Canadian Science Publishing | Canadian Journal of Civil Engineering

Development of a Block Layout Optimization Model

Journal:	*Canadian Journal of Civil Engineering*
Manuscript ID	cjce-2025-0022.R1
Manuscript Type:	Research Article
Date Submitted by the Author:	16-Oct-2025

Complete List of Authors:	Urbino, Ivan Jan; Chung-Ang University, Department of Smart Cities Hwang, Hyunsik; Gil Solution Cho, Yoonho; Chung-Ang University, Department of Civil and Environmental Eng.

Is the manuscript for consideration in a Special Issue or Collection?:	Not applicable (regular submission)
Keyword:	Block layout Optimization, Harmony Search Algorithm, Cutting Loss

SCHOLARONE™ Manuscripts

Development of a Block Layout Optimization Model

Ivan Jan Urbino[a] , Hyunsik Hwang[b,*], YoonHo Cho[c]

[a] Department of Smart Cities, Chung-Ang University, Seoul 06974, Republic of Korea. Email address: iju202112080@cau.ac.kr
[b] GIL Solution, 315, Goyang-daero, Ilsanseo-gu, Goyang-si, Gyeonggi-do, Republic of Korea. Email address: hshwang@gilsolution.com
[c] Department of Civil and Environmental Engineering, Chung-Ang University, Seoul 06974, South Korea. E-mail address: yhcho@cau.ac.kr

* Corresponding author:
E-mail address: hshwang@gilsolution.com

Abstract

Block pavements involve the placement of rectangular blocks in an area with a certain pattern. However, blocks will need to be cut to fit the area which will yield additional costs and waste. This study presents a model to minimize cutting loss in block pavements for any shape, improving previous studies that only considered rectangular bounding areas and disregarded block placement patterns. The model mathematically incorporates constraints such as amount, geometric boundary, overlap, and pattern; and uses the harmony search algorithm to optimize block layouts for stack bond, stretcher, and herringbone patterns. Results demonstrated improved layouts compared to existing pavements. Sensitivity analysis showed that while the pattern does not affect the results, cutting loss decreases with smaller block sizes, larger pavement areas, or zero convexity boundaries. It also showed that optimizing the block layout was able to follow construction guidelines of allowing additional blocks to 8% of the estimated quantity.

Keywords

Block layout Optimization, Harmony Search Algorithm, Cutting Loss

1. Introduction

Interlocking block pavements are made with concrete or clay blocks being laid in patterns on a bedding sand layer between restraints (Mudiyono et al. 2006). Pavement blocks are used for low-speed highways, pedestrian zones, cycle paths, or parking lots due to the artificial irregular surface for speed restriction, permeability, decorative aesthetic, and ease of installation. One of the structural components is the various block patterns such as stack bond, basketweave, stretcher, and herringbone. These patterns can affect the distribution of compressive and shear stresses into the underlying layers (Jamshidi et al. 2019). Herringbone patterns are frequently used due to its performance in resisting horizontal creep caused by wheel friction (Abate 1993), however recent studies by Soutsos, et al. (2011) determined that all patterns (basketweave, stretcher and herringbone) have similar displacement under loading and Mampearachchi and Senadeera(2014) concluded that stretcher and herringbone pattern would have the least deflection than basketweave or stack bond pattern. This practice is similar to different other industries such as textile, tiling or solar paneling where a rectangular layout is necessary. This concept can also be extended to designing parking lots, where similar rectangular spaces for vehicles are allocated to a certain area on a certain arrangement (Mikusova et al. 2020).

Regardless of the pattern, cutting loss will always be present in most layouts since the blocks cannot always fit the allocated pavement area, particularly when there is an irregular shape. Cutting loss in pavement blocks yield additional waste and cost. The current design and construction guidelines in Seoul specify that additional quantities should be brought in within 8% of the estimated quantity to account for cutting or breakage during construction (Seoul Metropolitan Government and Seoul Facilities 2020). This could be the major cause that leads to excessive transportation and waste disposal cost. While this additional quantity is obtained empirically, there is no theoretical basis to determine the number of blocks necessary for a certain area.

Rectangular cutting stock problem is a 2-dimensional(2D) optimization that includes a supply of rectangular sheets, an order for a specified number of each certain type of rectangles and finding a way of cutting the shapes out of the sheets that minimizes the total wastes. 2D cutting stock problems are also referred to as packing problems since it can be viewed either as the problem of cutting smaller pieces from a larger stock rectangle or packing smaller pieces into a larger stock rectangle (Beasley 2004). Similarly, the rectangle packing problem aims to find the maximum number and layout of rectangles of a certain size that can fit a larger boundary, which is usually also rectangular in shape (Huang and Korf 2013). There have been several approaches to solve these problems such as branch and bound methods (Arenales and Morabito 1995) (Chan and Markov 2004), greedy algorithm, corner fit, deepest fit, hybrid corner/deepest fit, modified best fit heuristics (To 2006), depth first search (Dowsland 1987) and using machine learning algorithms (Fang et al. 2023) (Pitombeira-Neto and Murta 2022). All of these methods have been used to solve this optimization problem within rectangular boundaries.

The main limitation those literatures is that the bounding polygon for the smaller rectangles is also a rectangular shape, and it does not consider if the smaller rectangles have to be placed in a certain pattern, which is a normal practice in the design of block pavements. A model was presented by Urbino et al. (2023) to be able to overcome this problem by using harmony search algorithm(HSA) to determine the optimum layout of blocks on a certain pattern to minimize the cutting loss on an enclosing polygon with the top edge approximated as a mathematical function while a horizontal base. This paper aims to improve their study by developing a mathematical model to be able to optimize the block placement on any pavement shape with just the coordinates of its vertices using different patterns (stack bond, stretcher and 90° herringbone). The model could also be used to estimate the amount of blocks needed to be placed on a certain area and to verify whether the additional 8% quantity set by the guidelines is feasible.

2. Problem Formulation

Given a general pavement area bounded by points $(x_{i,A},y_{i,A})$in a clockwise order. For the pavement blocks, its length and width is denoted by l and w respectively, while its position is indicated by $(x_{k,B},y_{k,B})$ or the center of the block that will be placed in the pavement area. $x_{min,A}$ and $x_{max,A}$ represents the smallest and largest x-dimension of the pavement area respectively and $y_{min,A}$ and $y_{max,A}$ for the y-dimension. Figure 1 shows the allocation of a sample block k in the pavement. Joint thickness in between blocks can be considered by adding half its value to l and w. The total amount of pavement boundary points and blocks that are placed in the pavement can be notated by n and m respectively.

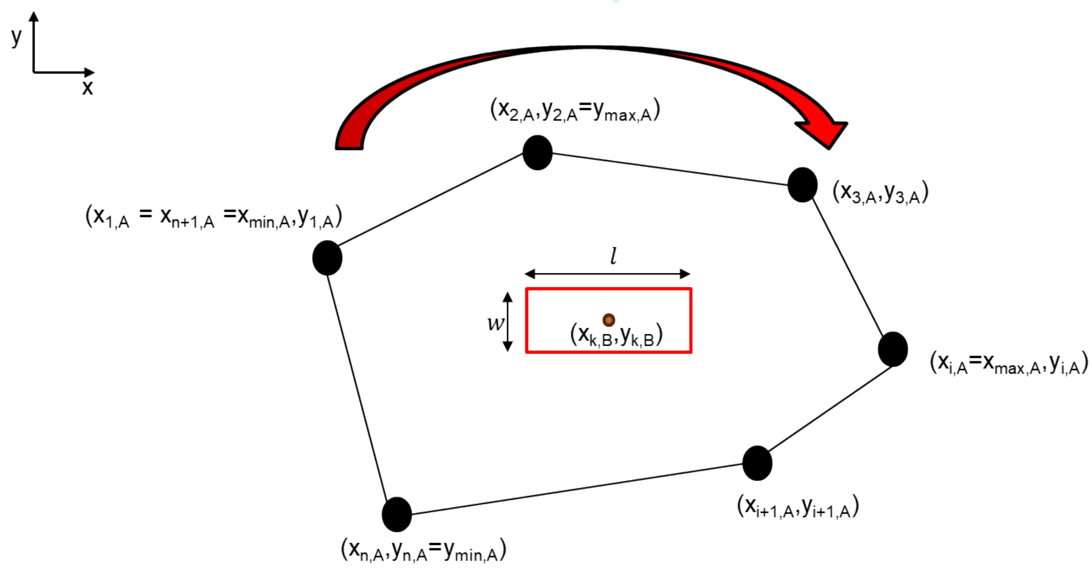


Figure 1 . Block k placed in the pavement bounded by points $x_{1,A},\ldots,x_{n,A}$

