) as well as two square shapes— $3 \times 3$  and  $4 \times 4$ . The square shape  $2 \times 2$  is rarely utilized and therefore not included. Furthermore, a “star”-shaped probe card with 5 contact needles is included to analyze the behavior and difficulties on non-rectangular probe cards.

With 5 wafer diameters, 3 wafer types and 6 probe card shapes, there are a total of  $5 \cdot 6 \cdot 3 = 90$  WCOP instances that are analyzed. Notably, in the test suite not every combination of wafer diameter and probe card is feasible—for example, any wafer with test structures only has 3 required dies between test structures, meaning that  $4 \times 4$  or larger probe cards could not test these dies without also touching a sensitive test structure. Of the 90 problem instances, 64 are feasible.

The problem instances can be found at <https://github.com/ddeckertHTW/WaferCovering>. There, the wafers are represented as 2-dimensional arrays where the value 1 represents a required die, 2 represents an optional die, and 0 represents positions that are neither (and should not be touched).

It should be noted that the details presented in this paper are focused on solving the WCP as it presents itself at Robert Bosch Semiconductor Manufacturing Dresden GmbH. The problem instances described here, although based on these realistically occurring problems, have been generalized to be applicable beyond the limits of this one manufacturing scenario.

### 4.3 Exact Solutions

*Constraint satisfaction programming* was utilized in order to produce good solutions for the WCOP problem instances introduced above and be able to prove their quality. Its straightforward modeling process and the availability of a well-performing solver make constraint satisfaction a suitable solution paradigm for the problems.

In constraint satisfaction, a satisfiability problem is represented by a set of *decision variables* and a set of *constraints* on these variables. The decision variables are each assigned domains of potentially feasible values, whereas the constraints represent any restrictions on the possible combinations of values from these domains. These constraints can then be combined, propagated and otherwise manipulated to allow an efficient traversal of the solution space. Constraint satisfaction programming can be utilized for optimization problems as well, finding optimal values for a given objective function. This functionality is included in many constraint satisfaction solvers. For a more in-depth introduction to constraint satisfaction, see ref. [27].

The constraint satisfaction program presented in this article is written in python, using the `cpmpy` library [28]. The library offers support for array structures of variables with the same domains, utilizing the well-performing `numpy` arrays as internal representation. `Cpmpy` aids in instantiation and precomputation, passing the simplified model to an external constraint satisfaction solver. In this paper, the program is solved using the award-winning OR-Tools constraint satisfaction solver [29]. Array structures allowed a very straightforward modeling of the WCP as a constraint satisfaction program. The code can be found at <https://github.com/ddeckertHTW/WaferCovering>.

For a WCOP  $(D_{req}, D_{opt}, PC, b)$ , let  $D \subset \mathbb{Z}^2$  be a 2-dimensional grid such that all required and optional die positions  $(x, y)$  can be represented in  $D$ . The constraint satisfaction program here is presented in mathematical language. These constraints are formulated in the same manner as the python code of the program. It can be expressed as follows:

- Decision variables:
  - $touch[x, y] \in \{0, 1\}$  for all  $(x, y) \in D$
  - $tdc[x, y] \in \{0, 1, \dots, |PC|\}$  for all  $(x, y) \in D$
- Constraints:
  - $CI: touch[x - p_1, y - p_2] = 0$  for all positions  $(x, y) \in D \setminus (D_{req} \cup D_{opt})$  and for all  $(p_1, p_2) \in PC$ .

- $C2: tdc[x, y] = \sum_{(p_1, p_2) \in PC} touch[x - p_1, y - p_2]$  for all required positions  $x, y \in D_{req}$ .
- $C3: tdc[x, y] \geq 1$  for all required positions  $(x, y) \in D_{req}$ .

- Objective Function:

$$\sum_{(x, y) \in D_{req}} tdc[x, y]$$

Constraint  $C1$  expresses that no part of a probe card touchdown should land on a prohibited position, while constraint  $C2$  defines the manner in which the touchdown count is calculated. Lastly, constraint  $C3$  enforces that every required die is touched at least once. Notably,  $\sum$  is used to denote a sum of finite elements in constraint  $C2$ . This sum can be easily instantiated, i.e., transformed into an explicit sum of ground variables. This instantiation is performed by `cpmpy` in the program code.

