



HAL
open science

Theoretical Complexity of Grid Cover Problems Used in Radar Applications

Yann Briheche, Frédéric Barbaresco, Fouad Bennis, Damien Chablat

► **To cite this version:**

Yann Briheche, Frédéric Barbaresco, Fouad Bennis, Damien Chablat. Theoretical Complexity of Grid Cover Problems Used in Radar Applications. *Journal of Optimization Theory and Applications*, 2018, 10.1007/s10957-018-1354-x . hal-01862803

HAL Id: hal-01862803

<https://hal.science/hal-01862803v1>

Submitted on 17 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Theoretical complexity of grid cover problems used in radar applications

Briheche Yann · Barbaresco Frederic ·
Bennis Fouad · Chablat Damien

Received: 31 December 2017 / Accepted: 23 July 2018

Abstract Modern radars are highly flexible, using digital antennas which can dynamically change the radar beam shape and position through electronic control. Radar surveillance is performed by emitting sequentially different radar beams. Optimization of radar surveillance requires finding, among a collection of available radar beams with different shapes and positions, a minimal subset of radar beams which covers the surveillance space, ensuring detection while minimizing the required scanning time.

Optimal radar surveillance can be modelled by grid covering, a specific geometric case of set covering where the universe set is laid out on a grid, representing the radar surveillance space, which must be covered using available subsets, representing the radar beams detection areas. While the set cover problem is generally difficult to solve optimally, certain geometric cases can be optimized in polynomial time.

This paper studies the theoretical complexity of grid cover problems used for modelling radar surveillance, proving that unidimensional grids can be covered by strongly polynomial algorithms based on dynamic programming, whereas optimal covering of bidimensional grids is generally non-deterministic polynomially (NP) hard.

This work is partly supported by a DGA-MRIS scholarship.

This is a pre-print of an article published in Springer Journal of Optimization Theory and Applications. The final authenticated version is available online at: <https://doi.org/10.1007/s10957-018-1354-x>

Y. Briheche, F. Barbaresco
THALES AIR SYSTEMS, Voie Pierre-Gilles de Gennes, 91470 Limours, France
E-mail: yanis.briheche@thalesgroup.com, frederic.barbaresco@thalesgroup.com

Y. Briheche, F. Bennis, D. Chablat
Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004, 44321 Nantes, France
E-mail: yann.briheche@ls2n.fr, fouad.bennis@ec-nantes.fr, damien.chablat@cnrs.fr

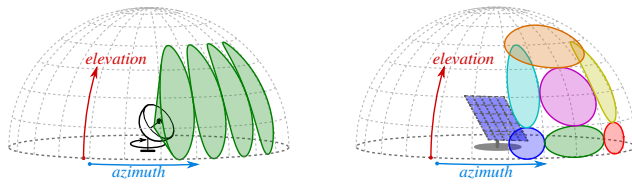


Fig. 1 Search pattern for a rotating radar (left) and a modern electronic radar (right)

Keywords Set cover problem · Grid covering · Combinatorial optimization · NP-hardness · Radar ressource management

Mathematics Subject Classification (2000) 05B40 · 52C15 · 90C39

1 Introduction

Set covering is a fundamental problem in combinatorial optimization and complexity, and has been extensively studied since its description as one of Karp’s 21 classical NP-complete problems [1], which is the common class for difficult industrial problems. The set cover problem is also hard to approximate: while the greedy heuristic has a logarithmic approximation ratio in the number of constraints in both weighed and unweighed cases [2,3], the problem cannot really be more efficiently approximated unless $P=NP$ [4,5,6]. Alternate approximation bounds have also been found using randomized rounding algorithms [7]. In practice, the branch-and-bound approach for integer programming is usually efficient in most case [8], despite lacking good worst-case complexity, as branch-and-bound could be equivalent to complete enumeration in worst cases [9]. On the other hand, certain geometric cover problems which are specific cases of the set cover problem can be solved or approximated in polynomial time [10,11,12].

Optimization of radar search patterns can naturally be formulated as set covering [13]. Active electronic scanned antennas can dynamically perform beam-forming and beam-steering, controlling the shape and steering direction of the radar emission pattern. Modern radars offer more possibilities for designing search patterns, beyond the mechanical limitations of rotating scanning radars, which repeat at each azimuth the same pattern, see Figure 1. Furthermore, modern radars are usually required to perform multiple functions in operation: radar scanning, target tracking, environment analysis, etc. [14,15]. In the context of electronic warfare, minimizing the scanning time frees resources for other functions and ensures early target detection, maximizing available reaction time.

Radar cover problems have different theoretical properties, depending on the considered radar model parameters:

- the azimuthal range of the radar which can be bounded or fully circular
- the feasible geometrical shapes for the emitted radar beams
- whether the radar detects in bidimensional or tridimensional coordinates.

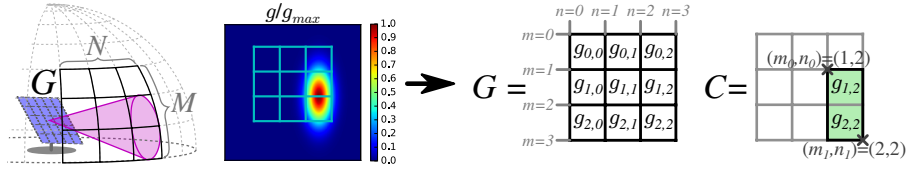


Fig. 2 Radar emission and radiation pattern (left), detection grid G and detection discrete cover C (right)

$$\mathcal{C} = \left\{ \begin{array}{c} T_1=2 \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \\ , \\ \begin{array}{c} T_2=2 \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \\ , \\ \begin{array}{c} T_3=2 \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \\ , \\ \begin{array}{c} T_4=2 \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \\ , \\ \begin{array}{c} T_5=2 \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \\ , \\ \begin{array}{c} T_6=1 \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \\ , \\ \begin{array}{c} T_7=1 \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \\ , \\ \begin{array}{c} T_8=1 \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{array} \right\}$$

Fig. 3 Collection \mathcal{C} of available discrete covers in the radar database for a covering problem

This paper presents grid cover problems for different radar models respectively to the above parameters, and proves for each whether it is polynomial-solvable or NP-hard. Approximability of the NP-hard problems is also discussed. Section 2 states the generic radar cover problem and presents the different radar models and for each the corresponding grid cover problem. Section 3 describes properties of unidimensional grid covering, and polynomial-time algorithms for its resolution. Section 4 proves the NP-hardness of bidimensional grid covering and discusses its approximability.

2 Problem Statement

2.1 Grid covering

Let $G = \{g_i\}$ be a set of elements, called the *universe set*. Let $\mathcal{C} = \{C_j \subset G\}$ be a collection of subsets in G , a set cover is a sub-collection $\mathcal{S} \subset \mathcal{C}$ whose union covers the universe: $\bigcup_{C \in \mathcal{S}} C = G$.

The decision form of the *set cover problem* asks whether for a given $K \in \mathbb{N}$ there exists a set cover $\mathcal{S} \subset \mathcal{C}$ with cardinality inferior to K , i.e. $|\mathcal{S}| \leq K$. An instance of the set cover problem is described by the system (G, \mathcal{C}, K) . The optimization form, sometimes called *minimum set cover problem*, consists in finding a minimum-size set cover:

$$\begin{array}{ll} \min |\mathcal{S}| \\ \text{s.t. } \forall g_i \in G, \exists C \in \mathcal{S}, g_i \in C \\ \mathcal{S} \subset \mathcal{C} \end{array} \quad (1)$$

If each element $C_j \in \mathcal{C}$ has an associated cost $T_j \in \mathbb{N}$, the problem of finding a set cover with minimal aggregate cost $\sum_{C_j \in \mathcal{S}} T_j$ is called the *weighted set cover problem*. The case with $\forall j, T_j = 1$ is said to be *unweighted*.