The main objective of this problem is to minimize the cutting loss or the area around the pavement area not covered by a whole block. The objective function can be written as in equation (1). As shown in figure 2, blocks that can be fully placed inside the area are shown in gray color while blocks that need to be cut to fit inside are shown in shaded area. The total area of the shaded area is the difference of the pavement area to the total area of full blocks inside.

(1) $minimize\ Z = A_0 - l * w * m$

where,

A_o is the total area of the pavement

m is the total number of blocks used

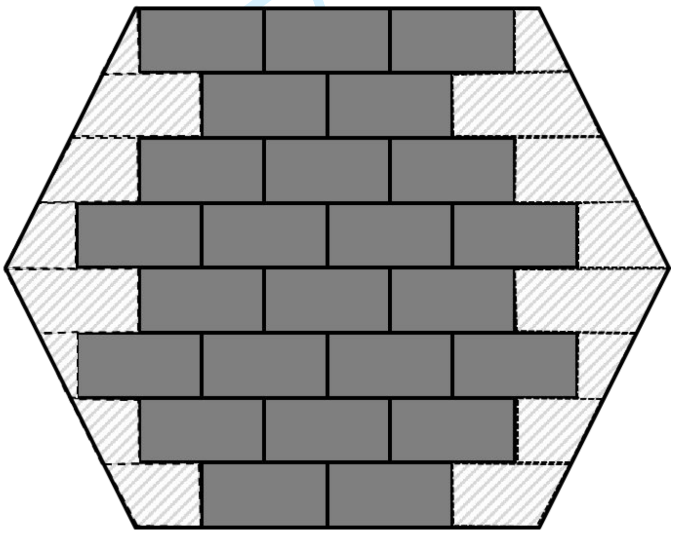


Figure 2. Cutting loss area described as shaded blocks

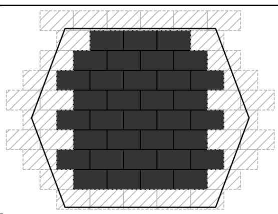
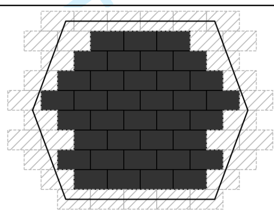
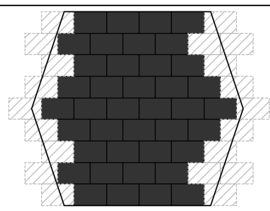
The optimization model for minimizing cutting loss in block pavements are subjected to several constraints. An amount constraint must be defined to have an allowable range on the quantity of blocks that can be placed. A geometric boundary constraint is necessary to ensure that blocks are fully placed within the boundary of the pavement area, while an overlap constraint should be considered to prevent blocks from overlapping and maintain the independence of each placement. Lastly, a pattern constraint must be defined to ensure that the blocks are arranged according to specified placement patterns, such as stack bond, stretcher, or herringbone.

2.1. Amount Constraint

Since the number of blocks placed on the pavement is not known, there must be a range on the allowable number of blocks. The number of blocks can change based on the layout or pattern used. The amount constraint or the maximum allowable number of blocks (m) is given in equation (2). For example, given a pavement area of 1m^2, the maximum number of 20cm by 10cm blocks that can be placed is 50, however, depending on how the blocks are placed even with the same pattern, the number of blocks can vary as shown in table 1.

(2) $$m \leq \frac{A_o}{l*w}, \quad \forall m \in \mathbb{Z} \geq 0$$

Table 1 Varying number of whole blocks that can be placed depending on layout

Block Layout			
Number of whole blocks placed (m)	34	36	40

2.2. Geometric Boundary Constraint

The blocks must also be within the pavement area and no part of the block must be outside of the boundary. For the pavement block to be within the boundary, all four corner points of the block must be on or inside of the boundary and all block edges should not intersect with the pavement boundary. A method was determined if a point lies on the interior of a polygon, where if a point is extended eastward and it intersects the polygon odd number of times then it is considered inside and otherwise it is outside (Bourke 1997). This method is effective since it can also consider non-convex polygons. The intersection of 2 segments (block edge and polygon edge) can be determined by its orientation (Gavrilova and Rokne 2000). Equations(3)~(8) show what conditions needs to be satisfied in order for a block (x_{kB},y_{kB}) to be considered completely inside the boundary.

Figure 3 shows how these equations are applied to an example block. Equations (3) determines that when the top edge of the block is extended eastward it would intersect with the boundary and equation (9) would ensure that it would intersect with the boundary odd number of times, while equation (4) is for bottom edge of the block. Equations (5)~(8) ensures that all the edges of the block does not intersect with any of the geometric boundary.

$$(3)\quad \left\{-\left[y_{i,A}-\left(y_{k,B}+\frac{w}{2}\right)\right]*\left[x_{max,A}-x_{min,A}\right]\right\}*\left\{-\left[y_{i+1,A}-\left(y_{k,B}+\frac{w}{2}\right)\right]*\left[x_{max,A}-x_{min,A}\right]\right\}*b_i\leq 0,\quad i=1,\ldots,n$$

$$(4)\quad \left\{-\left[y_{i,A}-\left(y_{k,B}-\frac{w}{2}\right)\right]*\left[x_{max,A}-x_{min,A}\right]\right\}*\left\{-\left[y_{i+1,A}-\left(y_{k,B}-\frac{w}{2}\right)\right]*\left[x_{max,A}-x_{min,A}\right]\right\}*b_i\leq 0,\quad k=1,\ldots,m$$

$$(5)\left\{\left[y_{i+1,A}-y_{i,A}\right]*\left[\left(x_{k,B}-\frac{l}{2}\right)-x_{i+1,A}\right]-\left[\left(y_{k,B}+\frac{w}{2}\right)-y_{i+1,A}\right]*\left[x_{i,A}-x_{i+1,A}\right]\right\}*$$
$$\left\{\left[y_{i+1,A}-y_{i,A}\right]*\left[\left(x_{k,B}-\frac{l}{2}+x_{max,A}-x_{min,A}\right)-x_{i+1,A}\right]-\left[y_{i,A}-y_{i+1,A}\right]*\left[\left(x_{k,B}-\frac{l}{2}+x_{max}-x_{min}\right)-x_{i+1,A}\right]\right\}*b_i\leq 0$$

$$(6)\quad \left\{\left[y_{i+1,A}-y_{i,A}\right]*\left[\left(x_{k,B}+\frac{l}{2}\right)-x_{i+1,A}\right]-\left[\left(y_{k,B}+\frac{w}{2}\right)-y_{i+1,A}\right]*\left[x_{i,A}-x_{i+1,A}\right]\right\}*$$
$$\left\{\left[y_{i+1,A}-y_{i,A}\right]*\left[\left(x_{k,B}+\frac{l}{2}+x_{max,A}-x_{min,A}\right)-x_{i+1,A}\right]-\left[y_{i,A}-y_{i+1,A}\right]*\left[\left(x_{k,B}+\frac{l}{2}+x_{max,A}-x_{min,A}\right)-x_{i+1,A}\right]\right\}*b_i\leq 0$$

$$(7)\left\{\left[y_{i+1,A}-y_{i,A}\right]*\left[\left(x_{k,B}+\frac{l}{2}\right)-x_{i+1,A}\right]-\left[\left(y_{k,B}-\frac{w}{2}\right)-y_{i+1,A}\right]*\left[x_{i,A}-x_{i+1,A}\right]\right\}*$$
$$\left\{\left[y_{i+1,A}-y_{i,A}\right]*\left[\left(x_{k,B}+\frac{l}{2}+x_{max,A}-x_{min,A}\right)-x_{i+1,A}\right]-\left[y_{i,A}-y_{i+1,A}\right]*\left[\left(x_{k,B}+\frac{l}{2}+x_{max,A}-x_{min,A}\right)-x_{i+1,A}\right]\right\}*b_i\leq 0$$

