# Exploring the Optimal Camera Placement Problem and its Relationship with the Set Covering Problem

Malek Almousa, Matthias Ehrgott, and Ahmed Kheiri

Department of Management Science, Lancaster University
Lancaster LA1 4YX, United Kingdom
{m.almousa,m.ehrgott,a.kheiri}@lancaster.ac.uk

**Abstract.** Optimal Camera Placement (OCP) is the process of finding a subset of cameras that either maximises the coverage, such that the cost of cameras is reduced, or minimises the total cost of cameras, such that coverage constraints are satisfied. By adopting the latter formulation, the OCP problem can be formulated as a Set Covering Problem (SCP), as the concepts of the two problems are inherently similar. Until recently, the literature has not explicitly discussed this similarity. Hence, this paper examines the OCP problem by leveraging the formulation established in prior research. Our focus lies in the practical application, as we implement the model on all instances to derive meaningful insights. Furthermore, we explore techniques from the SCP literature that can be applied to address the OCP problem in future studies. In this study, we address 69 problem instances, utilising a benchmark set generated by other researchers. These instances were employed as part of the GECCO 2021 competition on the optimal camera placement problem and the unicost set covering problem. We provide detailed results, and we conclude with recommendations for future research.

**Keywords:** Combinatorial Optimisation, Set Covering Problem, Optimal Camera Placement

## 1 Introduction

In recent years, establishing an optimal camera network for surveillance purposes has been the subject of interest in several studies. This increased attention is prompted by the worldwide spread of surveillance systems, which are being employed to address various issues, including analysing crowd movements, monitoring transportation systems, or simply observing certain places for general purposes [13]. The idea of camera placement was first discussed in computational geometry in the 1970s by Chvátal [6]. The author's widely known Art Gallery Problem (AGP) has inspired numerous camera placement studies since its introduction. AGP is an approach for placing guards in an art gallery, where the goal is to minimise the number of guards, ensuring that every point in the art gallery is covered by at least one guard. Transforming this idea to camera

placement, guards become cameras, and the general goal is to select the smallest subset of cameras that achieves full coverage.

Generally and more formally, *Optimal Camera Placement* (OCP) is the process of finding a subset of cameras that either maximises the coverage, such that the cost (or number) of cameras is reduced; or minimises the total cost (or number) of cameras, such that coverage constraints are satisfied. When working with the minimisation objective, the problem can be viewed as a *Set Covering Problem* (SCP). However, the similarity between the two problems has not been explicitly discussed in the OCP literature until recently [13]. Therefore, the aim of our study is to explore the connection between the two problems and discuss techniques from the SCP literature that can be applied to the OCP problem.

The rest of the paper is structured as follows: Section 2 provides a brief summary of the OCP literature, discussing different techniques that have been used to deal with the problem. This is followed by a description of SCP in Section 3. Section 4 provides a detailed problem description and formulations of our OCP problem. Then, a summary of the results of the problem is given in Section 5. Finally, Section 6 concludes our study and gives a hint of what can be done in future work.

## 2 OCP Literature

Inspired by the AGP, the use of optimal camera networks for surveillance has increased in the past few decades in order to fully monitor different areas, including public places, warehouses, buildings, and so on. The general goal is to maximise coverage or to minimise the cost (or number) of cameras, given a set of constraints [14]. When it comes to solving OCP, researchers have employed various tools. Some used exact methods to deal with the problem and find optimal solutions, while others used heuristic methods to find near-optimal solutions within a reasonable time. To elaborate on the latter, since the OCP is an $\mathcal{NP}$-hard problem [13, 14], finding optimal solutions can be time-consuming for sufficiently large instances. Moreover, if a client's priority is time rather than solution quality, then it might be sensible to use heuristic methods instead of exact methods to find a satisfactory solution in a reasonable amount of time.

