Efficient Hybrid Metaheuristics for the Optimal Camera Placement Problem

Juliana Henao[1]

Advisor:
Juan Carlos Rivera[2]

Research practice I
Research proposal
Mathematical Engineering
Department of Mathematical Sciences
School of Sciences
Universidad EAFIT

June 2021

[1]Mathematical Engineering student. Universidad EAFIT jhenaoa4@eafit.edu.co (CvLAC: https://scienti.minciencias.gov.co/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0001839296)

[2]Mathematical Modeling Research Group. Mathematical Science Department. School of Sciences. Universidad EAFIT. jrivera6@eafit.edu.co

Abstract

To perform various surveillance tasks the use of camera networks is now important and the way this networks are placed can be a very complex problem to solve. In this regard, a variation of the well-known Set Covering Problem can be used to model this issue and its solution brings a way to place the camera network concerned. Thus, in this project there was implemented a Hybrid Metaheuristics, also evaluated its performance in order to solve the Optimal Camera Placement problem. The algorithm consist in a constructive part with a local neighbourhood search and a genetic part with exacts methods solved in Gurobi. The performance of the metaheuristics shows good solutions to the data sets used in the experiments.

Keywords: Hybrid Metaheuristic, Set Covering Problem, Optimal Camera Placement Problem, Genetic algorithm, LNS.

1 Introduction

The *Set Covering Problem* (SCP) is a well-known NP-hard problem (Karp, 1972). Consequently, it is very important in fields like operations research, complexity theory, computer science, and combinatorics. As well, it is known that many problems can be formulated as an SCP, such as the *Unicost Set Covering Problem* and the *Optimal Camera Placement Problem*.

The *Optimal Camera Placement Problem* (OCP) is a special case of the of the *Unicost Set Covering Problem* (USCP) that consists of finding the optimal way to place cameras in order to cover an area to be monitored. These camera networks can be used for a numerous surveillance tasks and, along with intelligent systems, it can analyze video streams to detect, identify and track events of interest, objects or persons. Therefore, the configuration of the cameras is crucial to reduce costs, by minimizing the number of cameras, and to follow the constraints of coverage and image quality. To this end, is considered a surveillance area that is reduced to a set of three-dimensional sample points to be covered and camera configurations are sampled in the candidates, each with a given set of position and orientation coordinates. A candidate can have several samples within its range, and a sample can be viewed by several candidates (Brévilliers *et al.*, 2020).

A mathematical model for the OCP has been proposed by Brévilliers *et al.* (2020) as follows:

The monitored area is defined as a rectangular box whose point coordinates (in meters) range from $(0, 0, 0)$ to $(X_{max}, Y_{max}, Z_{max})$ in a Cartesian coordinate system of the three–dimensional Euclidean space $\mathbb{R}^3$. This area is discretized and approximated by a regular grid of sample points with a step size U (in meters) between two adjacent samples. If the samples are labelled with integer numbers (representing the set of elements to be covered), then any candidate can be modelled as a set of integers (corresponding to the labels of the samples it covers), and the OCP can be formulated as a USCP in a straightforward manner: given the set I of elements (i.e. samples) and a collection J of sets (i.e. candidates), solving the OCP comes down to find the minimum subset of J that covers I. Now, the following decision variables can be defined:

$$x_c = \begin{cases} 1 & \text{if candidate c is used,} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ c \in J \tag{1}$$

Then, the corresponding 0–1 integer linear programming model can be written as follows:

$$\min Z = \sum_{c \in J} x_c \tag{2}$$

$$\text{subject to} \sum_{\substack{c \in J \\ s \in c}} x_c \geq 1, \qquad \forall\ s \in I \tag{3}$$

$$x_c \in \{0, 1\}, \qquad \forall\ c \in J \tag{4}$$

The objective function (Equation 17) minimizes the total number of used candidates. Restrictions (3) indicate that each sample has to be covered by at least one candidate (full coverage constraint). Expressions (4) give the set of binary constraints needed for the decision variables described in definition (12) (Brévilliers *et al.*, 2020).