(8)

$$\left\{\left[y_{i+1,A}-y_{i,A}\right]*\left[\left(x_{i,B}-\frac{l}{2}\right)-x_{i+1,A}\right]-\left[\left(y_{i,B}-\frac{w}{2}\right)-y_{i+1,A}\right]*\left[x_{i,A}-x_{i+1,A}\right]\right\}*$$
$$\left\{\left[y_{i+1,A}-y_{i,A}\right]*\left[\left(x_{i,B}-\frac{l}{2}+x_{max,A}-x_{min,A}\right)-x_{i+1,A}\right]-\left[y_{i,A}-y_{i+1,A}\right]*\left[\left(x_{i,B}-\frac{l}{2}+x_{max,A}-x_{min,A}\right)-x_{i+1,A}\right]\right\}*b_i\leq 0$$

$$(9)\quad \sum_{i=1}^{n} b_i-2w=1,\quad \forall w\in\mathbb{Z}\geq 0$$

$$(10)\quad b_i\in\{0,1\},\quad i=1,\ldots,n$$

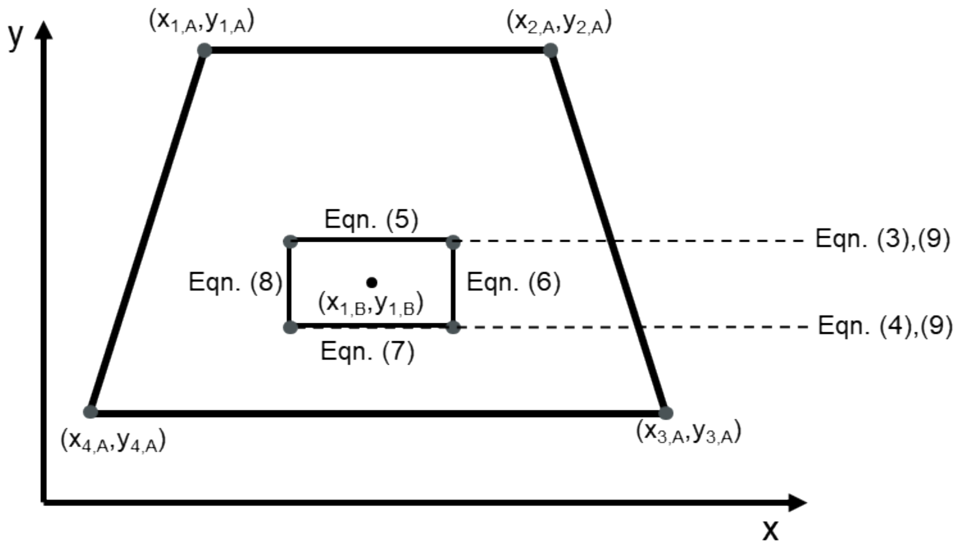


Figure 3. Visualization of Geometric Boundary Constraint

2.3. Overlap Constraint

Logically, any two blocks must not overlap with one another. The difference between the x-coordinates and y-coordinates of the blocks should be at least greater than or equal to the length and width respectively. At least one or both the dimensions (x or y) must not overlap. Equations (11)-(14) show the overlap constraint with a variable M denoting a very large number. As shown in figure 4, when the blocks are next to each other, the y-coordinate will have the same value, but the x-coordinate will have at least l distance apart and vice versa when the blocks are on vertically on top of each other.

(11) $|x_{i,B} - x_{j,B}| + Mb_1 \geq l \qquad i = 1, \ldots, m, \quad j = 1, \ldots, m, \quad i \neq j$

(12) $|y_{i,B} - y_{j,B}| + Mb_2 \geq w$

(13) $b_1 + b_2 \leq 1$

(14) $b_1, b_2 \in (0,1)$

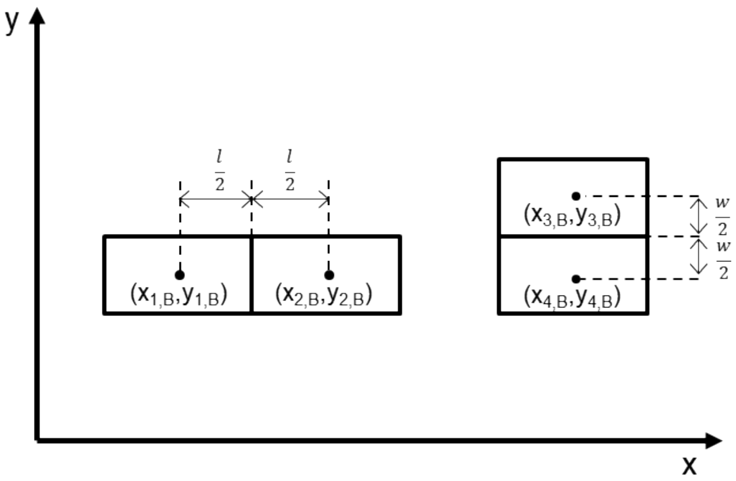


Figure 4. Overlap Constraint

2.4. Pattern Constraint

For the blocks to have a certain pattern on the pavement, a pattern constraint needs to be present where the positions of the blocks relative to one another are fixed. There are three patterns presented which are: Stack Bond Pattern, Stretcher Bond Pattern, and the 90° Herringbone Pattern.

2.4.1. Stack Bond Pattern

The stack bond pattern is one of the simplest patterns since it is just simply an organized row and columns of blocks. This pattern can also be considered as a simplification of the basketweave pattern since the basketweave pattern shows a repetition of the same pattern organized the same way as the stack bond pattern as shown in figure 5. Equations (15)-(19) show the constraint for this type of pattern. As shown in figure 6, for a block with a length and width of l and w respectively the difference between the x-coordinates and y-coordinates of any two blocks would be multiples of l and w respectively as shown in equations (15) and (16), and the maximum amount of blocks that can be placed in a row and column is shown in equations (17) and (18) respectively, and in this specific example the value of p and q is 2 and

3 respectively. Both p and q cannot have a value of 0, because then the coordinates for 2 blocks would be the same as shown in equation (19).