The study in [11] employed three heuristic algorithms to tackle an OCP. The authors focused on finding the best algorithm capable of solving OCP problems for the surveillance of bridges. Their model aimed to minimise the total cost of cameras, while ensuring that a minimum coverage level is satisfied. Moreover, this study included different types of cameras to cover specific target points in a three-dimensional space, resulting, for example, in an increase in the number of camera locations, which can lead to the expansion of the size of the problem instances. Because of that, the authors started by examining their OCP instance using two popular heuristic methods, greedy and genetic algorithms, as well as a novel heuristic method that is called the Uniqueness Score with Local Search Algorithm (ULA). For the first method, the greedy algorithm allocates cameras starting from the cheapest one and going up until it reaches the highest possi-

ble number of cameras or achieves the minimum coverage level. At this point, the algorithm halts and provides the proposed camera network. For the second method, the genetic algorithms begins by randomly generating a population of chromosomes, where each chromosome represents a camera placement. This process continues until the minimum coverage level is reached by each chromosome. Subsequently, a subset of chromosomes that satisfy the minimum coverage is selected. Mutation and crossover operators are then applied to this subset, creating new generations and ensuring that the solutions do not become stuck in a local optimum. As demonstrated in [11], the last method, proven to be more effective than the other two approaches, begins with the first solution of ULA, inspired by the uniqueness score. It emphasises specific areas that are left uncovered by other cameras. Subsequently, a local search is used to enhance the solution by changing the selected cameras, aiming to find a new solution with a lower cost.

Another study in [22] also examined three different approaches for three OCP cases, one of which utilised an exact method, while the other two employed heuristic methods. In the first case, the objective was to maximise the coverage within a limited budget. The authors formulated the optimisation problem as a set covering problem and then applied dynamic programming to address it. For the second case, the goal was to minimise the total cost while ensuring full coverage constraints. Similar to the first case, the optimisation problem was formulated as an SCP, and branch-and-bound algorithms were employed. This involved relaxing the binary constraints, solving the new optimisation problem using primal and dual simplex methods. A solution to the original problem would be optimal if it satisfied the binary constraints. If not, further steps were taken until a satisfactory solution was found. Lastly, the third case integrated the previous two cases to form a multi-objective problem, aiming to both maximise coverage and minimise cost. Instead of looking for one optimal solution, the goal was to find a set of optimal trade-offs (Pareto optimal solutions). In this context, 'Pareto optimal' refers to a set of selected candidate cameras, where no feasible candidate cameras could improve coverage without worsening the cost simultaneously. To address this problem, the authors suggested using a heuristic method, specifically the multi-objective genetic algorithm NSGA-II [7].

Another method that was used to address an OCP problem is Differential Evolution (DE). In [23], this heuristic method was utilised to improve the performance of greedy algorithms in order to achieve full coverage. Their DE starts by using an array containing a number of cameras, where each array represents an individual within the population. The algorithm then uses the vector differential of two individuals from the previous generation to create a new individual. Consequently, the algorithm continues to generate improved individuals compared to those in the previous generations.

The work in [4] used a variant of DE, called set-based DE, inspired by the study in [15], to address their OCP problem. In contrast to the study in [23], the authors focused on minimising the number of cameras (i.e., cost reduction) while ensuring complete coverage. The difference between set-based DE and the original DE lies in the fact that the former is primarily utilised to solve

permutation-based problems, whereas the latter can be applied to solve general problems (including set-based problems).

Other methods employed in recent literature include dynamic algorithms [2], simulated annealing [17], greedy algorithms [19] and hill climbing [1]. While some OCP studies have formulated their problem as an SCP, the study in [12] argues that almost none of these studies have fully exploited this similarity or employed techniques from the SCP literature to address the OCP problem. As a result, many techniques employed in the SCP literature have not yet been utilised to address the OCP problem. This suggests potential opportunities for contributions that can make a difference in the OCP field.

## 3 Set Covering Problem (SCP)

The Set Covering Problem (SCP) is a popular combinatorial optimisation problem classified as $\mathcal{NP}$-hard [10]. Throughout the years, numerous studies have explored the SCP to tackle a diverse range of real-world applications. These include solving transit crew scheduling problems [8], optimising transit crew scheduling design with multiple objectives [16], finding optimal quantity and location of gas detectors [20], assigning fire stations with ladder trucks [21], and scheduling wireless sensor networks [24], among others. For more information regarding the SCP and its applications, readers are advised to refer to [5].

A brief definition of SCP would be: given a zero-one matrix, the goal is to obtain a subset of columns that minimises the total cost associated with the selected columns, ensuring that all the rows of the matrix are covered by these columns [18]. To elaborate on that, consider matrix $A$ with three rows and three columns:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

In this matrix, the value 1 indicates that a given row is covered by a given column, and 0 otherwise. For instance, element $a_{11}$ shows that row 1 is covered by column 1. While element $a_{33}$ shows that row 3 is not covered by column 3. If each column is associated with a specific cost, the objective of SCP would be to find a subset of columns that minimises the total cost while ensuring that all the rows of the matrix are covered by this subset. For example, if the cost is the same for all columns, an optimal solution could be selecting both columns 1 and 3, as this combination covers all given rows.

