Rectangle Covering

Kristóf Kovács[1,a),*] and Boglárka G.-Tóth[2,b),*]

[1]*Budapest University of Technology and Economics, 1111 Budapest, Egry József u. 1.*
[2]*University of Szeged, Department of Computational Optimization, 6725 Szeged, Árpád tér 2.*

[a)]Corresponding author: kkovacs@math.bme.hu
[b)]boglarka@inf.szte.hu

Abstract. The problem we discuss was proposed by an industrial partner. The aim is to locate light sources around a rectangular field so that the whole field is covered. We assume these lighted areas to be rectangular as well, parallel to the field. Covering an area with multiple light sources is allowed. There are multiple types of light sources, priced differently with different attributes. Our aim is to minimize the cost of the cover.

We propose two solution approaches for solving this covering problem.In the first direct approach, we formulate the problem by a single MIP model, using constraints that force every rectangle to have proper neighbors. In the second constraint generation approach, we formulate a MIP model to locate the light sources such that a finite number of predetermined points of the field have to be covered. If the result solves the original problem, since it covers the whole field, then we stop. Otherwise, a constraint generation model calculates a non-covered point such that the first model has to improve its previous solution to cover this new point as well.

INTRODUCTION

Covering problems are very common in optimization, as they have many applications in computer graphics, artificial intelligence and in the industry. In general, a covering problem is to find a minimal cover of a given set (e.g. a polygon) by some mold (e.g. rectangles). In [2] D. S. Franzblau and D. J. Kleitman introduce an algorithm to find the minimal cover of a rectilinear polygon by rectangles. The rectangles can be arbitrary sized and overlapping is allowed, but all of them must be wholly contained in the polygon. The objective is to find a cover with the least amount of rectangles.

The Manhattan p-center problem can also be seen as a covering problem by squares, where a set of points has to be covered by p equal sized squares, minimizing the size of the congruent covering squares. In [1] Drezner gives an $O(n \log n)$ algorithm for the rectangular (Manhattan) 3-center problem. A general algorithm for the p-center problem also exists, although far less efficient.

The dual problem of covering is the packing problem. E. Huang and R. E. Korf study a rectangle packing problem in [4], where the objective is to find the smallest enclosing rectangle to contain a given set of rectangles without overlap. The problem appears frequently in computer graphics, as compressing and using a single large graphics file is more efficient than using many smaller ones.

Our problem is different from the above mentioned ones in many aspects. We have different types of lights, with varying sizes of lighted areas simplified to rectangles and different prices. The real lighted area is fan-shaped however in the industrial problem it is required to avoid elongated illuminated spots. Thus, cutting off these unfitting areas, the shape simplifies almost into a rectangle. The objective is to cover the whole field while minimizing the total cost of the lights. Each light source needs to be at one of the sides of the field, hence the rectangles have to lie on the border of the field.

*This research was supported by the project "Deepening the activities of the Hungarian Industrial Innovation Mathematical Service Network HU-MATHS-IN ", no. EFOP-3.6.2-VEKOP-16-2017-0016, by the Hungarian National Research, Development and Innovation Office - NKFIH (OTKA grant PD115554) and by the European Union. It was also co-funded by the European Social Fund.

MODEL

We initially formulate a single model to solve the problem (hereinafter referred to as the single model). Our main variables are the x_i and y_i coordinates of the light sources. Also, for all light $i \in L$ its type $k \in K$ and rotation $r \in R$ is represented by a binary variable t_{irk} (that is 1 if light i is type k with rotation r).

A few constraints with dummy variables ensure that the light sources are placed around the edges of the field as they can only be placed on the side of the field. The model ensures that every corner of a covered rectangle is inside of at least 2 other rectangles. That is what we call having proper neighbors.

We formulate here the most important and not too technical parts of the model, the complete model can be seen in [3]. The objective function is minimizing the price of the used light sources:

$$\min \sum_{i \in L,\ k \in K,\ r \in R} t_{irk} price_k, \tag{1}$$

where $price_k$ is the price of a light source type k.

We introduce a couple of dummy variables α_{ijc}, each of which takes value 1 if corner c of light i is in light j, 0 otherwise. To set these variables we have constraints (2) and (3), where xc_{ic} are dummy variables set to the x coordinate of corner c of light source i and M is the usual big-M set to a reasonably high number.

$$xc_{j1} - (1 - \alpha_{ijc})M \leq xc_{ic} \quad \forall i \in L, j \in L, c \in C \tag{2}$$

$$xc_{ic} \leq xc_{j3} + (1 - \alpha_{ijc})M \quad \forall i \in L, j \in L, c \in C \tag{3}$$

We have similar constraints for the y coordinates. For a visual explanation of the corner variables xc_{ic}, yc_{ic} see Figure 1.