From now on we will use a different, radar-based, terminology. The universe set $G = \{g_{m,n}\}$ usually represents a finite bi-dimensional M -by- N regular grid, as displayed in Figure 2, called the *detection grid*, on which:

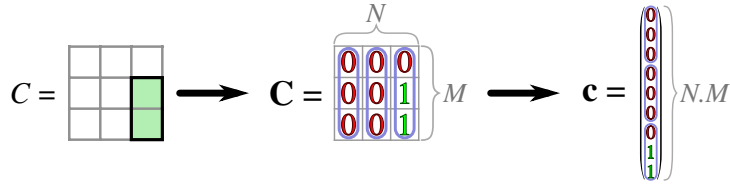


Fig. 4 Dwell discrete cover (left), its binary matrix representation (center) and its binary vector representation (right)

- each element $g_{m,n}$ represents a grid cell indexed by $(m, n) \in [0, M[\times [0, N[\subset \mathbb{N}^2$. The grid contains MN detection cells, each cell corresponding to a certain scanning direction for the radar.
- each node (m, n) represents the intersection of the m -th horizontal line with the n -th vertical line, indexed by $(m, n) \in [0, M] \times [0, N] \subset \mathbb{N}^2$. The grid has $(M + 1)(N + 1)$ nodes.

A subset $C \in \mathcal{C}$ represents the detection area of a radar “emission” (called *dwells* in radar terminology), see Figure 2, and is called a (*dwell*) *discrete cover*. The associated cost T_j of a discrete cover C_j is the time duration required for emitting the radar signal, and receiving then processing the reflected echo. The collection of all available discrete covers is the *radar dwell database*, representing all the discrete covers the radar can emit. A sub-collection of dwell discrete covers, in the radar database, ensuring detection over the entire surveillance, space is called a *radar search pattern*. It corresponds to a set cover of the combinatorial problem. The cost of a radar search pattern is the time required to emit all dwells in sequential order, it is the aggregate cost of its discrete covers.

2.2 Integer program and Matrix formulation

Set cover problems can be written as integer programs by using matrix formulations. Each cover $C \in \mathcal{C}$ can be represented as a binary M -by- N matrix noted \mathbf{C} , or as a binary vector of length MN noted \mathbf{c} , see Figure 4:

$$\mathbf{C}(m, n) = \mathbf{c}(m + Mn) = \begin{cases} 1 & \text{if } g_{m,n} \in C \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

For each cover $C_i \in \mathcal{C}$, let $x_i \in \{0, 1\}$ be the binary selection variable of cover C_i , such that the vector $\mathbf{x} = (x_1, \dots, x_D) \in \{0, 1\}^D$ represents the sub-collection $\mathcal{S} = \{C_i \in \mathcal{C} \text{ s.t. } x_i = 1\}$, containing the chosen covers.

Let $\mathbf{T} = (T_1 \cdots T_D)^T$ be the cost vector and let

$$\mathbf{A} = (\mathbf{c}_1 \cdots \mathbf{c}_D) = \begin{pmatrix} \mathbf{C}_1(0,0) & \cdots & \mathbf{C}_D(0,0) \\ \mathbf{C}_1(1,0) & \cdots & \mathbf{C}_D(1,0) \\ \vdots & \ddots & \vdots \\ \mathbf{C}_1(m,n) & \cdots & \mathbf{C}_D(m,n) \\ \vdots & \vdots & \vdots \end{pmatrix}$$

be the cover matrix.

Then the set cover problem can be written as the following integer program:

$$\begin{aligned} \min \quad & \mathbf{T}^T \cdot \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A} \cdot \mathbf{x} \geq \mathbf{1} \\ & \mathbf{x} \in \{0,1\}^D \end{aligned} \quad (3)$$

where $\mathbf{1}$ is the all-ones vector $(1 \cdots 1)^T$ of length MN .

2.3 Detection grid dimension

Beam-steering is a technique used to control the emission direction of the radar antenna. In electronic antennas, beam-steering is performed by controlling phase differences in the antenna array [16, 17]. The most advanced radar systems can perform bidimensional beam-steering in azimuth and elevation, such radars are said to be *tridimensional*, as they work in with three coordinates: azimuth, elevation and range. There exists radars performing only azimuthal beam-steering, working only in two coordinates: azimuth and range, either because the radar beam covers the entire elevation at once using a cosecant pattern or a fan-beam, or either because the surveillance is very narrow on the horizon. Such radars are said to be *bidimensional*. There are also tridimensional radars stacking multiple beams in elevation, which can be viewed as bidimensional radars from a modelling perspective. See Figure 5 for examples of each radar model, with two possibilities for modelling the detection grid in radar covering problems:

- for bidimensional models, the detection grid has only one dimension. This corresponds to a particular case where $M = 1$ and $N \in \mathbb{N}$.
- for tridimensional models, the detection grid has two dimensions. This is the general case where $(M, N) \in \mathbb{N}^2$.

2.4 Dwell shapes

Radar antennas usually emit single beams, with a connected shape, which are simpler to manipulate from an engineering perspective. It also minimizes energy lost in side-lobes. The definition of a connected subset on grid G is

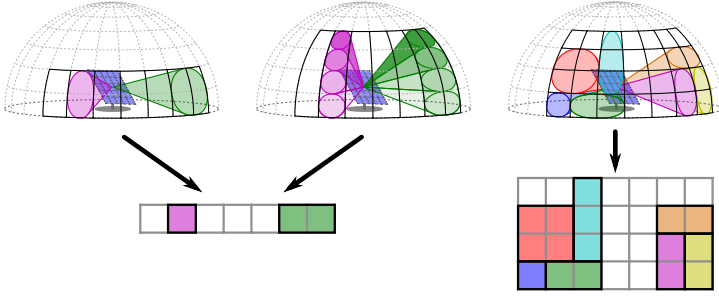


Fig. 5 Bidimensional radar (top-left), tridimensional stacked radar (top-center) and tridimensional radar (top-right), are modelled either as unidimensional covering (bottom-left) or as bidimensional covering (bottom-right)

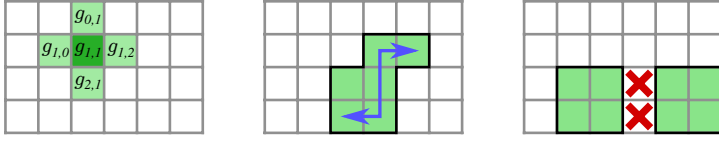


Fig. 6 Set of neighbours $\{g_{0,1}, g_{2,1}, g_{1,0}, g_{1,2}\}$ for a given cell $g_{1,1}$ (left), connected shape (center) and disconnected shape (right)

based on cell neighbourhood, see Figure 6, which contains the four adjacent cells for a given cell $g_{a,b}$:

$$\{g_{a+1,b}, g_{a-1,b}, g_{a,b+1}, g_{a,b-1}\}$$

a subset on the grid is connected if for any two cells in the subset, there is path between them moving from neighbour to neighbour. A subset which is not connected is said to be disconnected, and would theoretically correspond to a radar with multiple emission lobes, see Figure 6.

An interesting case of connected covers are rectangular-shaped covers. In radar engineering, a feasible radar beam-shape is synthesized to fill as closely as possible a desired shape. Rectangular shapes are usually good candidates, easier to synthesize on an antenna phased-array than more complex shapes.

On the grid, a rectangular-shaped cover is a subset of elements included in a rectangle, uniquely defined by its upper left corner node (m_0, n_0) and its lower right corner node (m_1, n_1) , such that $0 \leq m_0 < m_1 \leq M$ and $0 \leq n_0 < n_1 \leq N$. The set representation of a cover defined by corners (m_0, n_0) and (m_1, n_1) is:

$$C = \{g_{m,n}, (m,n) \in [m_0, m_1[\times [n_0, n_1[\}$$

see Figure 3 for an example, C_7 with $(m_0, n_0) = (0, 1)$ and $(m_1, n_1) = (1, 2)$.

The number of possible rectangles on M -by- N grid is

$$\binom{M+1}{2} \binom{N+1}{2} = \frac{MN(M+1)(N+1)}{4} = O(M^2N^2)$$

and gives a broad choice of available discrete covers for computing the pattern, but does not explode exponentially when increasing the grid resolution.