So, in order to solve this problem an efficient hybrid metaheuristic algorithm was developed, this includes a constructive part and a genetic part. The *Set Covering Problem* (SCP) is a well-known NP-hard problem (Karp, 1972). Consequently, it is very important in fields like operations research, complexity theory, computer science, and combinatorics. As well, it is known that many problems can be formulated as an SCP, such as the *Unicost Set Covering Problem* and the *Optimal Camera Placement Problem*. Due to that, one motivation of this research is the study and development of innovating techniques for solving that kind of problem, in particular the OCP problem. Furthermore, the applications are very wide, including arrangement of ambulances, scheduling, personnel shift planning, transportation safety, construction of optimal logical circuits, inspection of computer viruses (Musliu, 2006a) and, of course, in surveillance.

With this in mind, first there is a literature review with some of the work made before related to this matter. Then, the solution method for the problem described above is presented, this includes the whole hybrid metaheuristic algorithm. After this, there are some experiments and its results so the performance of the algorithm can be shown. Finally, the last section summarizes the main conclusions, contributions and findings of the project.

2 State of the art

The *Set Covering Problem* is a classic problem which Richard Karp appointed as one of Karp's 21 NP-complete problems in 1972 (Karp, 1972). After that, many researchers have addressed this subject and proposed new methods to find solutions to the problem. Such as the one proposed by J.E. Beasley and RC. Chu in 1994 (Beasley & Chu, 1995), which is a genetic algorithm-based heuristic that propose several modifications to the basic genetic procedures including a new fitness-based crossover operator (fusion), a variable mutation rate and a heuristic feasibility operator tailored specifically for the set covering problem (Beasley & Chu, 1995). Also, two set-covering algorithms are proposed and proved in 1989 (Murav'ev, 1989), one selecting and ordering the covering candidates by degrees and the other using lexicographic ordering (Murav'ev, 1989).

As well a new solutions were given by researchers over time, they have also proposed variants of the original problem, like the ones that will be discussed in this research practice: the *Unicost Set Covering Problem* and the *Optimal Camera Placement Problem* which, as it is said above, is a special case of the first one. There are some recent proposed algorithms as the one by Joaquín Bautista and Jordi Pereira (Bautista & Pereira, 2006), who presents a GRASP algorithm to solve the unicost set covering problem by incorporating a local improvement procedure based on the heuristics to solve binary constraint satisfiability problems (Bautista & Pereira, 2006). Also it is proposed a local search based algorithm for the same problem (Musliu, 2006b) in which a fitness function is proposed and different neighborhood relations are considered for the exploration of the neighborhood of the current solution (Musliu, 2006b).

For the *Optimal Camera Placement Problem* there have been also recent studies in the subject, one proposed method is based on a semi-definite programming approximation (Ercan *et al.*, 2006). Also in the paper "Optimal Camera Placement for Automated Surveillance Tasks" (Bodor *et al.*, 2007), it is developed a general analytical formulation of the observation problem, in terms of the statistics of the motion in the scene and the total resolution of the observed actions (Bodor *et al.*, 2007). Finally, A. Bottino and A. Laurentini (Bottino & Laurentini, 2005) proposed the following, a optimal sensor position is determined based on the intersecting line of vertexes and regions of the objects inside specific 3D setting (Bottino & Laurentini, 2005).

3 Solution method

In order to solve the problem set above, a hybrid metaheuristic was developed, this algorithm has two parts, first there is a constructive hybrid part where a initial solution is built using also a Local Neighborhood Search (LNS) algorithm (destroy and repair), then that solution will be part of a initial population for a hybrid genetic algorithm. This algorithms are described below.

Hybrid Constructive Metaheuristic

This metaheuristic follows this algorithm:

1. Creates a matrix *cover*, which for each sample s: the number of candidates which cover sample s, and a list of the candidates which cover sample s.

2. Creates a matrix *candidates*, which for each candidate c: the number of samples which are covered by candidate c, and a list of the samples which are covered by candidate c.

3. Look for, in the list *cover*, the sample with the least number of candidates that cover it.

4. Look for, in the set of candidates that cover the selected sample in 3, the candidate with the most number of samples. And include that candidate in the solution.

5. Actualize lists *cover* and *candidates* by deleting the samples that are covered by the selected candidate in 4 and deleting the candidate that is already in the solution. Also, delete from *candidates* the samples already covered.