Formally, given a finite universal set $U = \{1, 2, \ldots, n\}$ and a family of subsets $S = \{S_1, S_2, \ldots, S_m\}$ where each $S_i$ is a subset of $U$, and each subset $S_i$ has an associated cost $c_i$, the set covering problem is to find a minimum-cost subset $C \subseteq S$ such that the union of the selected subsets covers the entire universal set, i.e., $\bigcup_{S_j \in C} S_j = U$. The goal is to minimise $\sum_{S_i \in C} c_i$, the total cost of the selected subsets.

From an optimisation perspective, the study in [13] asserts that the Optimal Camera Placement (OCP) can be reformulated as a set covering problem after certain pre-processing steps. In this context, the pre-processing phase involves the transformation of the OCP problem into a visibility matrix. This matrix matches each location in the surveillance area with every possible configuration (position and orientation) of the given cameras, resulting in a 0-1 matrix similar to matrix $A$. To elaborate, columns in this matrix represent the cameras, and rows represent the locations that need to be covered by the cameras. The objective of the transformed OCP problem is to find a subset of cameras that minimises the cost, ensuring that all the specified locations are covered by this subset. In essence, this conversion enables the application of SCP methodologies to address the OCP challenge effectively.

In the study conducted in [13], an in-depth illustration is provided regarding the possibility of using methods from the SCP literature and applying them to OCP problems. The study introduces different heuristic and metaheuristic methods that were employed to address SCP, and consequently, these methods could be adapted for solving OCP problems. For instance, greedy algorithm is one of the heuristic methods utilised in the SCP literature to address the problem's $\mathcal{NP}$-hard nature. As will be pointed out later, the greedy algorithm was also used in the OCP literature in recent years. Despite being tailored specifically for OCP problems, the main conceptual framework remains the same for both domains. Another approach discussed in this study involves the work of [9], who utilised the Row-Weighting Local Search (RWLS) algorithm for a special type of SCP known as the Unicost SCP. According to [13], this approach has not been explored in the OCP literature, and questions arise regarding its potential efficiency in solving OCP problems. In a subsequent study [12], different approaches from both OCP and SCP literature were employed on real OCP cases. Notably, they utilised the RWLS algorithm and demonstrated its efficacy in solving OCP problems. This illustrates the possibility of leveraging techniques from the SCP literature to effectively address challenges in the OCP domain.

## 4 Problem Description and Formulation

In this study, we address 69 three-dimensional problem instances, comprising 32 academic problem instances and 37 real-world instances. These instances were utilised as part of the GECCO 2021 competition on the optimal camera placement problem and the unicost set covering problem [3]. The academic problem instances represent different sizes of rectangularly modelled rooms, where ceilings are used for camera placement. On the other hand, the real-world instances represent diverse sizes of actual urban spaces, with the walls of multiple buildings serving as potential camera locations.

For each problem instance, a set of camera configurations (locations and orientations coordinates), referred to as *candidates*, is provided. Additionally, a grid containing a set of points in three-dimensional space, denoted as *samples*,

must be covered by a subset of candidates. For each sample, there exists multiple candidates capable of overseeing it.

Consider the example with two samples ($a$ and $b$) and four candidates (1, 2, 3, and 4). Suppose sample $a$ can be covered by candidates 1, 2, and 4, while sample $b$ can be covered by candidates 2 and 3. We can create a visibility matrix $A$ for this example, where the first row represents sample $a$, the second row represents sample $b$, and columns represent candidates 1, 2, 3, and 4. Matrix $A$ is presented below.

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

The objective of the OCP problem is to cover all the given samples by identifying a subset of candidates that achieves this goal at the minimum cost. In the presented OCP example, assuming equal costs for each candidate, the optimal solution would be to select candidate 2. This choice is optimal because candidate 2 is the only one overseeing both samples, enabling the coverage of both samples with a single candidate. Other solutions, such as candidates 1 and 3 or candidates 3 and 4, would require paying for two candidates to cover the two samples. However, the optimal solution, in this case, is achieved by selecting just one candidate (i.e., candidate 2).