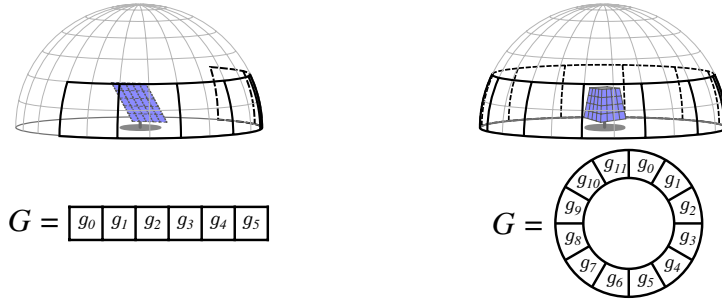


Fig. 7 Radars with limited azimuthal range (left) and with full azimuthal range (left)

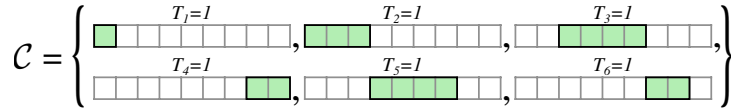


Fig. 8 Available covers for an example of line cover problem

2.5 Azimuthal range and circular grid cover problems

The surveillance space of a fixed-panel radar has a limited azimuthal range. Radar systems can achieve full azimuthal range and scan in all directions by using a rotating-panel or multiple fixed-panels. Limited azimuthal range is modelled by rectangular grids, while full azimuthal range is modelled by circular grids, see Figure 7.

3 Unidimensional grid covering

The set cover problem is NP-hard to solve in general. Certain specific cases, among which unidimensional grid cover problems ($M = 1$), can however be solved in polynomial time. Interestingly, greedy method or linear programming can solve to optimality certain but not all cases, despite the fact that all problems presented in this section can be solved by a polynomial algorithm based on dynamic programming.

3.1 Line cover problem

For a bidimensional radar model with bounded azimuthal range, the associated combinatorial problem is a unidimensional grid cover problem. Its unidimensional grid can be viewed as a line segment on which the discrete cover, being connected sets, represent intervals, see Figure 8.

$$\mathcal{C} = \left\{ \begin{array}{c} T_1=3 \\ \text{ratio}=2/3 \end{array} \right\}, \left\{ \begin{array}{c} T_2=2 \\ \text{ratio}=1/2 \end{array} \right\}, \left\{ \begin{array}{c} T_3=3 \\ \text{ratio}=2/3 \end{array} \right\}, \left\{ \begin{array}{c} T_4=3 \\ \text{ratio}=2/3 \end{array} \right\}$$

Fig. 9 Example for sub-optimality of the greedy method in the unweighed case

3.1.1 Greedy method

For unweighed line covering ($\forall j, T_j = 1$), a straightforward algorithm is the greedy method: among intervals covering the first not-yet-covered detection cell, choose the interval covering the furthest cell, and iterate until the line is covered, see Algorithm 1.

Algorithm 1 Greedy method

```

i ← 0                                ▷ index of first not-covered cell
S ← ∅                                ▷ start with empty solution

while i < N do                    ▷ loop as long as not all cells are covered
  n ← i − 1                            ▷ index of last covered cell
  for  $C_j \in \mathcal{C}$  do
    f, l ← indexes of first and last cell of  $C_j$ 
    if (f ≤ i) ∧ (n ≤ l) then    ▷ Keep cover of the furthest cell
      C ←  $C_j$ 
      n ← l
    end if
  end for
  S ← S ∪ {C}
  i ← n + 1
end while

```

The worst case-complexity of Algorithm 1 is $O(|\mathcal{C}|^2)$. It can be improved to $O(|\mathcal{C}| \log(|\mathcal{C}|))$ by sorting the discrete covers in increasing order of their starting point, and fusing the “while” and “for” loops in a single pass.

The solution is optimal: consider an optimal solution \mathcal{S} of the problem, and $C_a \in \mathcal{S}$ the discrete cover over the first cell, replace C_a by the largest cover C_b which includes the first cell, and solution $(\mathcal{S} \setminus \{C_a\}) \cup \{C_b\}$ remains optimal. Iterating the process on the rest of the original optimal solution turns it into the greedy method solution while keeping the same cost.

The greedy method is however sub-optimal for weighted problems, where the cover added at each iteration should maximize the improvement/cost ratio, i.e. the number of newly covered cells over the discrete cover cost. With weighed costs, a discrete cover with a better improvement/cost ratio (best local choice) can result in sub-optimal solution (bad global choice). See Figure 9, where the greedy method solution $\{C_1, C_3, C_4\}$ has cost 9, whereas the optimal solution $\{C_1, C_2, C_4\}$ has cost 8.

3.1.2 Dynamic programming

In the unweighed case, the greedy method reached optimality by exploiting the problem *optimal substructure*, where an optimal solution can be constructed by combining solutions of sub-problems. *Dynamic programming* generalizes this approach, and is particularly efficient if the problem can be broken down into a polynomial number of sub-problems.

An optimal solution covering the first n cells is built from an optimal solution covering some first $k (< n)$ cells. The n -th sub-problem is “to cover $\{g_i : 0 \leq i < n\}$, i.e. the first n cells”. Iterating the process on n yields a valid solution, see Algorithm 2.

Algorithm 2 Dynamic programming for line cover

```

 $S_0 \leftarrow \emptyset$  ▷ the solution for covering no cells is the empty set

for  $n \in \{1, \dots, N\}$  do ▷ loop on all sub-problems
   $T_{best} \leftarrow +\infty$ 
  for  $C \in \{C \in \mathcal{C} : g_{n-1} \in C\}$  do ▷ loop on all covers containing next cell
     $k \leftarrow$  index of first cell in  $C$ 
     $\mathcal{S} \leftarrow \mathcal{S}_k \cup \{C\}$  ▷ construct candidate solution
     $T_{\mathcal{S}} \leftarrow \sum_{C_j \in \mathcal{S}} T_j$  ▷ compute candidate cost
    if  $(T_{\mathcal{S}} \leq T_{best})$  then
       $\mathcal{S}_n \leftarrow \mathcal{S}$  ▷ keep best valid solution for the  $n$ -th sub-problem
       $T_{best} \leftarrow T_{\mathcal{S}}$ 
    end if
  end for
end for

```

Algorithm 2 requires $O(N|\mathcal{C}|)$ steps and returns an optimal solution: consider an optimal solution \mathcal{S}_n for the n -th sub-problem, then \mathcal{S}_n contains a discrete cover C starting at some cell g_k and including cell g_{n-1} , and $\mathcal{S}_n \setminus \{C\}$ is a valid solution for the k -th sub-problem. Let \mathcal{S}_k be an optimal solution for the k -th sub-problem, then $\mathcal{S}_k \cup \{C\}$ is a valid solution for the n -th sub-problem:

- by optimality of \mathcal{S}_n : $\sum_{C_j \in \mathcal{S}_n} T_j \leq \sum_{C_j \in \mathcal{S}_k \cup \{C\}} T_j$
- by optimality of \mathcal{S}_k : $\sum_{C_j \in \mathcal{S}_n \setminus \{C\}} T_j \geq \sum_{C_j \in \mathcal{S}_k} T_j$

and by combining the two equations

$$\sum_{C_j \in \mathcal{S}_n} T_j = \sum_{C_j \in \mathcal{S}_k \cup \{C\}} T_j$$

so $\mathcal{S}_n \setminus \{C\}$ is an optimal solution for the k -th sub-problem and $\mathcal{S}_k \cup \{C\}$ is an optimal solution for the n -th sub-problem, see Figure 10.