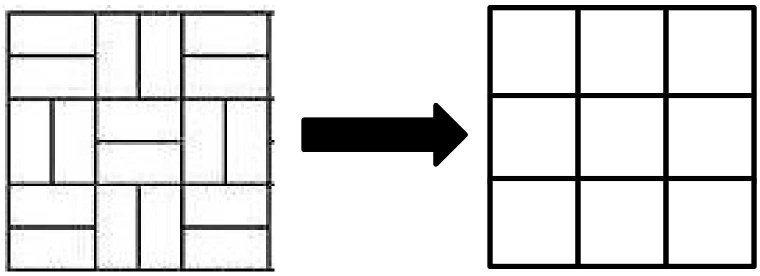


Figure 5 Simplified basketweave pattern as stack bond pattern

(15) $|x_{i,B} - x_{j,B}| = l * p \qquad i = 1, \ldots, m, \quad j = 1, \ldots, m, \quad i \neq j$

(16) $|y_{i,B} - y_{j,B}| = w * q$

(17) $0 \leq p \leq \left(\frac{x_{max,A} - x_{min,A}}{l}\right)$

(18) $0 \leq q \leq \left(\frac{y_{max,A} - y_{min,A}}{w}\right)$

(19) $p + q > 0, \quad p, q \in \mathbb{Z} \geq 0$

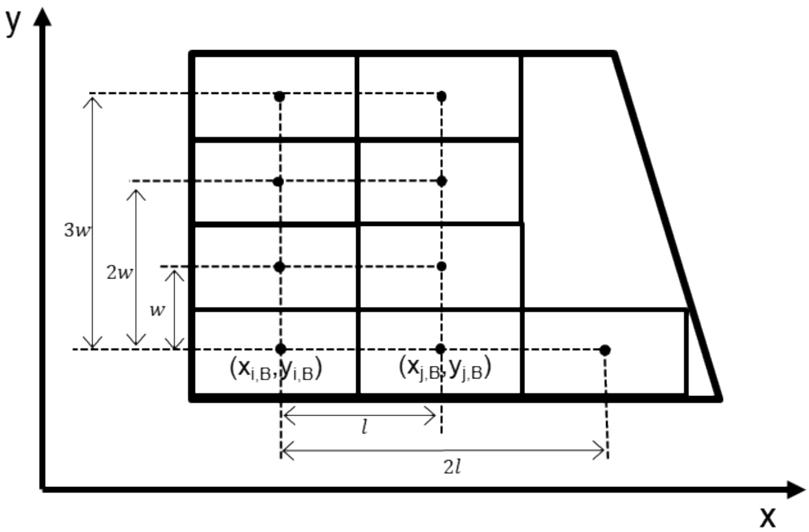


Figure 6. Visualization of Stack Bond Pattern Constraint

2.4.2. Stretcher Pattern

The stretcher pattern is created by laying a row of blocks then followed by a second row of blocks laid with an offset by a half of block. This pattern continues until the entire pavement area is filled. Figure 7 shows a typical stretcher bond pattern. Equations (20)-(26) show this constraint. It has a similar formulation to the stack bond constraint with the x-axis being able to have an offset of half the length of the block when the two blocks have a difference in y-coordinates as an odd multiple of w as shown in equation (20) and figure 8. Additionally, if the difference of y-coordinates is an even multiple of the width, then the difference of x-coordinates are also an even multiple of half the length and vice versa when the coordinates are odd multiples of l and w. This property is represented by the addition of b_1 and b_2, if $b_1 = 1$ and $b_2 = 0$, then both p and q would be even while if $b_1 = 0$ and $b_2 = 1$, then both p and q would be odd.

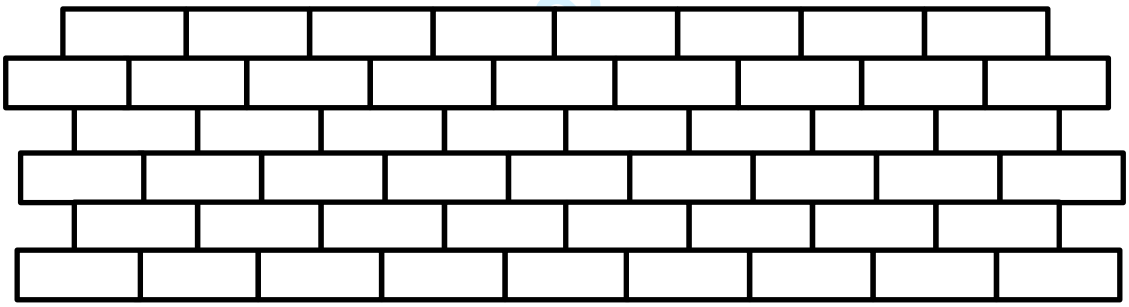


Figure 7 Stretcher Bond Pattern

(20) $|x_i - x_j| = 0.5 * l * p, \quad i = 1, \dots, m, \quad j = 1, \dots, m, \quad i \neq j$

(21) $|y_i - y_j| = w * q$

(22) $0 \leq 2p * b_1 + (2p + 1) * b_2 \leq \left(\frac{x_{max,A} - x_{min,A}}{l}\right)$

(23) $0 \leq 2q * b_1 + (2q + 1) * b_2 \leq \left(\frac{y_{max,A} - y_{min,A}}{w}\right)$

(24) $b_1 + b_2 = 1$

(25) $p + q > 0, \quad p, q \in \mathbb{Z} \geq 0$

(26) $b_1, b_2 \in (0,1)$

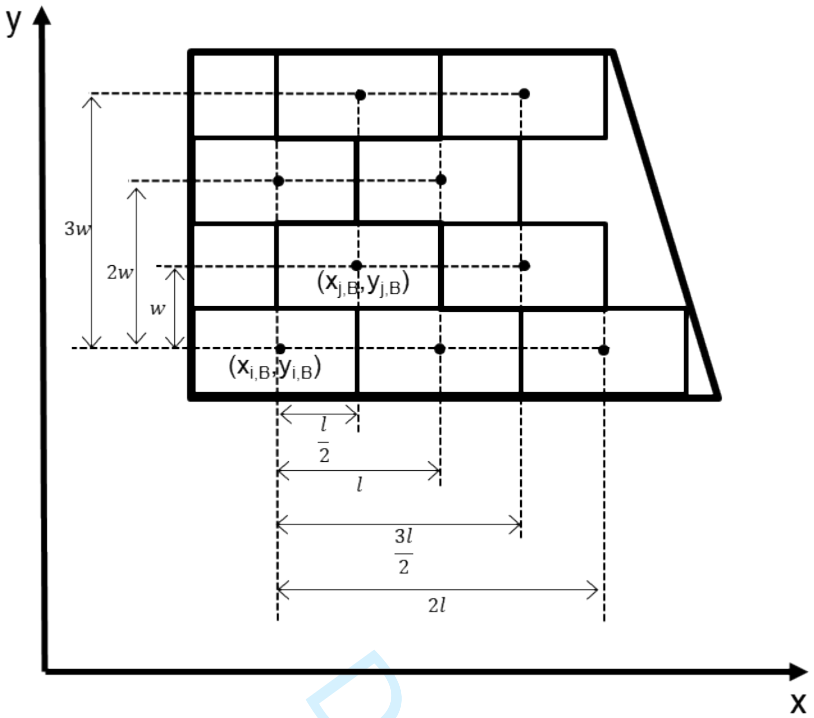


Figure 8. Visualization of Stretcher Pattern

2.4.3. Herringbone Pattern

Lastly, the herringbone pattern is made from horizontal and vertical blocks arranged in a staggered zig-zag pattern. The distinct characteristic of herringbone is that one rectangle is placed precisely so that the horizontal of one block meets to the vertical side of the other block and vice versa. Specifically, the 90° Herringbone model or simply called the Herringbone pattern have blocks parallel and perpendicular to the edge. Figure 9 shows a typical Herringbone pattern.