### 4.1 Mathematical Formulation

As previously mentioned, the OCP problem shares similarities with the set covering problem. Consequently, we model the OCP problem as an SCP. The formulation involves a binary matrix $A$ ($a_{ij} = 1$ if row $i$ can be covered by column $j$, and 0 otherwise) with $m$ rows and $n$ columns, where each column $j$ is assigned a specific cost $c_j$. The mathematical model is expressed as follows:

$$\begin{aligned} \text{minimise} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq 1, \forall i \in \{1, \ldots, m\} \\ & x_j \in \{0, 1\}, \quad \forall j \in \{1, \ldots, n\} \end{aligned}$$

Here, $x_j = 1$ if column $j$ is selected, and $x_j = 0$ otherwise. The objective function minimises the total cost, and the first constraints ensure that each row $i$ is covered by at least one selected column $j$.

Now, for our specific OCP problem, all cameras have the same cost [3]. This simplification transform our SCP model into a Unicost SCP (USCP), where the cost parameter $c_j$ is omitted:

$$\begin{aligned} \text{minimse} \quad & \sum_{j=1}^{n} x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq 1, \forall i \in \{1, \ldots, m\} \\ & x_j \in \{0, 1\}, \quad \forall j \in \{1, \ldots, n\} \end{aligned}$$

In this formulation, $x_j = 1$ if candidate $j$ is included in the solution, and $x_j = 0$ otherwise. The objective remains to minimise the total number of selected candidates, ensuring that each sample is covered by at least one candidate. This transition to a USCP model simplifies the objective by focusing on minimising the number of candidates, each having an equal cost, while maintaining the essential constraints for effective camera placement in the OCP context.

## 5 Computational Results

For this study, we utilised the PuLP[1] package in Python to build the model, which was then solved using Gurobi, a commercial Linear Programming solver. Experiments were conducted on an Intel(R) Core(TM) i5-8500T CPU @ 2.10GHz 2.11GHz with 8.00GB RAM. A time limit of 3 hours was imposed on each problem instance, encompassing both model building and problem solving. The results for both the academic and real-world problem instances are presented in Table 1 and Table 2, respectively. For instances where obtaining the optimal solution was unattainable, we provide the upper and lower bounds along with the calculated optimality gap using the formula:

$$\text{Gap} = \frac{|UB - LB|}{|UB|}$$

where $UB$ is the upper bound, and $LB$ is the lower bound. In cases where the optimal solution was found, both the upper and lower bounds will have the same value. Instances where no results were produced due to insufficient memory or time limitations are denoted by a dash '-' in the respective table entries.

As seen in both tables, there are many cells filled with a dash '-', mostly because the three-hour time limit was reached before completing the model building stage, rendering it impossible to solve the problem. This issue arises mainly due to the substantial size of these problem instances, requiring hours or even days solely for model building. Another observation is that, even when the model is built for some instances, an optimal solution is not always achieved. In such cases, having the upper and lower bounds is particularly valuable, offering useful information about how close we are to reaching the optimal solution. For some instances, like RW_18, it appears that we are quiet close to the optimal solution. For these, extending the runtime may yield the optimal solutions. However, there

[1] https://pypi.org/project/PuLP/

Table 1: Results for academic problem instances. $m$ is the number of samples, $n$ is the number of candidates, UB is the upper bound, LB is the lower bound, Gap is the optimality gap, and Time is the time taken in seconds