3.1.3 Linear programming

Another approach for solving set cover problems is based on the matrix formulation (3). Integer programming is NP-hard to solve [1]. Replacing integer

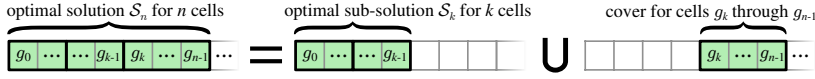


Fig. 10 Line covering optimal substructure of the n -th sub-problem

variables by continuous variables transforms the problem into a linear program

$$\begin{aligned} \min \mathbf{T}^T \cdot \mathbf{x} \\ \text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{1} \\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{1} \end{aligned} \quad (4)$$

which is the *linear relaxation* of (3). A valid solution of an integer program is also a valid solution of its linear relaxation. While the reverse is false in general, there are some cases where integer programming can be reduced to linear programming:

An integer matrix \mathbf{A} is *unimodular* if it is invertible and $\det \mathbf{A} \in \{-1, 1\}$. A direct consequence of Laplace's formula $\mathbf{A}^{-1} = (\det \mathbf{A})^{-1} \text{com } \mathbf{A}^T$, with $\text{com } \mathbf{A}$ the cofactor matrix of \mathbf{A} , is that \mathbf{A}^{-1} is integer if \mathbf{A} is unimodular.

If \mathbf{x} is a vertex (also called *basic solution*) of the polyhedron defined by $\{\mathbf{x} : \mathbf{A} \cdot \mathbf{x} \geq \mathbf{1}\}$, then \mathbf{A} has a square sub-matrix \mathbf{A}_B such that $\mathbf{A}_B \cdot \mathbf{x}_B = \mathbf{1}$ where \mathbf{x}_B are the non-null variables in \mathbf{x} (also called *basic variables*) [18]. If \mathbf{A}_B is unimodular, $\mathbf{x}_B = \mathbf{A}_B^{-1} \cdot \mathbf{1}$ has integral values, and since all other variables are zeroes, \mathbf{x} is also integral and a valid solution of (3).

An integer matrix \mathbf{A} is *totally unimodular* if all its square regular sub-matrices are unimodular, in which case all basic solutions have integral values. In which case, integer programming can be solved in polynomial time by linear programming methods [19].

For line covering, the cover matrix \mathbf{A} has the *consecutive-ones* property, i.e. in a column of \mathbf{A} , all values are zeros or ones, with all ones consecutive. This type of matrix is called *interval matrix* and is totally unimodular [20]. Line covering can be solved in polynomial time by linear programming.

3.2 Circle cover problem

For a bidimensional radar model with full azimuthal range, the detection grid is no longer bounded and represents a circle, see Figure 7. Dynamic programming can still be used to compute an optimal solution in polynomial-time.

The problem still has an optimal substructure. Let the cells be numbered in clockwise order starting from an arbitrary first cell: $G = \{g_0, \dots, g_{N-1}\}$ with cell g_{N-1} and g_0 being neighbours, see Figure 7. The (n, w) -th sub-problem is "to cover $\{g_k : k = n + i \bmod N, 0 \leq i < w\}$, i.e. the w cells in clockwise order starting by g_n ". A sub-problem can be described by its (starting) index $n \in \{0, \dots, N-1\}$ and its width $w \in \{1, \dots, N\}$. The substructure of circle covering can be viewed as splitting the problem into all possible arc segments.

Algorithm 3 requires $O(N^2|\mathcal{C}|)$ steps and returns an optimal solution: consider an optimal solution $\mathcal{S}_{n,w}$ for the (n, w) -th sub-problem with $w \geq 1$, then

Algorithm 3 Dynamic programming for circle cover

```

for  $n \in \{0, \dots, N-1\}$  do
     $\mathcal{S}_{n,0} \leftarrow \emptyset$  ▷ the solution for covering no cells is the empty set
end for

for  $n \in \{0, \dots, N-1\}$  do ▷ loop on all sub-problems
    for  $w \in \{1, \dots, N\}$  do
         $T_{best} \leftarrow +\infty$ 
         $l \leftarrow n + w - 1 \pmod N$  ▷ compute index of the next cell to cover
        for  $C \in \{C \in \mathcal{C} : g_l \in C\}$  do ▷ loop on all covers containing next cell
             $k \leftarrow \text{index of clockwise-leftmost cell in } C$ 

            if  $k - n \pmod N \leq l - n \pmod N$  then ▷ check if “ $n \leq k \leq l$ ” clockwise
                 $s \leftarrow k - n \pmod N$  ▷ complementary sub-pattern width
                 $\mathcal{S} \leftarrow \mathcal{S}_{n,s} \cup \{C\}$  ▷ construct candidate solution
            else ▷ otherwise “ $k < n \leq l$ ” clockwise
                 $\mathcal{S} \leftarrow \{C\}$  ▷  $C$  suffices to solve current problem
            end if

             $T_{\mathcal{S}} \leftarrow \sum_{C_j \in \mathcal{S}} T_j$  ▷ compute candidate cost
            if  $T_{\mathcal{S}} \leq T_{best}$  then
                 $\mathcal{S}_{n,w} \leftarrow \mathcal{S}$  ▷ keep best valid solution for  $(n, w)$ -th sub-problem
                 $T_{best} \leftarrow T_{\mathcal{S}}$ 
            end if
        end for
    end for
end for

```

$\mathcal{S}_{n,w}$ contains a discrete cover C starting (clockwise) at cell g_k and including cell g_l with $l = n + w - 1 \pmod N$. There are two possible situations:

- “ $k < n \leq l$ ” clockwise:
 $\{C\}$ suffices to cover the cells $\{g_n, \dots, g_l\}$ and is an optimal solution of the (n, w) -th sub-problem: $\mathcal{S}_{n,w} = \{C\}$.
- “ $n \leq k \leq l$ ” clockwise:
 Let $s = k - n \pmod N$, then $\mathcal{S}_{n,w} \setminus \{C\}$ is a valid solution for the (n, s) -th sub-problem. Let $\mathcal{S}_{n,s}$ be an optimal solution for the (n, s) -th sub-problem, then $\mathcal{S}_{n,s} \cup \{C\}$ is a valid solution for the (n, w) -th sub-problem:
 - by optimality of $\mathcal{S}_{n,w}$: $\sum_{C_j \in \mathcal{S}_{n,w}} T_j \leq \sum_{C_j \in \mathcal{S}_{n,s} \cup \{C\}} T_j$
 - by optimality of $\mathcal{S}_{n,s}$: $\sum_{C_j \in \mathcal{S}_{n,w} \setminus \{C\}} T_j \geq \sum_{C_j \in \mathcal{S}_{n,s}} T_j$
 and by combining the two equations

$$\sum_{C_j \in \mathcal{S}_{n,w} \setminus \{C\}} T_j = \sum_{C_j \in \mathcal{S}_{n,s}} T_j$$

so $\mathcal{S}_{n,w} \setminus \{C\}$ is an optimal solution for the (n, s) -th sub-problem.

Any optimal solution for a given sub-problem is either a unique cover, or the union of a smaller sub-problem optimal solution and a cover, see Figure 11.

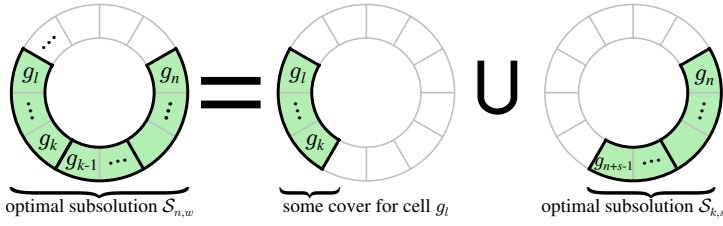


Fig. 11 Circle covering optimal substructure of the (n, w) -th sub-problem

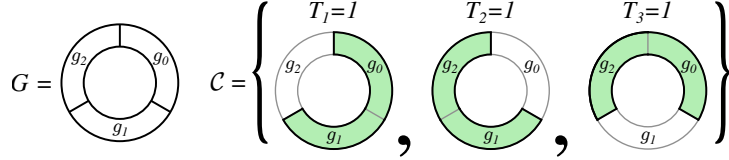


Fig. 12 Example of non-integral circle cover problem

3.2.1 Integrality gap

Linear programming, however, cannot be used to solve circle covering, because the cover matrix \mathbf{A} can be non-unimodular. The simplest problem instance for which this situation appears is displayed in Figure 12.