## 4.4 Results

In its traditional form, constraint satisfaction is an exact solution paradigm. It may not be able solve an NP-complete optimization problem in a reasonable amount of time. With the OR-Tools CP solver, the constraint satisfaction program presented above can calculate optimal solutions for problems with a wafer diameter of at most 50 dies reliably. For instances with up to 150 dies, the program is capable of solving most (but not all) instances with. For those instances it cannot solve exactly in an acceptable amount of time, the constraint satisfaction program can still be used to calculate meaningful lower bounds on the optimal objective function value.

### 4.4.1 Experimental Setup

The algorithm is used to produce optimal solution values or bounds for all problem instances. The calculations were performed on a Celsius workstation with 251 Gbyte RAM and 32 physical cores (Intel Xeon Gold 5220 CPU @ 2.20 GHz), using the cores to perform parallel calculations. Due to limited calculation resources, a maximum of 24h of calculation times was allotted for each instance (not including any model optimization or other precomputations `cpmpy` performed).

The OR-Tools solver uses parallelism in an ensemble setup, i.e., separate threads attempt different solution strategies for the same problem. The number of solver threads is set to 32. This allows all solution strategies to be utilized [30].

### 4.4.2 Hot Start Improvement

A significant improvement to the results of more complex optimization problems lies in the utilization of a precomputed solution. This is called a *hot start* (see, e.g., [31]). It allows the algorithm to use the solution as an upper bound (and cut off any areas of the solution space that would result in solutions with worse objective values than this initial solution). Furthermore, valid (and especially, good) solutions are usually

situated very closely in the solution space, so that an initial good solution allows the solver to explore a good area of the solution space before worse areas.

The hot start solution can be used to guide the exploration of the solution space. Solvers for constraint satisfaction programs regularly utilize operations that reduce one or more of the domains of the decision variables or set a variable to a specific value. Using the `solution_hint` functionality, the OR-Tools solver can be guided on which value assignment to perform first.

A straightforward greedy search method has been used to construct the initial solution. The best touchdowns are repeatedly chosen according to a fairly simple evaluation heuristic: the touchdown receives a reward according to the number of newly covered required die, while every new overlap of  $o$  touchdowns on a required die entails a penalty of  $b^{o-1}$ . This hot start approach significantly improves the chance that the constraint satisfaction program produces a solution as well as the produced objective bound.

#### 4.4.3 Optimal Results and Objective Value Bounds

Table 1 shows the best objective values found as well as the associated lower bounds “LB” for all problem instances. For a few problem instances, the constraint satisfaction program did not find a solution and defaulted to the hot start solution value instead. These solution values are supplied in parentheses. Any optimal solution value is marked with \* in the table.

To compare these solutions and assess their quality, two metrics are utilized: the *optimality gap* and the *overlap ratio*. The optimality gap is an established measure of solution quality that compares the best found value  $b$  and the lower bound  $l$ . It is calculated as  $\frac{b-l}{b}$  (since  $b$  and  $l$  are positive). The ratio of required dies that have to be tested multiple times in the solution will be referred to as the overlap ratio. This value is thus a measure for the least possible loss that can be achieved. Both metrics are always non-negative and should be minimized. The optimal value is therefore 0.

The solution values presented here constitute a reasonable basis for further research into the wafer covering problem. Overall, the solution quality is high: In more than 70% of the problem instances, optimality has been proven for the solution. Furthermore, the average optimality gap is less than 1%, with the maximum optimality gap at 11.25%. However, it must also be noted that the results are less satisfying for the larger problem instances: less than 50% of the WCOP instances for wafers with a diameter of 151 dies are optimal, and there are multiple instances where the solver did not find a better solution than the pregenerated hot start solution.

To add to that, the solution values posted here allow some general statements about the relative difficulty of the different problem types to be made (at least, as it pertains to the constraint satisfaction solver). It should be noted that while the median is usually regarded as the better approach to getting a general, outlier-resistant statement about the trend of the distribution, it is not an insightful measure for the optimality gap. This is caused by the fact that for all problem types a majority of solutions is optimal, with the median optimality gap consequently being 0. Table 2 displays the average and maximum values for the optimality gap and median and maximum values for the overlap ratio of the different problem types.