6. Repeat 3, 4 and 5 until all samples are covered.

7. The solution given in the previous procedure is destroyed using $Destruction1$

8. The destroyed solution is repaired with an exact method

9. The solution given in the previous procedure is destroyed again, but this time using $Destruction2$
10. The last destroyed solution is repaired with another exact method
11. Pick the solution with the less number of candidates, between the ones before the destructions, and the two after each repair.
12. Return this solution in order to use it in the Genetic Metaheuristic as part of the initial population

For the LNS part of the algorithm, four methods are used:

- **Destruction 1:** takes the solution and an *alpha* percent is deleted from it.
- **Repair 1:** takes the destroyed solution and uses an exact method in Gurobi to repair it, using the following model

$$x_i = \begin{cases} 1 & \text{if candidate i is used,} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ c \in J \tag{5}$$

$$P_{ij} = \begin{cases} 1 & \text{if candidate i covers sample j,} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ c \in J \tag{6}$$

$$\min Z = \sum_{i=1}^{n} x_i \tag{7}$$

$$\text{subject to} \qquad \sum_{i \in J} P_{ij} x_i \geq 1, \qquad \forall\ i \in J \tag{8}$$

$$x_i \geq binarySolution_i, \qquad \forall\ i \in J \tag{9}$$

$$x_i \in \{0, 1\}, \qquad \forall\ i \in J \tag{10}$$

- **Destruction 2:** takes the solution and a *beta* percent of the solution is added to this.
- **Repair 2:** takes the destroyed solution and uses an exact method in Gurobi to repair it, using the following model

$$x_i = \begin{cases} 1 & \text{if candidate i is used,} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ c \in J \tag{11}$$

$$P_{ij} = \begin{cases} 1 & \text{if candidate i covers sample j,} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ c \in J \tag{12}$$

$$\min Z = \sum_{i=1}^{n} x_i \tag{13}$$

$$\text{subject to} \qquad \sum_{i \in J} P_{ij} x_i \geq 1, \qquad \forall\ i \in J \tag{14}$$

$$x_i \leq binarySolution_i, \qquad \forall\ i \in J \tag{15}$$

$$x_i \in \{0, 1\}, \qquad \forall\ i \in J \tag{16}$$

This exacts methods where solve by the mathematical optimization solver Gurobi, as well as the exact methods proposed bellow.

Hybrid Genetic Metaheuristic

The algorithm for this metaheuristic is

Alg. 1 Hybrid GA|PM–LNS for the OCP

$P \leftarrow initial\_population()$
while $Stop\_critera = false$ **do**
  for $s = 1$ to $|P|$ **do**
    $(i, j) \leftarrow Selection(P)$
    $h \leftarrow Crossover(i, j, P)$
    $h' \leftarrow Mutation(h)$
    $H \leftarrow H \cup h \cup h'$
  end for
  $P \leftarrow Update(P, H)$
end while
$s^* \leftarrow Best(P)$
Return s^*

Initial population

The initial population, P, is created by a small set of solutions, about 10. One solution is created by the constructive procedure described above, while the remaining solutions are created by a randomized procedure.

The randomized constructive solutions are created by the following procedure: Candidates are chosen randomly until all samples are covered by at least one candidate. This procedure should generate a solution with more candidates than necessary. For this reason, the repair procedure 2 is executed to eliminate the excess candidates in an optimal way.

Selection

Two individuals are selected from the population so that one of the parents (i) is selected among the best 5 individuals, while the second one can be anyone (j). The more samples a individual covers, without repetitions, the best it is.

Crossover

The child solution h is created in two phases. The first consists of selecting all candidates belonging to the parent solutions, i and j.

$$h(k) = maxP(i, k), P(j, k)$$

.

In the second phase, the repair 1 procedure is used to eliminate as many candidates as necessary. In this procedure it is suggested to use as a tie-breaking criterion the choice of candidates that maximize the difference between the solution h and the solutions i.

Mutation

The mutation operator runs an LNS (destroy and repair) algorithm on the solution h to obtain an alternative h' solution. The procedures destruction 2 and repair 2 are used for the LNS.

Update

This function updates the population considering the population of parents P and the population of children H.

The *Update* function has the following characteristics: a.) It always includes the best solution in $P \cup H$, b.) It completes the population with the solutions that offer the greatest diversity to the population.