The relaxed linear program has the cover matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

with $\det(\mathbf{A}) = 2$ and yields the unique fractional optimal solution $\mathbf{x}_L = (\frac{1}{2} \ \frac{1}{2} \ \frac{1}{2})^T$, which combines a weighing of all three covers to produce the optimal fractional solution, and is strictly better than an integral optimal solution, say $\mathbf{x}_I = (1 \ 1 \ 0)^T$. The difference of cost between both solutions is called the *integrality gap*, here $\mathbf{T}^T \cdot (\mathbf{x}_I - \mathbf{x}_L) = \frac{1}{2}$.

3.3 Logarithmic encoding

All previous problems can be solved in polynomial time using dynamic programming. However, their computational complexity is polynomial in N , the “grid size”. If the problem input is given in matrix formulation, i.e. \mathbf{c} and \mathbf{A} , then the encoding size of the input is $|\mathcal{C}|N$, and the algorithm is truly polynomial.

But for interval covers, this encoding scheme is obviously suboptimal, since an interval can be described using only two integers, its starting index a and its ending index b , see Figure 13. The number of bits required to encode indices in $\{0, \dots, N-1\}$ is $p = \lceil \log_2(N) \rceil$, and the encoding size of a compressed input is $|\mathcal{C}|2p$. For this input size, Algorithm 2 complexity is $O(|\mathcal{C}|2^p)$ and Algorithm

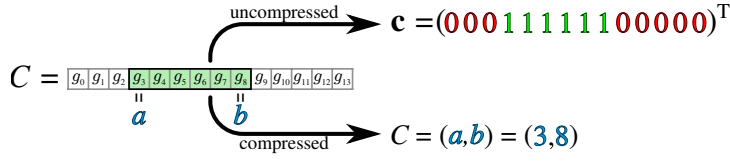


Fig. 13 Uncompressed boolean vector (top) and compressed logarithmic encoding (bottom)

3 complexity is $O(|\mathcal{C}|4^p)$. While those algorithms are polynomial in the size and the values of the input, they are exponential in the number of bits used to encode those values. Such algorithms are said to be *pseudo-polynomial*.

Problems with pseudo-polynomial algorithms can be NP-complete when considering the logarithmic cost, i.e. the computational cost on a machine using bits to encode values. Such problems are said to be *weakly NP-complete*, an example of a weakly NP-complete problem is the knapsack problem, which also possesses a dynamic programming pseudo-polynomial algorithm [21].

3.3.1 Input reduction

For unidimensional grid cover problems, true polynomial optimization can be achieved by removing redundant cells from the input before dynamic programming, see Algorithm 4 and Figure 14..

Algorithm 4 Input reduction

Input : $\mathcal{C} = \{(a_j, b_j)\}_{1 \leq j \leq D}$ with a_j, b_j integers encoded with p bits

$G' \leftarrow \bigcup_{(a,b) \in \mathcal{C}} \{a, b+1\}$

Sort G' and remove duplicates

$\mathcal{C}' \leftarrow \emptyset$

for $(a, b) \in \mathcal{C}$ **do**

$a' \leftarrow$ index of a in G'

$b' \leftarrow$ (index of $b+1$ in G') $- 1$

$\mathcal{C}' \leftarrow \mathcal{C}' \cup (a', b')$

end for

Output : $\mathcal{C}' = \{(a'_j, b'_j)\}_{1 \leq j \leq D}$ with a'_j, b'_j integers encoded with p bits

Input reduction only keeps cells which corresponds to a change in the problem instance structure respectfully to the previous cell, preserving the instance structure. The computational cost of input reduction is detailed in Table 1. The reduced grid G' contains at most $2|\mathcal{C}|$ elements. The indices of the reduced covers \mathcal{C}' are obtained by dichotomic search in G' . Each new index is encoded using $\lceil \log_2(2|\mathcal{C}|) \rceil = O(\log |\mathcal{C}|)$ bits. \mathcal{C}' is encoded using $2|\mathcal{C}| \cdot O(\log |\mathcal{C}|) = O(|\mathcal{C}| \log |\mathcal{C}|)$ bits.

The logarithmic cost of dynamic programming is the product of its arithmetic cost, $O(N|\mathcal{C}|)$ or $O(N^2|\mathcal{C}|)$, and the logarithmic cost of an arithmetic

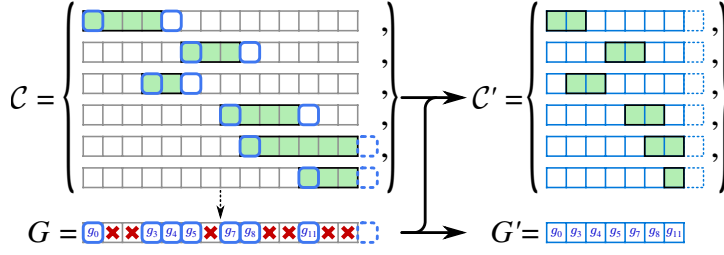


Fig. 14 Input reduction of the original problem instance (top-left) into a reduced instance (right) by removing redundant detection cells, from the grid (bottom-left)

Instruction	arithmetic cost	logarithmic cost
Sort G'	$O(\mathcal{C} \log \mathcal{C})$	$O(p \mathcal{C} \log \mathcal{C})$
Search indices a', b'	$2 \mathcal{C} \cdot O(\log \mathcal{C})$	$2 \mathcal{C} \cdot O(p \log \mathcal{C})$
Total	$O(\mathcal{C} \log \mathcal{C})$	$O(p \mathcal{C} \log \mathcal{C})$

Table 1 Computational cost of input reduction

Problem	Line covering	Circle covering
Input reduction	$O(p \mathcal{C} \log \mathcal{C})$	
Dynamic programming	$O(N \mathcal{C}) \cdot O(\log N)$ $= O(\mathcal{C} ^2 \log \mathcal{C})$	$O(N^2 \mathcal{C}) \cdot O(\log N)$ $= O(\mathcal{C} ^3 \log \mathcal{C})$
Total	$O((\mathcal{C} + p) \mathcal{C} \log \mathcal{C})$	$O((\mathcal{C} ^2 + p) \mathcal{C} \log \mathcal{C})$

Table 2 Logarithmic cost of dynamic programming and input reduction

Problem	Line covering	Circle covering
Input reduction	$O(\mathcal{C} \log \mathcal{C})$	
Dynamic programming	$O(N \mathcal{C}) = O(\mathcal{C} ^2)$	$O(N^2 \mathcal{C}) = O(\mathcal{C} ^3)$
Total	$O(\mathcal{C} ^2)$	$O(\mathcal{C} ^3)$

Table 3 Arithmetic cost of dynamic programming and input reduction

operation: $O(\log N)$. The reduced grid size is $N = |G'| \leq 2|\mathcal{C}| = O(|\mathcal{C}|)$. The overall cost of both input reduction and dynamic programming is in Table 2. Both line covering and circle covering can be solved in true polynomial time.

Those algorithms are actually *strongly polynomial*, meaning their computational cost when considering arithmetic operations, regardless of encoding size, is polynomial in the number of input values, see Table 3.

4 Bidimensional grid covering

Some grid cover problems remains NP-hard to solve, notably bidimensional grid cover problems. This means that tridimensional radar models produce NP-hard optimization problems.

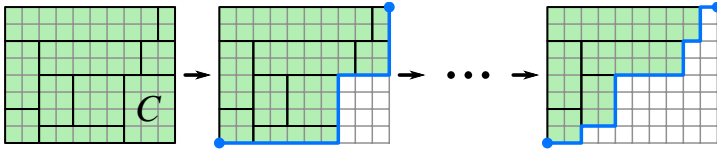


Fig. 15 Substructure decomposition of the rectangular grid cover problem

4.1 Rectangular grid cover problem

Radar phase feeds used to modify the emitted radiation pattern are controlled through numerical phasers. Their precision is often sufficient for the phase value to be modelled as continuous variable. The space of all feasible radiation patterns is tremendously big in theory, and considering all possibilities is unrealistic, so approximations must be made.