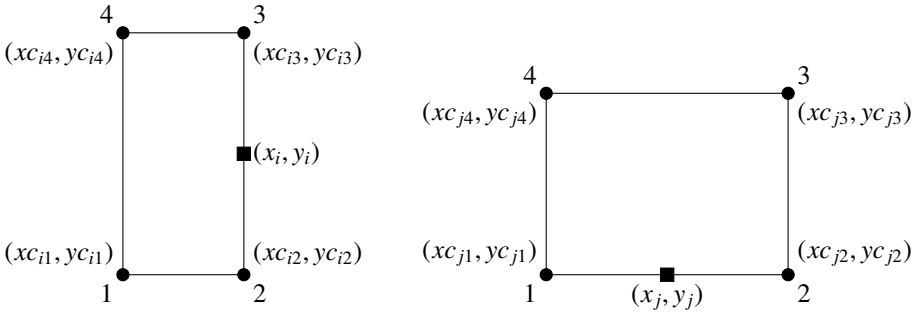

FIGURE 1. Visual explanation of the corner variables, for a laser i with rotation "left" and j with rotation "up".

Constraint (4) ensures that no two rectangles overlap completely, while constraint (5) ensures that each corner is covered by at least 3 light sources, including its own. Lastly constraint (6) ensures that light sources that we do not use are in turn not used to cover the corners of others. The light source i is used only if $\sum_{rk} t_{irk}$ is 1.

$$\sum_{c} \alpha_{ijc} \leq 2 \quad \forall i \in L, j \in L, i \neq j \tag{4}$$

$$\sum_{j} \alpha_{ijc} \geq 3 \sum_{rk} t_{irk} \quad \forall i \in L, c \in C \tag{5}$$

$$\alpha_{ijc} \leq \sum_{rk} t_{irk} \quad \forall i \in L, j \in L, c \in C \tag{6}$$

While this model properly describes the problem, it is far too complex even with a few numbers of lights. This can easily be seen by looking at constraints (2-4) and (6). The number of such constraints is quadratic in the maximum number of possible light sources.

Our next approach, named the constraint generation approach, tries to alleviate this problem by replacing these constraints with a much simpler one. Instead of covering the whole field, the main model aims to cover an initially

small set of points P spread across the field. The binary variable β_{ip} is 1 if light source i covers point p, 0 otherwise. The constraints (7-8), to set β_{ip}, are very similar to constraints (2-3) in the single model:

$$xc_{j1} - (1 - \beta_{ip})M \leq xc_{ic} \quad \forall i \in L, p \in P \tag{7}$$

$$xc_{ic} \leq xc_{j3} + (1 - \beta_{ip})M \quad \forall i \in L, p \in P. \tag{8}$$

This way (7-8) are linear with the number of points in P, and (4-6) simplifies to the constraint (9 to cover every point in P with at least one light source,

$$\sum_{i \in L} \beta_{ip} = 1 \quad \forall p \in P. \tag{9}$$

This model is similar to the rectangular p-center problem Drezner solved in [1], although more difficult due to the different properties of the covering rectangles and the restrictions on their location.

The new model is simpler, however by covering a finite set of points we may not ensure that the whole field will be covered. Thus, an additional model generates a new point to be included in P, one that was not covered by the solution of the main model. To find such a point a constraint generation model is constructed with the objective

$$\max \min_{i \in L} \|p - l_i\|_1 \tag{10}$$

where l_i denotes a rectangle and p is the new point. Namely, we want to find a point maximizing the Manhattan distance from the closest rectangle. If such a point cannot be found, then we have solved the main problem. Yet, as long as a new point can be generated, it is inserted into P, and we iterate on these two models. The choice of the new point heavily influences the speed of the algorithm.

RESULTS

We used AMPL to implement our models and CPLEX 12.6.3 to solve them. The tests were run on a machine with 2 GB of memory and a Core 2 Duo CPU. The execution times of CPLEX for the two approaches can be seen in Table 1 while the size of the reduced MIPs are shown in Table 2. The number of variables and constraints for the MIPs of the constraint generation approach are represented by their highest values from every iteration. The tests were run for different types of light sources $|K| = 2, 3, 5, 7$. The problems were generated by incrementally adding new types of light sources to the previous set K, with different attributes. It is worth mentioning that adding new light types can make a problem easier or harder depending on their attributes.

TABLE 1. Execution times in seconds for the two approaches

Number of types, $\|K\|$	2	3	5	7
Single model	132	271	1875	-
Constraint generation	31	559	1955	828

TABLE 2. Reduced MIP sizes for the two approaches

Number of types, $\|K\|$		2		3		5		7	
		#var.	#constr.	#var.	#constr.	#var.	#constr.	#var.	#constr.
Single model		311	1031	331	1031	371	1031	411	1031
Constraint generation	Main model	761	2511	471	1108	491	852	581	913
	CG model	65	135	65	132	65	135	65	35

While the execution times of the single model look good compared to the constraint generation approach, it is worth mentioning that the number of possible light sources was limited to the amount required for the optimal solution. Without this limitation, the single model could not be solved. While on the constraint generation approach we had the available number of light sources much higher. This means while the first model can find the optimal solution in the displayed time, it cannot verify that a better solution with higher number of light sources does not exist. However, the constraint generation approach can.