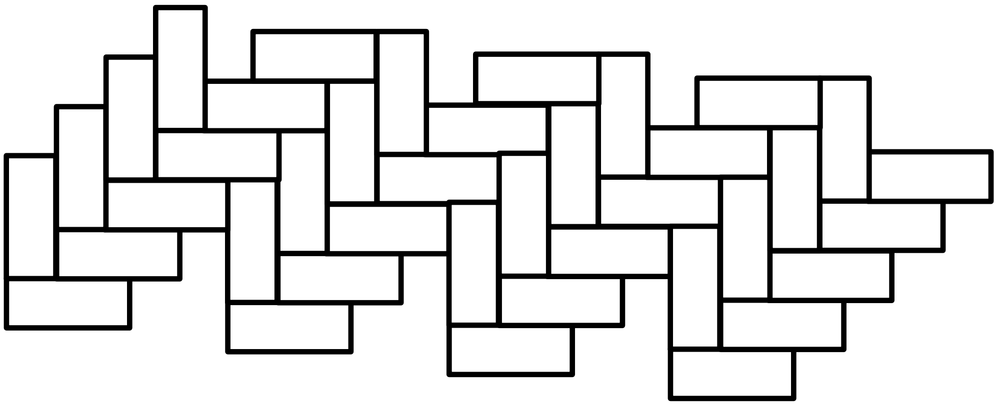


Figure 9. 90° Herringbone Pattern

Due to the presence of both horizontal and vertical blocks, there must be an additional constraint to allow for the rotation. Equations (27)-(32) show the orientation constraint. In this constraint L_i and W_i refers to the x-dimension and y-dimension of the block when it is placed in the pavement area. For example, when $b_{l1} = 1, b_{w1} = 0, b_{l2} = 0, b_{w2} = 1$, then the value of L_i and W_i is l and w respectively or the block is placed horizontally, and when $b_{l1} = 0, b_{w1} = 1, b_{l2} = 1, b_{w2} = 0$, the value of L_i and W_i is w and l respectively or the block is placed vertically as shown in Figure 10.

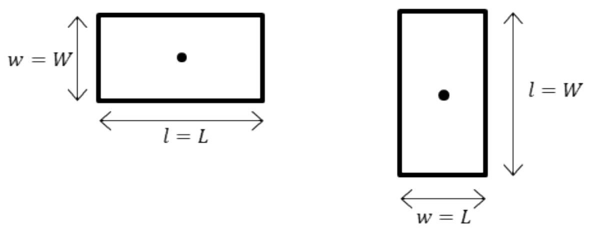


Figure 10. Visualization of orientation constraint

These notations for the length and width are used for the herringbone pattern constraint in equations (33)-(41). As shown in Figure 11, the blocks with the same orientation (when $b_1 = 1$ and $b_2 = 0$) either horizontal blocks or vertical blocks, the distance between the x-coordinates of any same orientation block is a sum of the multiples of the width and length while the distance of its y-coordinates is the difference of the multiples of width and length.

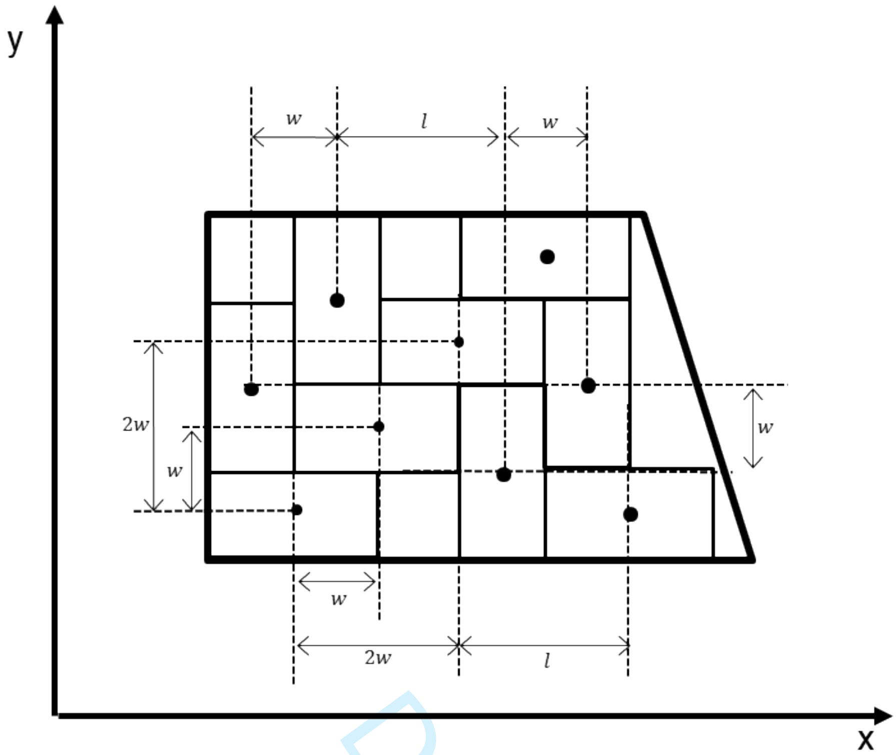


Figure 11. Relationship of blocks with similar orientation

As shown in figure 12, the blocks with different orientations (when $b_1 = 0$ and $b_2 = 1$), the difference between the x-coordinates is a sum of the odd multiples of half the width and length while the difference of its y-coordinates is the difference of the odd multiples of half the width and length.

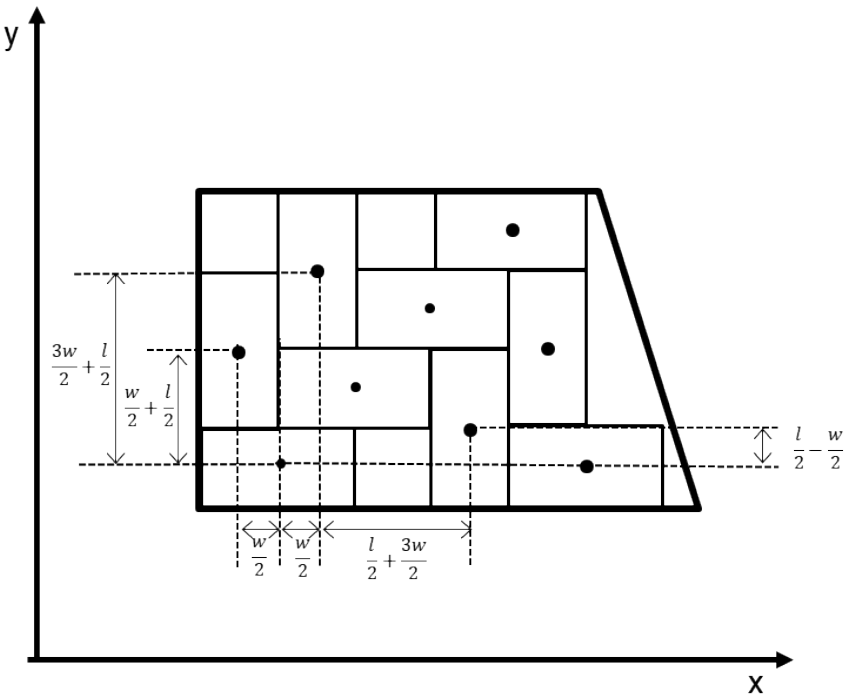


Figure 12. Relation of blocks with different orientation

(27) $L_i = l * b_{l1} + w * b_{w1}$

(28) $W_i = l * b_{l2} + w * b_{w2}$

(29) $b_{l1} + b_{w1} = 1$

(30) $b_{l2} + b_{w2} = 1$

(31) $b_{l1} + b_{l2} = 1$

(32) $b_{w1} + b_{w2} = 1$

(33) $|x_{i,b} - x_{j,b}| = [p_1 * w + q_1 * l] * b_1 + \left[(2p_3 + 1) * \left(\frac{w}{2}\right) + (2q_3 + 1) * \left(\frac{l}{2}\right)\right] * b_2, \quad i = 1, \dots, m, \quad j = 1, \dots, m, \quad i \neq j$

(34) $|y_{i,b} - y_{j,b}| = [p_2 * w - q_2 * l] * b_1 + \left[(2p_3 + 1) * \left(\frac{w}{2}\right) - (2q_3 + 1) * \left(\frac{l}{2}\right)\right] * b_2$