In practice, it is reasonable to optimize radar search patterns by considering only rectangular-shaped covers [13], which reduces the problem to *rectangular grid covering*, see Figure 3 for an instance example.

4.1.1 Dynamic programming approach

Considering the algorithms presented in Section 3, a natural attempt to solve rectangular grid covering problem would be to generalize the dynamic programming approach used on unidimensional grids to bidimensional grids.

Consider an optimal solution for the rectangular grid cover problem. It is combination of a rectangular cover C over the bottom-right corner and an optimal sub-solution covering the remaining “top-left” cells. By iterating the decomposition process, the grid cover sub-problems are to “cover the top-left part of G ”, see Figure 15. The number of sub-problems is equal to the number of way of cutting G in two, or equivalently to the number of paths between the top-right corner cell and the bottom-left corner cell of G . A cut is made by $N + M$ edges, with M vertical and N horizontal. The number of possible cuts is the choice of which cuts are vertical (or horizontal): $\binom{N+M}{N} = \binom{N+M}{M}$, and grows exponentially with $K = \min\{N, M\}$ according to Stirling’s formula

$$\binom{N+M}{N} \geq \binom{2K}{K} \simeq \frac{2^{2K}}{\sqrt{\pi K}}$$

making dynamic programming inefficient for bidimensional grid covering.

4.1.2 NP-hardness

All presented cover problems can be reduced to set covering (1), or to integer programming (3), and are thus in NP. Bidimensional covering is furthermore a NP-hard problem, to which any NP problem can be reduced. The following section describe a polynomial Karp reduction from vertex covering, a known NP-hard problem [1], to rectangular grid covering.

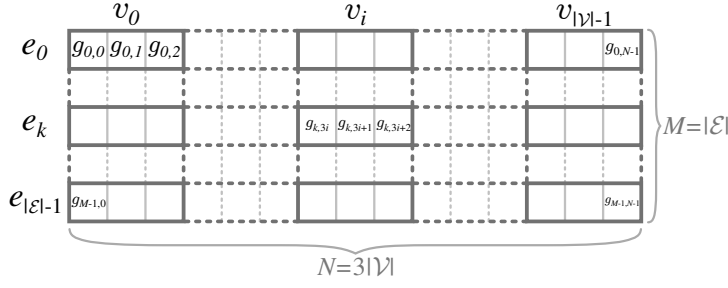


Fig. 16 Reduction grid of vertex covering into rectangular grid covering

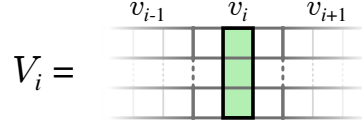


Fig. 17 Column cover

The vertex cover problem is defined as follow: let $(\mathcal{V}, \mathcal{E})$ a graph, let $K \in \mathbb{N}$. Is there a subset $\mathcal{U} \subset \mathcal{V}$ with cardinal $|\mathcal{U}| \leq K$ such that $\forall v \in \mathcal{V}, \exists v' \in \mathcal{U}$ with $(v, v') \in \mathcal{E}$? An instance of the problem is defined by the system $(\mathcal{V}, \mathcal{E}, K)$.

The decision form of rectangular grid covering is defined as: let G be a M -by- N rectangular grid, let $\mathcal{C} = \{C_1, \dots, C_D\}$ be a collection of rectangular covers, and let $F \in \mathbb{N}$. Is there a valid pattern $\mathcal{S} \subset \mathcal{C}$ covering the grid, with $|\mathcal{S}| \leq F$? An instance of the decision problem is defined by the system (G, \mathcal{C}, F) and can be encoded by a NMD boolean array using the matrix formulation in (3).

The reduction transforms a vertex cover problem instance $(\mathcal{V}, \mathcal{E}, K)$ into a decision grid cover problem instance (G, \mathcal{C}, F) . Let the graph vertices and edges be ordered as $\mathcal{V} = \{v_0, \dots, v_{|\mathcal{V}|-1}\}$ and $\mathcal{E} = \{e_0, \dots, e_{|\mathcal{E}|-1}\}$. Each edge connects a pair of distinct vertices $e_m = (v_i, v_j)$ with $i < j$.

Let G be a $|\mathcal{E}|$ -by- $3|\mathcal{V}|$ rectangular grid. Each row represents an edge, and each block of three columns represents a vertex, see Figure 16. Three types of rectangular covers are defined on the grid:

- **Column covers:** for each vertex v_i , the central column of the block column

$$V_i = \{g_{m, 3i+1} : 0 \leq m < M\}$$

see Figure 17. The set of column covers is $\mathcal{C}_{\mathcal{V}} = \{V_i : v_i \in \mathcal{V}\}$.

- **Side-row covers:** for each edge $e_k = (v_i, v_j)$, two side-row covers in the k -th row, one on left to V_i and the other right to V_j , see Figure 18

$$L_k = \{g_{k,n} : 0 \leq n \leq 3i\} \text{ and } R_k = \{g_{k,n} : 3j+2 \leq n < N\}$$

The set of side-row covers is $\mathcal{C}_S = \{L_k : e_k \in \mathcal{E}\} \cup \{R_k : e_k \in \mathcal{E}\}$. Side-row covers are all required in a valid pattern, for either the first or last cell of

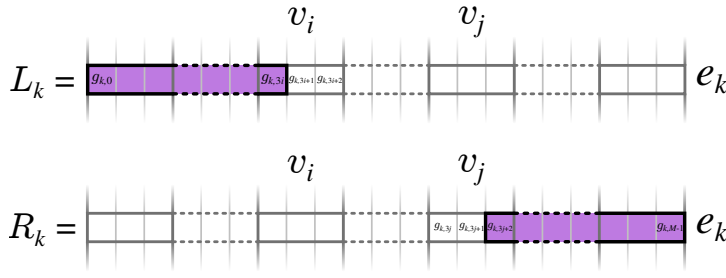


Fig. 18 Side-row covers

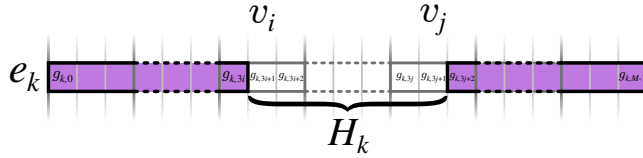
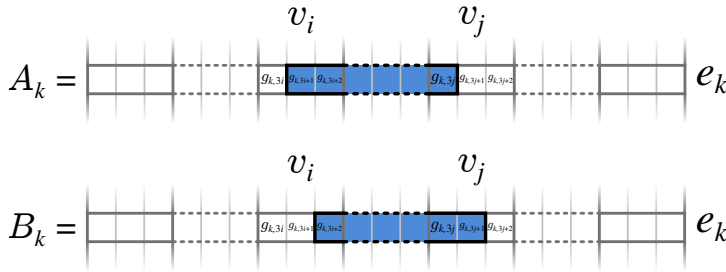
Fig. 19 The “central part” H_k 

Fig. 20 Center-row covers

each row, and only the “central part” $H_k = \{g_{k,n} : 3i + 1 \leq n \leq 3j + 1\}$ remains to be covered, see Figure 19.

- **Center-row covers:** for each edge $e_k = (v_i, v_j)$, the row “central part” H_k can be partially covered by a column cover, and the remaining uncovered cells can be covered by one of the two center-row covers, see Figure 20

$$A_k = \{g_{k,n} : 3i + 1 \leq n \leq 3j\} \text{ or } B_k = \{g_{k,n} : 3i + 2 \leq n \leq 3j + 1\}$$

whether the column cover is V_j or V_i . Either combination can cover the row “central part” since $H_k \subset A_k \cup V_j$ and $H_k \subset V_i \cup B_k$. Note that H_k can also be covered by $A_k \cup B_k$. Covering H_k requires two covers in any case, with the three possible configurations shown in Figure 21. The set of all center-row covers is $\mathcal{C}_H = \{A_k : e_k \in \mathcal{E}\} \cup \{B_k : e_k \in \mathcal{E}\}$. For each row at least one of the two possible center-row covers is required for cells $\{g_{k,n} : 3i + 2 \leq n \leq 3j\}$ between columns covers V_i and V_j .