**Table 1** Optimal values for the WCOP instances

Size	PC shape	Complete wafer		Contamination		Test structures	
		Objective	LB	Objective	LB	Objective	LB
15×15	2×3	69*		50*		70*	
	2×8	87*		—		—	
	2×16	—		—		—	
	3×3	69*		56*		68*	
	4×4	72*		80*		—	
	star	69*		52*		74*	
31×31	2×3	517*		462*		560*	
	2×8	558*		—		—	
	2×16	652*		—		—	
	3×3	517*		508*		553*	
	4×4	540*		582*		—	
	star	517*		480*		577*	
51×51	2×3	1649*		1554*		1817*	
	2×8	1689*		—		—	
	2×16	1776*		—		—	
	3×3	1653*		1644	1635	1810*	
	4×4	1689*		1780*		—	
	star	1649*		1588*		1876	1811
101×101	2×3	7209*		7008*		8032*	
	2×8	7279*		—		—	
	2×16	7490*		—		—	
	3×3	7214	7211	7221	7149	7988*	
	4×4	7380	7248	(7516)	7432	—	
	star	7209*		7086	7064	8297	7855
151×151	2×3	16,717*		16,414*		18,680	18,680
	2×8	(17,042)	16,906	—		—	
	2×16	(17,650)	16717	—		—	
	3×3	16,727	16,717	16,876	16,612	18,657*	
	4×4	(16,968)	16,717	(17,410)	16,406	—	
	star	16,717*		(16,568)	16,452	19,890	17,653

**Table 2** Optimality gap and overlap ratio of the WCOP problem types

Type	Optimality gap		Overlap ratio	
	avg	max	med	max
Complete	0.326%	5.286%	0.243%	26.112%
Contamination	0.550%	5.767%	3.500%	60.0%
Test structures	1.336%	11.247%	18.148%	26.874%
Total	0.633%	11.247%	3.897%	60.0%

It should be noted in the table that the maximum overlap ratio of 60% for contamination instances can be considered an outlier. The associated instance—a 15 die wafer with the  $4 \times 4$  probe card—has the highest proportion of required dies to probe card size. Excluding it, the contamination problems have a maximum overlap ratio of 25.974%, not significantly different from the other problem types.

Both the challenge a problem instances poses and the minimum overlap necessary appear to increase with the restrictions the wafer introduces. The complete problem, with no additional constraints, is the easiest category of problem to solve. Both optimality gap and overlap ratio are below average. Problems with a contamination introduce a central forbidden area, but it is couched in optional dies that relax the strictness of the constraints thus introduced. Problems with test structures, on the other hand, seem to pose the most difficulty to the algorithm. With more than double the average optimality gap and a significantly higher overlap ratio, the test structure problem stands out above the other types. They also introduce the strongest restrictions, since every die adjacent to a test structure must still be covered. Furthermore, this category introduces the most forbidden dies, spread over the full area of the wafer.

Similarly, the overall solution quality decreases when the problem size increases beyond what the constraint satisfaction algorithm can handle. Table 3 presents the maximum and average optimality gaps for the different problem sizes. The optimality gap is significantly increasing with growing problem size. However, even for the largest instances, the overall solution quality is sufficient for a general benchmark. The worst result (covering the 151 test structure wafer with the star probe card) can be considered an outlier to the overall solution quality of this problem suite. Its optimality gap is almost double the optimality gap of the next-worst problem instance.

**Table 3** Optimality gap of the WCOP problem sizes

Wafer diameter	Optimality gap	
	avg	max
15	0%	0%
31	0%	0%
51	0.309%	3.465%
101	0.737%	5.767%
151	1.336%	11.247%

**Table 4** Ratio of calculation runs stopped after 24 h

Size	Stopped runs
15	0%
31	0%
51	23.077%
101	46.154%
151	61.538%

#### 4.4.4 Calculation Runtime

As mentioned in Section 4.4.1, each calculation run was allocated a maximum of 24 h on the workstation. Within these limitations, more than 25% of the problem instances were stopped after reaching this maximum and were not solved optimally. As can be seen in Table 4, the larger problem instances in particular posed a significant challenge to the calculation.

Overall, the proposed constraint satisfaction program has provided near-optimal benchmarks for a good portion of the problem instances, although at a significant cost. Unfortunately, the program cannot be said to be a practical solution, especially for the largest problem instances.

## 5 Conclusion

The wafer test is an essential process step to ensure high quality semiconductor product. In the quickly changing modern world, semiconductor products are essential to the function of anything from automotive to electronics. High quality and high yield are central goals to the operation of semiconductor manufacturers. At the same time, exhaustive tests can become an impediment to the fabrication process and impacting the manufacturer in other ways. Multi-site wafer testing has proven to be a cost-efficient improvement of the testing process, but runs the risk of increased probe mark damage and yield decrease if dies are unnecessarily tested in overlaps. The wafer covering problem aims to minimize this risk while preserving the efficiency of the testing process.