(35) $p_1 * q_1 \geq 0$

(36) $p_2 * q_2 \leq 0$

(37) $(L_i - L_j) * M \geq b_2$

(38) $(L_i - W_j) * M \geq b_1$

(39) $b_1 + b_2 = 1$

(40) $b_{l1}, b_{l2}, b_{w1}, b_{w2}, b_1, b_2 \in (0,1)$

(41) $p_1, p_2, p_3, q_1, q_2, q_3 \in \mathbb{Z} \geq 0$

2.4.4. Optimization Solution Algorithm

This problem can be formulated as a function optimization problem because of the addition of the pattern constraint. For a particular pattern, when a single block is placed in the area, the entire layout is known, including the cutting loss. Therefore, similar to a function optimization problem, the process involves determining the location of a single block that would result for the entire layout to yield a minimum cutting loss. Due to the complexity and the non-linearity of the model, population-based meta-heuristic algorithms would be used to solve this problem. It is an approach that searches for near-optimal solutions by having a population of candidate solutions and using population characteristics to guide the search iteratively (He et al. 2018). Also, these population-based methods outperform traditional methods when the problems are of high complexity (Innocente and Sienz 2021). It was already determined by Urbino et al. (2023) that using Harmony Search Algorithm (HSA) is able to solve this model efficiently and with a faster computing time. HSA is a meta-heuristic optimizer inspired by the music improvisation process. In the music improvisation process, a predefined number of musicians attempt to tune the pitch of their instruments to achieve a pleasing harmony. The aim is to find the global optimum of the objective function under consideration by tuning a predefined number of decision variables. (Askarzadeh and Rashedi 2017)

The steps in the HSA are as follows: (1) Initialize the problem and algorithm parameters, (2) Initialize the harmony memory, (3) Improvise a new harmony, (4) Update the harmony memory, and (5) Check the stopping criterion. In the first step, the objective function and decision variables are defined, as well as the algorithm parameters such as the harmony memory size (HMS), or the number of solution vectors in the harmony memory; harmony memory considering rate (HMCR); pitch adjusting rate (PAR); and the number of improvisation (NI) or the stopping criterion which is usually defined in the amount of iterations to be done. Next, the harmony memory (HM) is the memory location where all the solution vectors or the sets of decision variables are stored, this is similar to the genetic pool in the genetic algorithm (GA). The HM is filled with as many randomly generated solution vectors as the HMS. Then, a new harmony vector, $x' = (x'_1, x'_2, \ldots, x'_n)$ is generated based on the HMCR, PAR and random selection. In the memory consideration, the value of the first decision variable (x'_1) for the new vector is chosen from any of the values in the specified HM range. Values for the other decision variables are chosen similarly. The HMCR, which varies from 0 to 1, is the rate of choosing one value from the historical values stored in the HM, while (1-HMCR) is the rate of randomly selecting one value from the possible range of values. When a value is obtained by the memory consideration, it would be examined if it should be pitch-adjusted which is determined by the PAR parameter. The PAR parameter also ranges from 0 to 1, where it is the probability of adjusting the memory consideration. The pitch adjustment would be done as follows:

(42) $x'_i \leftarrow x'_i \pm rand() * bw,$

where,

bw is an arbitrary distance bandwidth

rand() is a random number between 0 and 1

Next, when updating the harmony memory, the new harmony vector is compared against the worst harmony in the HM based on the objective value. If the new vector has a better objective value, then it replaces the current existing worst harmony, otherwise nothing is done. Lastly, the stopping criterion or the maximum number of iterations is checked if satisfied and computation is terminated. Otherwise, steps 3 and 4 are repeated. (Mahdavi et al. 2007)

3. Illustrative Example

To show the use of the model and algorithm, an example from an actual block pavement is shown in figure 13 with the pavement boundary approximated as vertices shown in table 2. It has a basketweave pattern in an irregular pavement surface with the largest horizontal and vertical dimensions of 3.54m and 2.72m respectively. The blocks used have length and width of 0.20m and 0.10m respectively with joint thickness of 5mm. The model is used to determine the optimum layout and compare it to the current one. The parameters used in the HSA is HMS=10, HMCR=0.9, PAR=0.5, and bw=0.2, with a stopping criterion of 100 iterations. These parameters were chosen based on initial runs of the algorithm that would balance both computation time and the difference between the cutloss area from the initial to final iteration which shows the improvement of cutting loss throughout the iterations (Urbino et al. 2023). The actual cutloss is determined to be $0.665m^2$.

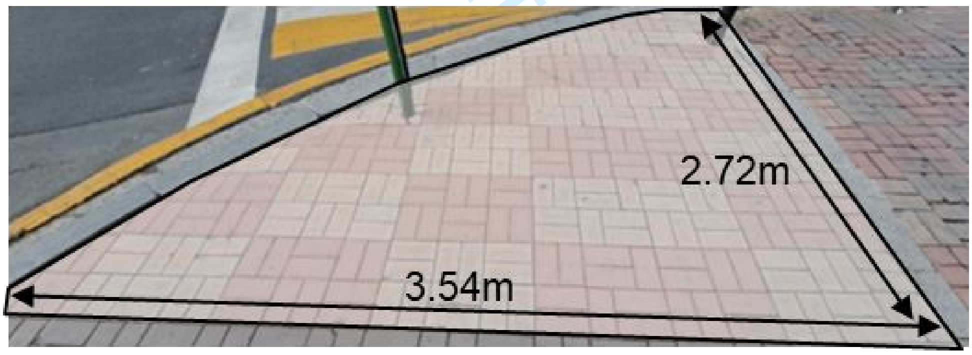
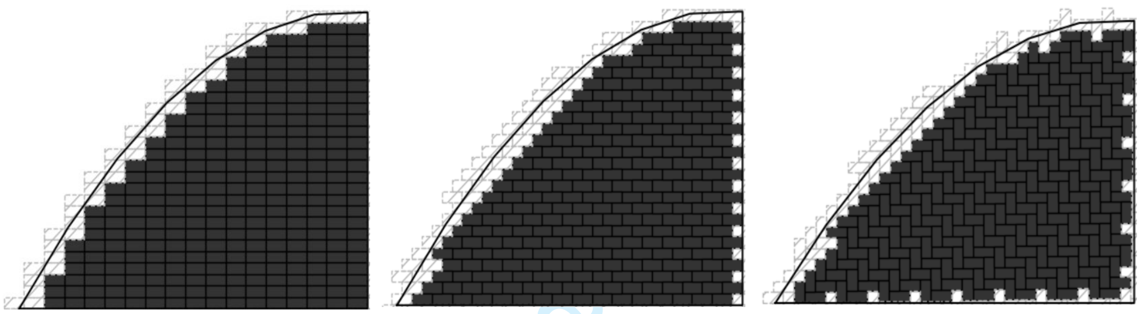


Figure 13. Actual Pavement Layout

Using the same pavement area and blocks, the best layout using the proposed model is shown in Figure 14. The blocks that can be fully placed in the boundary are shown in full color while blocks that need to be cut to fit the area or cut blocks are shaded. Using either a stack bond pattern or a stretcher pattern would result in a better layout with stack bond pattern would have the least cutloss area of $0.517m^2$ or a decrease of 33% from the actual layout as shown in table 2, furthermore, the number of cut blocks placed is decreased which would save on costs and time.

Table 2. Comparison of cutloss area between the actual layout and model results

	Actual Layout	Computed Layout		
	Basketweave Pattern	Stack Bond Pattern	Stretcher Pattern	Herringbone Pattern
Cutloss Area (m^2)	0.665	0.517	0.569	0.775
Number of cut Blocks placed	53	42	47	61



(a) (b) (c)

Figure 14. Optimized layout results using (a) stack bond pattern, (b) stretcher pattern, and (c) herringbone pattern

To show the effectiveness of the algorithm used to solve the problem, Figure 15 shows the convergence of the results of over 100 iterations when using the stack bond pattern. The algorithm was able to converge the cutting loss to a minimum value and consequently obtain a result with a decrease of 0.15m^2 in the cutting loss from the initial iteration.