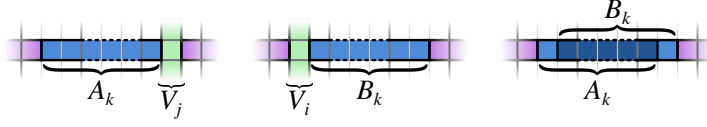


Fig. 21 The three configurations for covering H_k

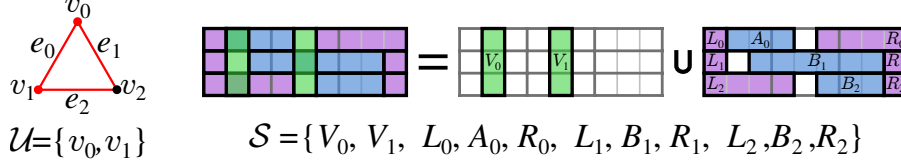


Fig. 22 Vertex cover (left) and reduced grid cover solution (right) for complete graph K_3

Let the set of all rectangular covers be $\mathcal{C} = \mathcal{C}_{\mathcal{V}} \cup \mathcal{C}_{\mathcal{S}} \cup \mathcal{C}_H$, the grid cover instance $(G, \mathcal{C}, 3|\mathcal{E}| + K)$ has a solution if and only if the vertex cover instance $(\mathcal{V}, \mathcal{E}, K)$ has a solution, see Figure 22 for an example with its solutions.

Suppose there is a valid solution \mathcal{U} with $|\mathcal{U}| \leq K$ for problem $(\mathcal{V}, \mathcal{E}, K)$. Let $\mathcal{S}_{\mathcal{U}} = \{V_i : v_i \in \mathcal{U}\} \subset \mathcal{C}_{\mathcal{V}}$, the set of column covers corresponding to the vertices in vertex cover \mathcal{U} . For each edge $e_k = (v_i, v_j)$, either $V_i \in \mathcal{S}_{\mathcal{U}}$ or $V_j \in \mathcal{S}_{\mathcal{U}}$. Let E_k be the center-row cover complementing the “central part”:

$$E_k = \begin{cases} A_k & \text{if } V_j \in \mathcal{U} \\ B_k & \text{otherwise} \end{cases}$$

and the corresponding row is covered by $L_k \cup E_k \cup V_i \cup R_k$ for some $V_i \in \mathcal{S}_{\mathcal{U}}$. So $\mathcal{S} = \mathcal{S}_{\mathcal{U}} \cup \{E_k : e_k \in \mathcal{E}\} \cup \mathcal{C}_{\mathcal{S}}$ is a valid pattern containing $K + |\mathcal{E}| + 2|\mathcal{E}|$ elements and thus a solution for the grid cover instance $(G, \mathcal{C}, 3|\mathcal{E}| + K)$.

Conversely, suppose there is a valid solution \mathcal{S} with $|\mathcal{S}| \leq 3|\mathcal{E}| + K$ for the grid cover instance $(G, \mathcal{C}, 3|\mathcal{E}| + K)$. For each edge $e_k = (v_i, v_j)$, the corresponding row contains the two side-row covers and at least one center-row cover. If said row has the two center-row covers, one can be replaced by a column cover: $\mathcal{S} \leftarrow \mathcal{S} \cup \{V_i\} \setminus \{B_k\}$ without changing the solution cardinality: $|\mathcal{S}| \leq |\mathcal{E}| + 3K$. Iterating this process for each row yields a pattern with one center-row cover per row, with the “central part” H_k covered by either $A_k \cup V_j$ or $B_k \cup V_i$. So for each edge $e_k = (v_i, v_j)$, \mathcal{S} contains either V_i or V_j , and $\mathcal{U} = \{v_i : V_i \in \mathcal{S} \cap \mathcal{C}_{\mathcal{V}}\}$ is a valid vertex cover. Furthermore

$$|\mathcal{U}| = |\mathcal{S} \cap \mathcal{C}_{\mathcal{V}}| = |\mathcal{S} \setminus \mathcal{C}_H \setminus \mathcal{C}_{\mathcal{S}}| = |\mathcal{S}| - |\mathcal{E}| - 2|\mathcal{E}| = |\mathcal{S}| - 3|\mathcal{E}| \leq K$$

since \mathcal{S} contains one cover from \mathcal{C}_H per row and all covers in $\mathcal{C}_{\mathcal{S}}$. \mathcal{U} is thus a valid solution for vertex cover instance $(\mathcal{V}, \mathcal{E}, K)$.

So there is a solution for the vertex cover instance $(\mathcal{V}, \mathcal{E}, K)$ if and only if there is a solution for grid cover instance $(G, \mathcal{C}, 3|\mathcal{E}| + K)$. Only left to

check is that the reduction is polynomial: suppose the grid cover instance is encoded using binary matrices. Each cover has $3|\mathcal{V}||\mathcal{E}|$ boolean elements, and \mathcal{C} contains $|\mathcal{V}| + 4|\mathcal{E}|$ covers. Generating all covers takes $O(3|\mathcal{V}|^2|\mathcal{E}| + 12|\mathcal{V}||\mathcal{E}|^2)$ operations and the problem instance is encoded using $3|\mathcal{V}|^2|\mathcal{E}| + 12|\mathcal{V}||\mathcal{E}|^2$ bits. So the reduction is polynomial and rectangular grid covering is NP-hard.

4.1.3 Approximability

There is currently not known methods for solving efficiently NP-hard problems and might never be. An important field in optimization is the design of polynomial approximation algorithms, whose solution is guaranteed to be within a given ratio of the optimal cost F_{opt} . For a minimization problem, an α -approximation algorithm guarantees a solution with cost $F_{apx} \leq \alpha F_{opt}$.

Set covering is generally not easy to approximate, being log-APX-complete [3,6]. In some cases, better approximations can be achieved. If the problem frequency parameter $f = \max_i |\{C \in \mathcal{C} : g_i \in C\}|$, i.e. the maximum number of covers sharing a common element, is bounded then f -approximation is possible by a primal-dual algorithm [22]. But the frequency parameter is generally not bound for grid covering.

While vertex covering is APX-complete [23], the previous reduction of vertex covering to rectangular grid covering is not approximation-preserving, and cannot be used to infer the approximation complexity of grid covering:

Consider a graph $(\mathcal{V}, \mathcal{E})$, for which a minimum vertex cover has optimal cardinal K_{opt} . The grid cover instance (G, \mathcal{C}) obtained via the previous reduction has an optimal solution with cost $F_{opt} = 3|\mathcal{E}| + K_{opt}$.

Consider an α -approximation algorithm for grid covering, which returns a solution with cost $F_{apx} \leq \alpha F_{opt} = \alpha(3|\mathcal{E}| + K_{opt})$. From this solution, a vertex cover for $(\mathcal{V}, \mathcal{E})$ can be computed by replacing and removing center-row and side-row covers, as has been done in 4.1.2. The vertex cover has a cost

$$K_{apx} = F_{apx} - 3|\mathcal{E}| \leq \alpha(3|\mathcal{E}| + K_{opt}) - 3|\mathcal{E}| = \alpha K_{opt} + (\alpha - 1)3|\mathcal{E}|$$

which can be arbitrarily high as a graph with size-bounded optimal vertex cover can have an arbitrarily high number of edges, like the star graph S_n . There is no ratio β such that $(\alpha - 1)3|\mathcal{E}| \leq \beta K_{opt}$, and the reduction is not approximation-preserving. The exact approximability of rectangular grid covering remains an open question, though the problem is at worst log-approximable.