| Instance | $m$ | $n$ | UB | LB | Gap | Time |
|---|---|---|---|---|---|---|
| AC_01 | 605 | 2904 | 7.00 | 7.00 | 0.00% | 9.33 |
| AC_02 | 2205 | 10584 | 4.00 | 4.00 | 0.00% | 140.30 |
| AC_03 | 4805 | 23064 | 3.00 | 3.00 | 0.00% | 1411.43 |
| AC_04 | 8405 | 40344 | 5.00 | 5.00 | 0.00% | 7542.09 |
| AC_05 | 13005 | 62424 | - | - | - | - |
| AC_06 | 18605 | 89304 | - | - | - | - |
| AC_07 | 32805 | 157464 | - | - | - | - |
| AC_08 | 51005 | 244824 | - | - | - | - |
| AC_09 | 73205 | 351384 | - | - | - | - |
| AC_10 | 605 | 2904 | 20.00 | 17.37 | 10.00% | 10800.00 |
| AC_11 | 2205 | 10584 | 72.00 | 52.00 | 26.39% | 10800.00 |
| AC_12 | 4805 | 23064 | 168.00 | 109.81 | 34.52% | 10800.00 |
| AC_13 | 8405 | 40344 | 344.00 | 187.18 | 45.35% | 10800.00 |
| AC_14 | 13005 | 62424 | 723.00 | 0.00 | 100.00% | 10800.00 |
| AC_15 | 18605 | 89304 | - | - | - | - |
| AC_16 | 32805 | 157464 | - | - | - | - |
| AC_17 | 51005 | 244824 | - | - | - | - |
| AC_18 | 73205 | 351384 | - | - | - | - |
| AC_19 | 99405 | 477144 | - | - | - | - |
| AC_20 | 129605 | 622104 | - | - | - | - |
| AC_21 | 163805 | 786264 | - | - | - | - |
| AC_22 | 202005 | 969624 | - | - | - | - |
| AC_23 | 244205 | 1172184 | - | - | - | - |
| AC_24 | 290405 | 1393944 | - | - | - | - |
| AC_25 | 340605 | 1634904 | - | - | - | - |
| AC_26 | 394805 | 1895064 | - | - | - | - |
| AC_27 | 453005 | 2174424 | - | - | - | - |
| AC_28 | 515205 | 2472984 | - | - | - | - |
| AC_29 | 581405 | 2790744 | - | - | - | - |
| AC_30 | 651605 | 3127704 | - | - | - | - |
| AC_31 | 725805 | 3483864 | - | - | - | - |
| AC_32 | 804005 | 3859224 | - | - | - | - |

are instances, such as AC_13, where the gap is notably large, indicating the need for an extended runtime to solve them.

If we take a look at the first problem instance in Table 1, AC_01 contains 605 samples and 2904 candidates. The solution for this problem instance involved using 7 candidates to cover all 605 samples, and the computation took 9.33 seconds. Figure 1 visualises this problem instance, with yellow points representing all candidates and blue points representing samples that must be covered by a subset of those candidates. The solution of AC_01 is also depicted in Figure 1, where 7 yellow points on the graph represent the optimal locations of the candidates that cover all 605 samples.

Similar to AC_01, we visualise another problem instance in Figure 2, namely RW_22. This real-world instance contains 17,203 candidates and 83,835 samples.

Table 2: Results for real-world problem instances. $m$ is the number of samples, $n$ is the number of candidates, UB is the upper bound, LB is the lower bound, Gap is the optimality gap, and Time is the time taken in seconds

| Instance | $m$ | $n$ | UB | LB | Gap | Time |
|---|---|---|---|---|---|---|
| RW_01 | 153368 | 32430 | - | - | - | - |
| RW_02 | 285698 | 56132 | - | - | - | - |
| RW_03 | 161099 | 32040 | - | - | - | - |
| RW_04 | 304655 | 59137 | - | - | - | - |
| RW_05 | 206900 | 34568 | - | - | - | - |
| RW_06 | 380420 | 65691 | - | - | - | - |
| RW_07 | 214889 | 42046 | - | - | - | - |
| RW_08 | 382651 | 77986 | - | - | - | - |
| RW_09 | 206816 | 39003 | - | - | - | - |
| RW_10 | 368114 | 71323 | - | - | - | - |
| RW_11 | 82437 | 15632 | 316.00 | 309.88 | 1.90% | 10800.00 |
| RW_12 | 136555 | 28109 | - | - | - | - |
| RW_13 | 293138 | 61741 | - | - | - | - |
| RW_14 | 81062 | 14916 | 337.00 | 334.77 | 0.59% | 10800.00 |
| RW_15 | 141309 | 27008 | - | - | - | - |
| RW_16 | 105829 | 21063 | - | - | - | - |
| RW_17 | 180453 | 35635 | - | - | - | - |
| RW_18 | 79947 | 14423 | 338.00 | 336.65 | 0.30% | 10800.00 |
| RW_19 | 141114 | 26483 | - | - | - | - |
| RW_20 | 332300 | 50284 | - | - | - | - |
| RW_21 | 654068 | 90050 | - | - | - | - |
| RW_22 | 83835 | 17203 | 399.00 | 392.93 | 1.50% | 10800.00 |
| RW_23 | 142326 | 31038 | - | - | - | - |
| RW_24 | 201967 | 33880 | - | - | - | - |
| RW_25 | 375680 | 59851 | - | - | - | - |
| RW_26 | 105566 | 18043 | - | - | - | - |
| RW_27 | 181090 | 32669 | - | - | - | - |
| RW_28 | 136755 | 27838 | - | - | - | - |
| RW_29 | 273964 | 49267 | - | - | - | - |
| RW_30 | 263518 | 49354 | - | - | - | - |
| RW_31 | 472660 | 87248 | - | - | - | - |
| RW_32 | 124289 | 30189 | - | - | - | - |
| RW_33 | 229231 | 55000 | - | - | - | - |
| RW_34 | 134479 | 27329 | - | - | - | - |
| RW_35 | 238546 | 47590 | - | - | - | - |
| RW_36 | 135043 | 28162 | - | - | - | - |
| RW_37 | 238492 | 50702 | - | - | - | - |