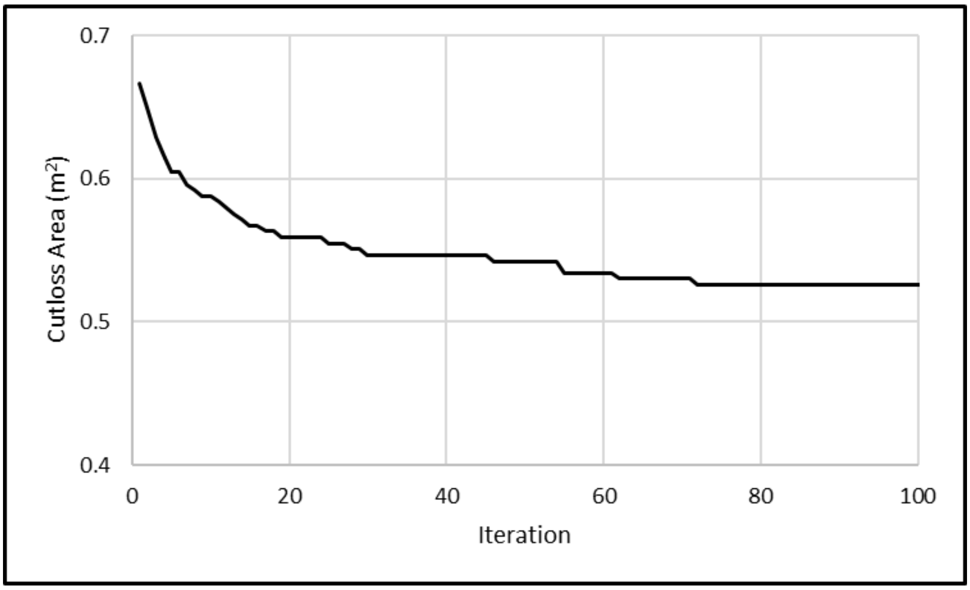


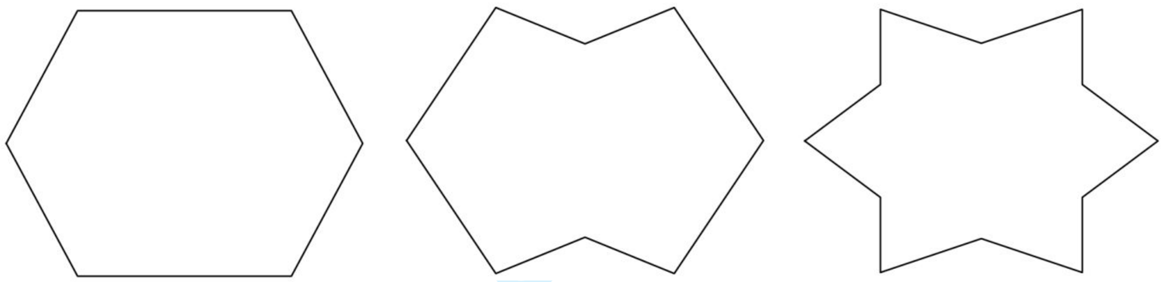
Figure 15. Minimization convergence of cutting loss for stack bond pattern

4. Sensitivity Analysis

To determine the applicability of this model, a sensitivity analysis is done by varying the pavement area, block size, block patterns and convexity. Convexity is described based on the ratio of the area of the polygon and the convex hull surrounding it (Brinkhoff et al. 1995). The convex hull is the smallest non-convex polygon that can contain the polygon. Table 3 describes the different parameters and values that were used. Overall, there were only 216 cases considered when the model was used since there is no herringbone pattern possible when the block size is 10cm by 10cm due to its symmetry of edge length. The pavement boundary shape is shown in figure 16 with different convexities with the same convex hull.

Table 3. Sensitivity Analysis Parameters

Pavement Area (m^2)	Block size (Rectangular)	Block Pattern	Convexity of Polygon
1	30cm by 10cm	Stack	0
5	20cm by 10cm	Stretcher	0.10
10	10cm by 10 cm	Herringbone	0.25
15			
20			
40			
80			
120			
200			



(a) (b) (c)

Figure 16. Pavement Boundary with (a) Convex=0, (b) Convex=0.1, (c) Convex=0.25

4.1. Results and Discussion

4.1.1. Effect of Block Pattern

Figure 17 shows how the different block patterns affected the relative cutting loss in the pavement. The relative cutting loss is the ratio of the cut loss area to the entire pavement area. Most of the cases had a similar result of around 0~5% of the pavement area. Using a one-way ANOVA with a significance value of α =0.05 showed no significance among the different patterns. Since there was no change in the block size and only the arrangement of the blocks was changed, it is unlikely that there would be substantial changes in the results.

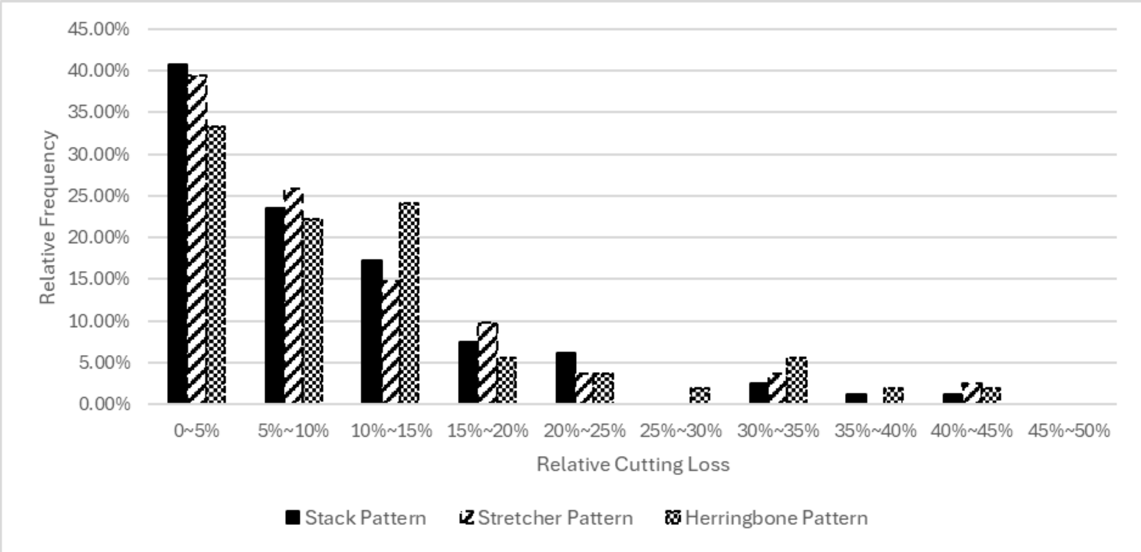


Figure 17 Frequency distribution of relative cutting loss based on block pattern

4.1.2. Effect of Block Size

Figure 18 shows how the different blocks sizes affect the cutting loss. There is a significant difference among the block sizes, where smaller blocks would have smaller cutting loss which was verified using one-way ANOVA with a significance value of α = 0.05. As shown in equation (2) with a smaller block area, the maximum number of blocks that can be placed, m, on the pavement would increase and therefore the cutting loss would decrease. This is also supported by the work of Yuan et al. (2022), where using a node grids and branch and bound algorithm to determine the next placement of the box, with smaller size of the box being packed into a convex polygon there would be more possible places to put the box and therefore maximize the number of boxes that can be placed.