4.2 Connected grid cover problem

The radar model with connected dwell shapes is a generalization of the rectangular case: the set of available dwell covers can contain any connected shape, as defined in 2.4. Since rectangular shapes are valid connected shapes, the problem is immediately NP-hard, since any problem instance of rectangular grid

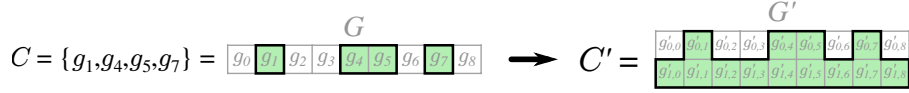


Fig. 23 Reduction from general set covering to connected grid covering

covering is a valid instance of connected grid covering. An alternate reduction from general set covering is also possible.

Let (G, \mathcal{C}, K) be an instance of the set cover problem. Let G' be a 2-by- $|G|$ grid. For each cover $C \in \mathcal{C}$, let

$$C' = \{g'_{0,i} \in G' : g_i \in C\} \cup \{g_{1,0}, \dots, g_{1,|G|}\}$$

such that the first line of cover C' replicates C , while the second line of C' contains all elements on the second line of G' , see Figure 23, ensuring that C' is a connected set. Let $\mathcal{C}' = \{C' : C \in \mathcal{C}\}$.

Suppose $\mathcal{S} \subset \mathcal{C}$ is a solution for set cover instance (G, \mathcal{C}, K) and let $\mathcal{S}' = \{C' : C \in \mathcal{S}\}$.

\mathcal{S} is a solution for (G, \mathcal{C}, K) if and only if \mathcal{S}' is a solution for (G', \mathcal{C}', K) and the two problems are computationally equivalent. This reduction keeps the same cost function for both problems, and is stronger than for the previous reduction of vertex covering to rectangular grid covering, as it preserves approximation properties. Thus connected grid covering is NP-hard, and also log-APX-complete, like general set covering [6].

5 Conclusion

Combinatorial covering is a natural and powerful framework for optimizing modern radar search patterns. While the increase of computational power in recent decades allows the processing of more complex optimization problems, the improved flexibility and precision of modern radars systems have also increased the radar models complexity. In modern warfare, ensuring robust detection under tight time constraints in operational situations is essential. This requires an accurate knowledge of radar covering problems theoretical complexity.

Cover problems presented in this paper each correspond to different radar models. Unidimensional radar covering can be solved in strongly polynomial time, while bidimensional radar covering is NP-hard and log-APX-complete in most cases. An open question remains on the exact approximability of rectangular grid covering, being at worst log-APX. While branch-and-bound usually performs well in most practical cases, having guaranteed worst-case approximated solutions is highly desirable, especially for such sensitive systems used in defence applications.

Acknowledgements This work is partly supported by a DGA-MRIS scholarship.

References

1. Karp, R.M.: Reducibility among combinatorial problems. In: Symposium on the Complexity of Computer Computations, pp. 85–103. Springer US, Boston, MA (1972). DOI 10.1007/978-1-4684-2001-2_9. URL https://doi.org/10.1007/978-1-4684-2001-2_9
2. Johnson, D.S.: Approximation algorithms for combinatorial problems. In: Proceedings of the 5th Annual ACM Symposium on Theory of Computing, STOC, pp. 38–49. ACM, New York, NY, USA (1973). DOI 10.1145/800125.804034. URL <http://doi.acm.org/10.1145/800125.804034>
3. Chvatal, V.: A greedy heuristic for the set-covering problem. *Mathematics of Operations Research* **4**(3), 233–235 (1979). URL <http://www.jstor.org/stable/3689577>
4. Lund, C., Yannakakis, M.: On the hardness of approximating minimization problems. *J. ACM* **41**(5), 960–981 (1994). DOI 10.1145/185675.306789. URL <http://doi.acm.org/10.1145/185675.306789>
5. Raz, R., Safra, S.: A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing, STOC, pp. 475–484. ACM, New York, NY, USA (1997). DOI 10.1145/258533.258641. URL <http://doi.acm.org/10.1145/258533.258641>
6. Escoffier, B., Paschos, V.T.: Completeness in approximation classes beyond apx. *Theoretical Computer Science* **359**(1), 369 – 377 (2006). DOI 10.1016/j.tcs.2006.05.023. URL <http://www.sciencedirect.com/science/article/pii/S0304397506003148>
7. Srinivasan, A.: Improved approximations of packing and covering problems. In: Proceedings of the 27th Annual ACM Symposium on Theory of Computing, STOC, pp. 268–276. ACM, New York, NY, USA (1995). DOI 10.1145/225058.225138. URL <http://doi.acm.org/10.1145/225058.225138>
8. Koch, T., et al.: Miplib 2010. *Mathematical Programming Computation* **3**(2), 103 (2011). DOI 10.1007/s12532-011-0025-9. URL <https://doi.org/10.1007/s12532-011-0025-9>
9. Pataki, G., Tural, M., Wong, E.B.: Basis reduction and the complexity of branch-and-bound. In: Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, pp. 1254–1261. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2010). URL <http://dl.acm.org/citation.cfm?id=1873601.1873701>
10. Li, J., Jin, Y.: A ptas for the weighted unit disk cover problem. In: Automata, Languages, and Programming: 42nd International Colloquium, ICALP, pp. 898–909. Springer Berlin Heidelberg (2015). DOI 10.1007/978-3-662-47672-7_73. URL https://doi.org/10.1007/978-3-662-47672-7_73
11. Chan, T.M., Grant, E.: Exact algorithms and apx-hardness results for geometric packing and covering problems. *Computational Geometry* **47**(2), 112 – 124 (2014). DOI 10.1016/j.comgeo.2012.04.001. URL <http://www.sciencedirect.com/science/article/pii/S0925772112000740>. Special Issue: 23rd Canadian Conference on Computational Geometry (CCCG11)
12. Schöbel, A.: Optimization in Public Transportation: Stop Location, Delay Management and Tariff Zone Design in a Public Transportation Network. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
13. Briheche, Y., Barbaresco, F., Bennis, F., Chablat, D., Gosselin, F.: Non-uniform constrained optimization of radar search patterns in direction cosines space using integer programming. In: 2016 17th International Radar Symposium (IRS) (2016). DOI 10.1109/IRS.2016.7497343
14. Barbaresco, F., Deltour, J., Desodt, G., Durand, B., Guenais, T., Labreuche, C.: Intelligent m3r radar time resources management: Advanced cognition, agility & autonomy capabilities. In: Radar Conference - Surveillance for a Safer World, RADAR. International (2009)
15. National Research Council: Evaluation of the Multifunction Phased Array Radar Planning Process. The National Academies Press, Washington, DC (2008). DOI 10.17226/12438
16. Stutzman, W., Thiele, G.: Antenna Theory and Design. Wiley (2012)

17. Orfanidis, S.J.: *Electromagnetic Waves and Antennas*. Rutgers University (2016). DOI 10.1016/B978-075064947-6/50011-3. URL <http://www.ece.rutgers.edu/~orfanidi/ewa/>
18. Matouek, J., Gärtner, B.: *Understanding and Using Linear Programming* (Universitext). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
19. Megiddo, N.: On finding primal- and dual-optimal bases. *ORSA Journal on Computing* **3**(1), 63–65 (1991). DOI 10.1287/ijoc.3.1.63. URL <https://doi.org/10.1287/ijoc.3.1.63>
20. Nemhauser, G.L., Wolsey, L.A.: *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY, USA (1988)
21. Vazirani, V.V.: *Approximation Algorithms*. Springer-Verlag New York, Inc. (2001). DOI 10.1002/rsa.10038
22. Williamson, D.P., Shmoys, D.B.: *The Design of Approximation Algorithms*, 1st edn. Cambridge University Press, New York, NY, USA (2011)
23. Ausiello, G., Protasi, M., Marchetti-Spaccamela, A., Gambosi, G., Crescenzi, P., Kann, V.: *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*, 1st edn. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1999)