It is also interesting to note why the case with only 5 types of light sources was the hardest problem. In this case, adding new light types simplifies the problem for the $|K| = 7$ case. However, for the single model, the MIP's size of the 7-type-light problem became too big, hence it could not be solved in less than an hour.

A visualization of an optimal cover and two intermediate steps used by the constraint generation approach for $|K| = 7$ can be seen on Figure 2. The rectangles are the areas covered by the light sources, where darker areas are covered by multiple sources. Each light source is represented by a number with a dark background on the side of its rectangle. The dots are the points to be covered by the main model. Some of them are initially set, like the corners and middle point, while the rest is generated by the constraint generation model. The dot that is not yet covered by the rectangles is the newly created point, that must also be covered in the next iteration.

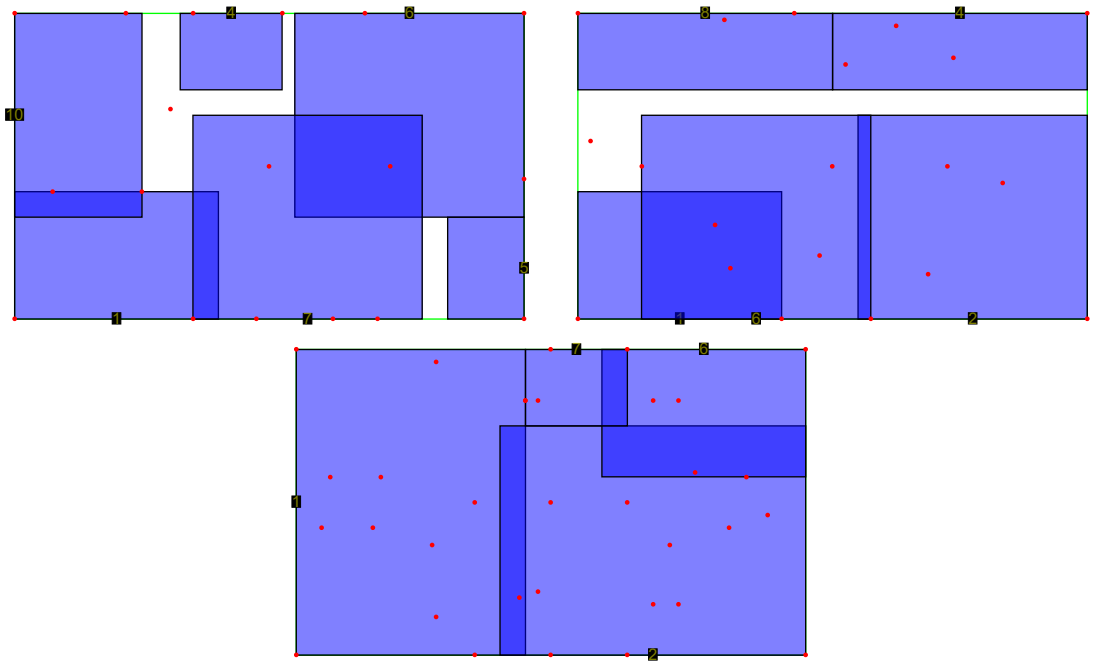

FIGURE 2. Visualization of two intermediate solutions and an optimal cover for the constraint generation approach

CONCLUSION

Two solution approaches are proposed to solve this specific covering problem. A single model solution that can only solve low-dimensional problems, due to the large size of the MIP. The second one is a constraint generation approach solving some smaller sized MIPs iteratively, hence it can solve larger problems. The solutions are satisfactory, although eliminating the symmetries of the problem would speed up the algorithm for both cases. As a future work, we would like to study the extension to have any kind of rotation of the light sources.

REFERENCES

[1] Zvi Drezner. On the rectangular p-center problem. *Naval Research Logistics (NRL)*, 34:229 – 234, 04 1987.

[2] D.S. Franzblau and D.J. Kleitman. An algorithm for covering polygons with rectangles. *Information and Control*, 63(3):164 – 189, 1984.

[3] Boglárka G.-Tóth and Kristóf Kovács. `http://math.bme.hu/~kkovacs/single-model.pdf`, 2018.

[4] Eric Huang and Richard E. Korf. Optimal rectangle packing: An absolute placement approach. *J. Artif. Int. Res.*, 46(1):47–87, 2013.