In this paper, an abstract formalization of the wafer covering problem was presented. This problem has been successfully shown to be NP-complete and thus difficult to solve efficiently. The focus of this proof was on the decision version of the problem—it can be argued that the optimization problem of wafer covering might be even more difficult to solve, since the corresponding optimization version of the set covering problem is strongly NP-hard. Most of the objective functions considered for the practical wafer covering problem are more complex than the WCP and thus should not be expected to be in an easier complexity class. Furthermore, the wafer covering optimization problem was introduced. It incorporates a realistically meaningful objective function, motivated by minimizing the damage to every individual die. A suite of problem instance was presented as well. Using a constraint satisfaction solver, solutions as well as upper

and lower bounds were calculated for these problem instances. For many solutions, optimality has been shown, while even for the other cases, the optimality gap of the solutions was encouraging. Furthermore, these results were used to compare the different types of problem instances. With at most 150 wafers diameter, the problem instances cover a good portion of the semiconductor manufacturing process, though larger wafers are not unheard of and may be of interest to further research.

Although an analytical solution for the WCP is thus very unlikely to be feasible—as is to be expected of an NP-hard problem—there is a chance that heuristical solutions may be feasible. There is fruitful ground for further research in this area. The best-performing approaches for the SCP can be expected to be particularly well-suited: with the structural similarities outlined in this paper, it is not unreasonable to expect solution paradigms such as ant colony optimization (e.g., [32]), genetic algorithms (e.g., [23]), or Lagrangian relaxation (e.g., [19]) to prove effective for wafer covering as well. Of equal interest are the challenges inherent to the repeated wafer test. This problem occurs naturally during the manufacturing process, since an optimal solution to the wafer covering problem in one iteration will—if repeated—place overlaps on the same dies. This would unnecessarily damage specific dies and lead to higher yield losses. Instead, this should be treated as a separate optimization problem that minimizes the touchdown counts where each die is already assigned a previous touchdown count.

**Acknowledgements** The authors would like to thank Rebekah-Anne Oosthuizen for her valuable input on improving the language and flow of the paper during the review process. The publication of this article is funded by the Open Access Publication Fund of Hochschule für Technik und Wirtschaft Dresden - University of Applied Sciences.

**Author Contribution** The problem description and formal definitions were based on work by P.H.-M. Furthermore, P.H.-M. produced the figures used in the article. The proof, solution program and text of the paper was written by D.D. Throughout the process, D.R. provided valuable insight, guidance and formal expertise. All authors reviewed the paper.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Data Availability** Data is provided within the manuscript and can be reproduced using the files stored at <https://github.com/ddeckertHTW/WaferCovering>