A feasible solution was obtained with an upper bound of 399, a lower bound of 392.93, and a gap of 1.5%. The visual representation of this problem's solution can be observed in the same figure.

Addressing the time issue is paramount for achieving improved results. One approach involves tackling the size of the problem; for instance, reducing it by eliminating unnecessary sets from the visibility matrix. Additionally, exploring different optimisation methods could be beneficial, such as employing multi-

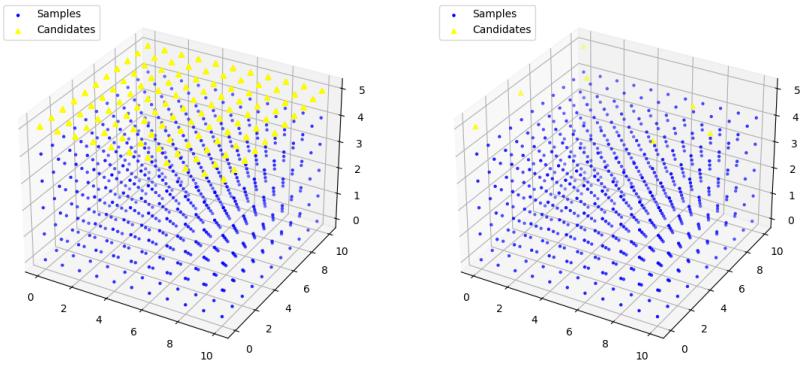


Fig. 1: Visualisation of the AC_01 problem instance (left) and its optimal solution (right), where yellow points represent candidates and blue points represent samples to be covered

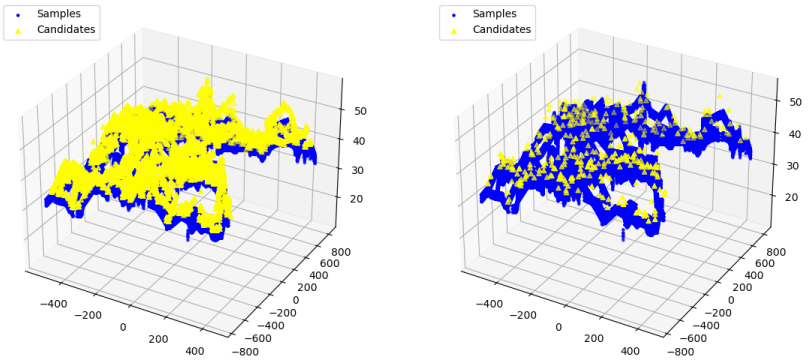


Fig. 2: Visualisation of the RW_22 problem instance (left) and its feasible solution (right), where yellow points represent candidates and blue points represent samples to be covered

objective optimisation. In this approach, the main objectives would involve maximising area coverage while minimising the number of cameras. Another viable approach is the utilisation of heuristic techniques. By studying various heuristic methods employed in the SCP literature and applying them to our OCP problem instances, we can potentially obtain near-optimal results within a reasonable amount of time.

## 6 Conclusions

OCP problem involves determining the optimal locations and orientations for a set of cameras, with the objective either being to maximise the coverage of a given surveillance area or to minimise the total cost or number of selected cameras.

Although this problem has been formulated in a manner resembling the SCP, this similarity has only recently been explicitly discussed in the literature.

This paper studied the OCP problem by exploring its literature and understanding its relationship with SCP. Given that OCP is an $\mathcal{NP}$-hard problem, and the majority of problem instances are large, relying solely on exact methods did not provide solutions for all instances. Therefore, addressing the $\mathcal{NP}$-hard nature of the problem becomes crucial for future work. Potential strategies include reducing the size of problem instances, and/or resorting to heuristic techniques.