To see how much diversity offers a solution a distance between solution is defined the following function

$$dist_{ij} = \sum_{k \in P_i} \sum_{k \in P_j} c_k$$

where

$$c_k = \begin{cases} 1 & \text{if candidate } k \text{ from solution } i \text{ is also in solution } j, \\ 0 & \text{otherwise} \end{cases}$$

Considering the distances between solution, the *Update* method consist in solving the following model

$$\begin{aligned} & & \min Z = \lambda & & (17)\\ \text{subject to} & & \lambda \geq dist_{ij} - 100(2 - x_i - x_j), & \quad \forall\, i, j & (18)\\ & & \sum_i x_i = |P|, & \quad \forall\, i \in J & (19)\\ & & x_b = 1 & & (20)\\ & & x_i \in \{0, 1\}, & \quad \forall\, i & (21) \end{aligned}$$

where x_i means if a solution will be considered in the actualized population and b is the index of the the solution with the best objective function.

Best

The *Best* function obtains the best solution in the population P, which is the one with less number of candidates.

Stop criteria

This algorithm stops running when 5 minutes have passed.

4 Results

The performance of the algorithm was measured by experiments which had different sets of data. Each instance is a experiment with different candidates and samples. So in Table 1 there a summary of the main results, first there is the objective function (which is the number of cameras selected) given by the constructive part of the method, then the objective function after the first and second local search, then the time in minutes that took the whole method to solve the problem without the time it takes reading the data, finally it is shown the size of the instances, this means the number of candidates considered and the number of samples to cover.

Instance	**Constructive**	**First LNS**	**Second LNS**	**Genetic**	**Time (min)**	**Samples**	**Candidates**
01	9	8	9	7	5.604561	605	2904
02	4	4	4	4	8.662752	2205	10584
03	4	4	4	4	49.58044	4805	23064
04	7	7	7	6	100.2926	8405	40344
05	10	10	13	10	212.4025	13005	62424

Table 1: Results for different instances

As it can be seen the constructive algorithm gives very good results, so that, often, it can not be improved by the local search or genetic algorithm. Nevertheless, in some cases the genetic algorithm leads to a feasible solution that the constructive could never reach, besides, for big instances most of the run time used is by the constructive method, since the genetic algorithm only used around five minutes of this time. This results show that the genetic part is worth the time because is not only short but it is effective.

In matter of time this algorithm starts failing when big instances are presented.

5 Conclusions and future research

The presented method is a novel solution strategy for the Optimal Camera Placement problem and as it shown it gives good solutions to it, in terms of the objective function, the performance of the algorithm gives satisfactory results; nevertheless in matter of run time, the algorithm had not such good performance. Because of that, in further time, experiments with a bigger set of data can be made if it has enough time to run this instances.

However, since the OCP is a special case of the Unicost Set Covering Problem, this solution method can be adapted to this kind of problem and solve more problems that maybe are not as specific as the OCP. This and others applications can be seen as it is seen how useful this model is and as effective this hybrid metaheuristic is. In conclusion, this project also shows the capability of merging different heuristic strategies as constructive, LNS and genetic methods.

References

Bautista, Joaquín, & Pereira, Jordi. 2006. A GRASP algorithm to solve the unicost set covering problem.

Beasley, J.E., & Chu, RC. 1995. A genetic algorithm for the set covering problem.

Bodor, Robert, Drenner, Andrew, Schrater, Paul, & Papanikolopoulos, Nikolaos. 2007. Optimal Camera Placement for Automated Surveillance Tasks.

Bottino, A., & Laurentini, A. 2005. Optimal Positioning of Sensors in 3D.

Brévilliers, Mathieu, Lepagnot, Julien, & Idoumghar, Lhassane. 2020. GECCO 2021 Competition on the Optimal Camera Placement Problem (OCP) and the Unicost Set Covering Problem (USCP).

Ercan, Ali O., Yang, Danny B., Gamal, Abbas El, & Guibas, Leonidas J. 2006. Optimal Placement and Selection of Camera Network Nodes for Target Localization.

Karp, R. M. 1972. *Reducibility among Combinatorial Problems.* Springer US, Boston, MA.

Murav'ev, S. V. 1989. Two methods of ordering the covering elements for the solution of the set covering problem.

Musliu, Nysret. 2006a. Local Search Algorithm for Unicost Set Covering Problem.

Musliu, Nysret. 2006b. Local Search Algorithm for Unicost Set Covering Problem.