## Declarations

**Conflict of Interest** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Mönch L, Fowler JW, Mason SJ (2013) Production planning and control for semiconductor wafer fabrication facilities. *Oper Res/Comput Sci Interfaces Ser 52*. Springer New York NY
2. Rivoir J (2004) Parallel test reduces cost of test more effectively than just a cheap tester. In: IEEE/CPMT/SEMI 29th international electronics manufacturing technology symposium (IEEE Cat. No. 04CH37585), IEEE, pp 263–272
3. Bernardi P, Cardone L, Foscale T (2024) Exploring trade-offs in multi-site wafer testing. In: 2024 IEEE 25th latin american test symposium (LATS), IEEE, pp 1–4
4. Hotchkiss G, Ryan G, Subido W, Broz J, Mitchell S, Rincon R, Rolda, R, Guimbaolibot L (2001) Effects of probe damage on wire bond integrity. In: 2001 Proceedings. 51st Electronic components and technology conference (Cat. No. 01CH37220), IEEE, pp 1175–1180
5. Lorenz G, Stephan T, Petzold M (2012) Characterizing the chip damage potential during wafer probing of highly integrated devices. In: 2012 4th electronic system-integration technology conference, IEEE, pp 1–5
6. Ahmed I, Baraiya S, Singhal R (2018) Case study on low pin count testing of industry transceiver chip. In: 2018 IEEE 27th North Atlantic test workshop (NATW), IEEE, pp 1–5
7. Saikiran M, Ganji M, Chen D (2020) Robust dft techniques for built-in fault detection in operational amplifiers with high coverage. In: 2020 IEEE international test conference (ITC), IEEE, pp 1–10
8. Peng CH, Yang C, Tsu A, Tsai CJ, Chen Y, Lin CY, Hong K, Kao K, Liang P, Tsai CL et al (2014) A novel rf self test for a combo soc on digital ate with multi-site applications. In: 2014 International test conference, IEEE, pp 1–8
9. Noun Z, Cauvet P, Flottes ML, Andreu D, Bernard S (2008) Wireless test structure for integrated systems. In: 2008 IEEE international test conference, IEEE Computer Society, pp 1–1
10. Mann WR, Taber FL, Seitzer PW, Broz JJ (2004) The leading edge of production wafer probe test technology. In: 2004 International conference on test, IEEE, pp 1168–1195
11. Shih M-K, Lai Y-S (2009) Design optimization of needle geometry for wafer-level probing test. *IEEE Trans Compon Packag Technol* 32(2):435–439
12. Lee SH, Park JH, Lee YW (2025) Design and implementation of test infrastructure for higher parallel wafer level testing of system-on-chip. *IEEE Access*
13. Chang BR, Tsai HF, Wu YR (2024) Detection and prediction of probe mark damage in wafer testing. *Electron* 13(20)
14. Singh AD, Krishna CM (1992) Chip test optimization using defect clustering information. *Digest of Papers. FTCS-22: The 22nd international symposium on fault-tolerant computing*, pp IEEE, Boston MA USA, pp 366–373
15. Horng SC, Tsou HT (2010) Apply genetic algorithm to minimize the overkills in wafer probe testing. In: 2010 5th IEEE conference on industrial electronics and applications, pp 593–598
16. Homer S, Selman AL (2011) *Computability and complexity theory*, vol 194. Springer Boston MA Springer, US
17. Karp R (1972) Reducibility among combinatorial problems. *Complex Comput Computations* 85–103
18. Balas E (1982) A class of location distribution and scheduling problems: modelling and solution methods. *JORBEL-Belg J Oper Res Stat Comput Sci* 22(2):36–69
19. Caprara A, Toth P, Fischetti M (2000) Algorithms for the set covering problem. *Ann Oper Res* 98(1):353–371
20. Wedelin D (1995) An algorithm for large scale 0–1 integer programming with application to airline crew scheduling. *Ann Oper Res* 57(1):283–301
21. Chen L, Yuan D (2010) Solving a minimum-power covering problem with overlap constraint for cellular network design. *Eur J Oper Res* 203(3):714–723
22. Repede JF, Bernardo JJ (1994) Developing and validating a decision support system for locating emergency medical vehicles in Louisville Kentucky. *Eur J Oper Res* 75(3):567–581
23. Beasley JE, Chu PC (1996) A genetic algorithm for the set covering problem. *Eur J Oper Res* 94(2):392–404
24. Lan G, DePuy GW, Whitehouse GE (2007) An effective and simple heuristic for the set covering problem. *Eur J Oper Res* 176(3):1387–1403
25. Ren ZG, Feng ZR, Ke LJ, Zhang ZJ (2010) New ideas for applying ant colony optimization to the set covering problem. *Comput Ind Eng* 58(4):774–784

26. Lessing L, Dumitrescu I, Stützle T (2004) A comparison between ACO algorithms for the set covering problem. In: Ant colony optimization and swarm intelligence, vol 3172. Springer Springer Berlin, Heidelberg . Series Title: Lecture Notes in Computer Science, pp 1–12
27. Apt K (2003) Principles of constraint programming. Cambridge University Press, Cambridge
28. Guns T (2019) Increasing modeling language convenience with a universal n-dimensional array cppy as python-embedded example. In: Proceedings of the 18th workshop on constraint modelling and reformulation at CP (Modref 2019), vol 19
29. Google (2011) Google OR-Tools. <https://developers.google.com/optimization>. Online resource last Accessed 28 Feb 2025
30. Google (2024) OR-Tools troubleshooting. Accessed at <https://github.com/google/or-tools/blob/stable/ortools/sat/docs/troubleshooting.md#improving-performance-with-multiple-workers>. Last Access 04 Mar 2025
31. Demirović E, Stuckey PJ (2018) Constraint programming for high school timetabling: a scheduling-based model with hot starts. In: Van Hoesel WJ. (ed) Integration of constraint programming artificial intelligence and operations research, vol 10848. Springer Springer Cham, pp 135–152
32. Sundar S, Singh A (2012) A hybrid heuristic for the set covering problem. Oper Res Int J 12(3):345–365. <https://doi.org/10.1007/s12351-010-0086-y>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.