## References


1. Ahn, J.W., Chang, T.W., Lee, S.H., Seo, Y.W.: Two-phase algorithm for optimal camera placement. Scientific Programming **2016** (2016)
2. Altahir, A.A., Asirvadam, V.S., Hamid, N.H.B., Sebastian, P., Saad, N.B., Ibrahim, R.B., Dass, S.C.: Optimizing visual surveillance sensor coverage using dynamic programming. IEEE Sensors Journal **17**(11) (2017) 3398–3405
3. Brévilliers, M., Lepagnot, J., Idoumghar, L.: Gecco 2021 competition on the optimal camera placement problem (ocp) and the unicost set covering problem (uscp) (2021)
4. Brévilliers, M., Lepagnot, J., Kritter, J., Idoumghar, L.: Parallel preprocessing for the optimal camera placement problem. International Journal of Modeling and Optimization **8**(1) (2018) 33–40
5. Caprara, A., Toth, P., Fischetti, M.: Algorithms for the set covering problem. Annals of Operations Research **98**(1) (2000) 353–371
6. Chvátal, V.: A combinatorial theorem in plane geometry. Journal of Combinatorial Theory, Series B **18**(1) (1975) 39–41
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2) (2002) 182–197
8. Desrochers, M., Soumis, F.: A column generation approach to the urban transit crew scheduling problem. Transportation science **23**(1) (1989) 1–13
9. Gao, C., Yao, X., Weise, T., Li, J.: An efficient local search heuristic with row weighting for the unicost set covering problem. European Journal of Operational Research **246**(3) (2015) 750–761
10. Garey, M.R., Johnson, D.S.: Computers and intractability. A Guide to the (1979)
11. Jun, S., Chang, T.W., Yoon, H.J.: Placing visual sensors using heuristic algorithms for bridge surveillance. Applied Sciences **8**(1) (2018) 70
12. Kritter, J., Brévilliers, M., Lepagnot, J., Idoumghar, L.: On the real-world applicability of state-of-the-art algorithms for the optimal camera placement problem. In: 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), IEEE (2019) 1103–1108
13. Kritter, J., Brévilliers, M., Lepagnot, J., Idoumghar, L.: On the optimal placement of cameras for surveillance and the underlying set cover problem. Applied Soft Computing **74** (2019) 133–153
14. Liu, J., Sridharan, S., Fookes, C.: Recent advances in camera planning for large area surveillance: A comprehensive review. ACM Computing Surveys (CSUR) **49**(1) (2016) 1–37

15. Maravilha, A.L., Ramirez, J.A., Campelo, F.: A new algorithm based on differential evolution for combinatorial optimization. In: 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, IEEE (2013) 60–66
16. Owais, M., Osman, M.K., Moussa, G.: Multi-objective transit route network design as set covering problem. IEEE Transactions on Intelligent Transportation Systems **17**(3) (2015) 670–679
17. Rahimian, P., Kearney, J.K.: Optimal camera placement for motion capture systems. IEEE transactions on visualization and computer graphics **23**(3) (2016) 1209–1221
18. Ren, Z.G., Feng, Z.R., Ke, L.J., Zhang, Z.J.: New ideas for applying ant colony optimization to the set covering problem. Computers & Industrial Engineering **58**(4) (2010) 774–784
19. Suresh, M.S., Narayanan, A., Menon, V.: Maximizing camera coverage in multi-camera surveillance networks. IEEE Sensors Journal **20**(17) (2020) 10170–10178
20. Vianna, S.S.: The set covering problem applied to optimisation of gas detectors in chemical process plants. Computers & Chemical Engineering **121** (2019) 388–395
21. Walker, W.: Application of the set covering problem to the assignment of ladder trucks to fire houses. Operations Research **22** (1974) 275–277
22. Yang, X., Li, H., Huang, T., Zhai, X., Wang, F., Wang, C.: Computer-aided optimization of surveillance cameras placement on construction sites. Computer-Aided Civil and Infrastructure Engineering **33**(12) (2018) 1110–1126
23. Zhang, B., Zhang, X., Chen, X., Fang, Y.: A differential evolution approach for coverage optimization of visual sensor networks with parallel occlusion detection. In: 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), IEEE (2016) 1246–1251
24. Zhang, X.Y., Zhang, J., Gong, Y.J., Zhan, Z.H., Chen, W.N., Li, Y.: Kuhn–munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks. IEEE Transactions on Evolutionary Computation **20**(5) (2015) 695–710