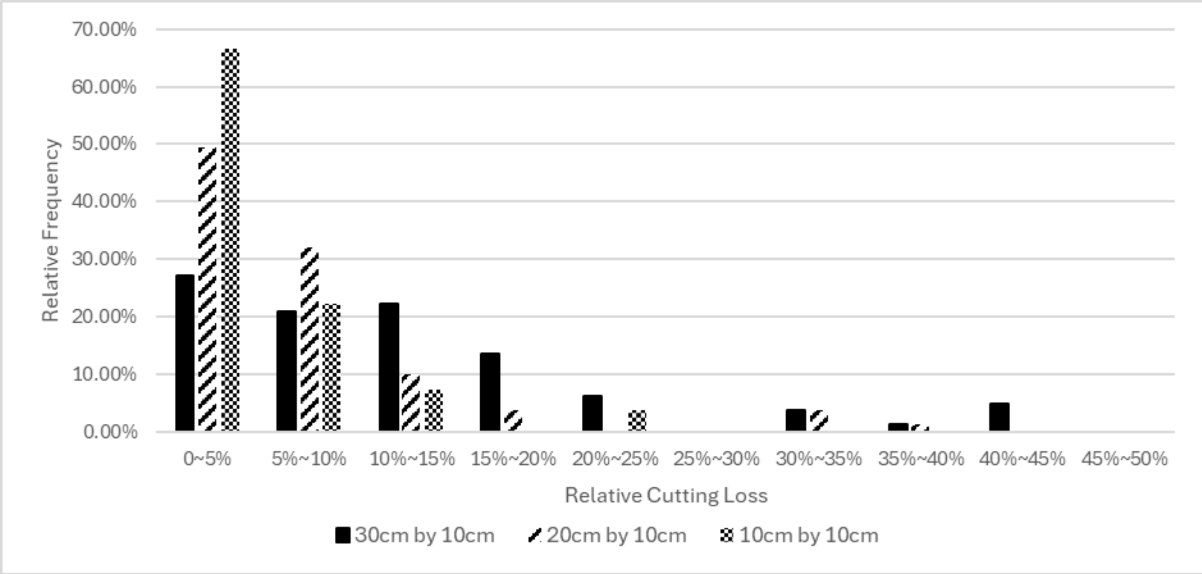


Figure 18. Frequency distribution of relative cutting loss based on block size

4.1.3. Effect of Pavement Area and Convexity

Performing a one-way ANOVA with a significance value of α = 0.05 have shown that both area and convexity have a significant effect on the cutting loss which can be clearly seen in figure 19. It could also be seen that using simple convex polygons would have a decrease in cutting loss, especially with smaller pavement areas, while the cutting loss is similar for both non-convex polygons. Since a higher convexity polygon would have a more irregular shape and a greater number of vertices for the same convex hull (Brinkhoff et al. 1995), the polygon would have more diagonal edges and a rectangular shape with horizontal and vertical edges would be harder to completely fit inside the polygon.

Additionally, the trend showed that as the pavement area increased, the relative cutting loss tends to converge up to less than 5% of the pavement area, which meant that with a higher pavement area, the relative cutting loss would be similar regardless of convexity.

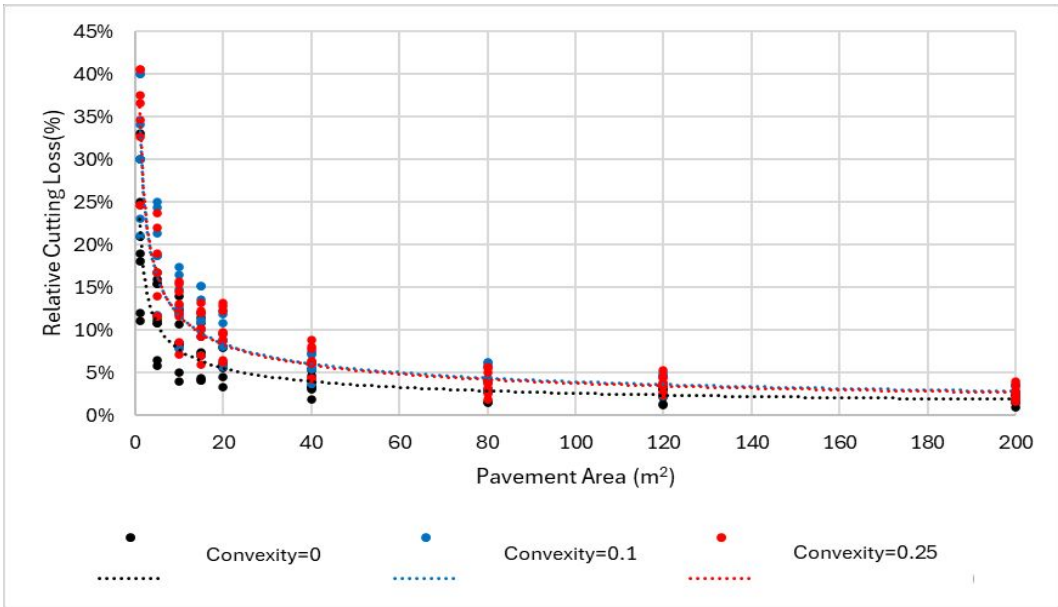


Figure 19. Effect of Pavement Area and Convexity on Cutting Loss

4.1.4. Relationship of cutting loss to amount of cut blocks

When analyzing the results, the cutting loss was mainly described. While there are many parameters to describe the layout such as number of full blocks placed inside the boundary, or the number of blocks that need to be cut. Since the main objective function is about minimizing the cutting loss, it is more important to describe the cutting loss itself and how the cutting loss relates to the other parameters such as cut blocks. A cut block is defined as a block that has either (1) intersected the boundary of the pavement area exactly two times, or (2) one to three of its corners lies inside or exactly on the boundary as shown in Figure 20. These conditions are defined by identifying cases where only some (but not all) of the boundary constraint equations (5)–(8) are satisfied, resulting in the edges or corners of the block intersecting the pavement boundary.

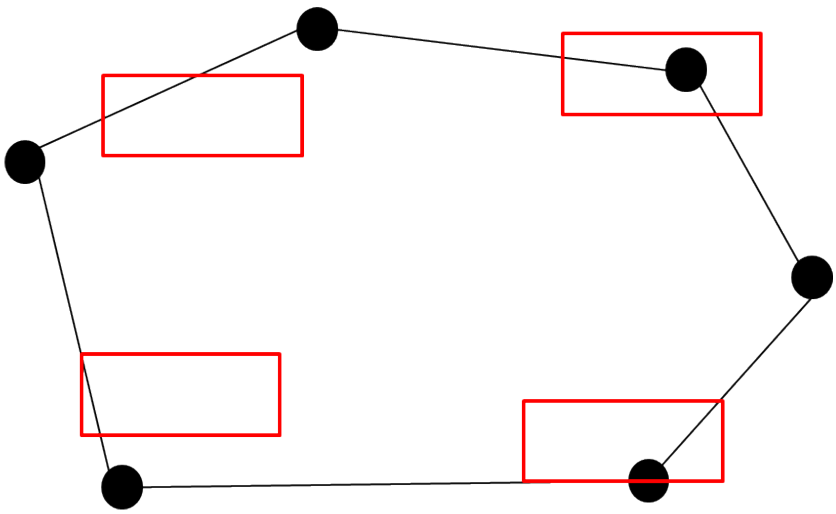


Figure 20. Definition of cut block

Figure 21 shows that overall, there is a linear relationship between the cutting loss and the amount of cut blocks placed. Also, with smaller block sizes, it is expected that there would be a smaller amount of cut blocks as previously discussed. Therefore, with a higher cutting loss, it is expected that there would be more cut blocks needed to place to fit the boundary area and it would increase costs and placement time.

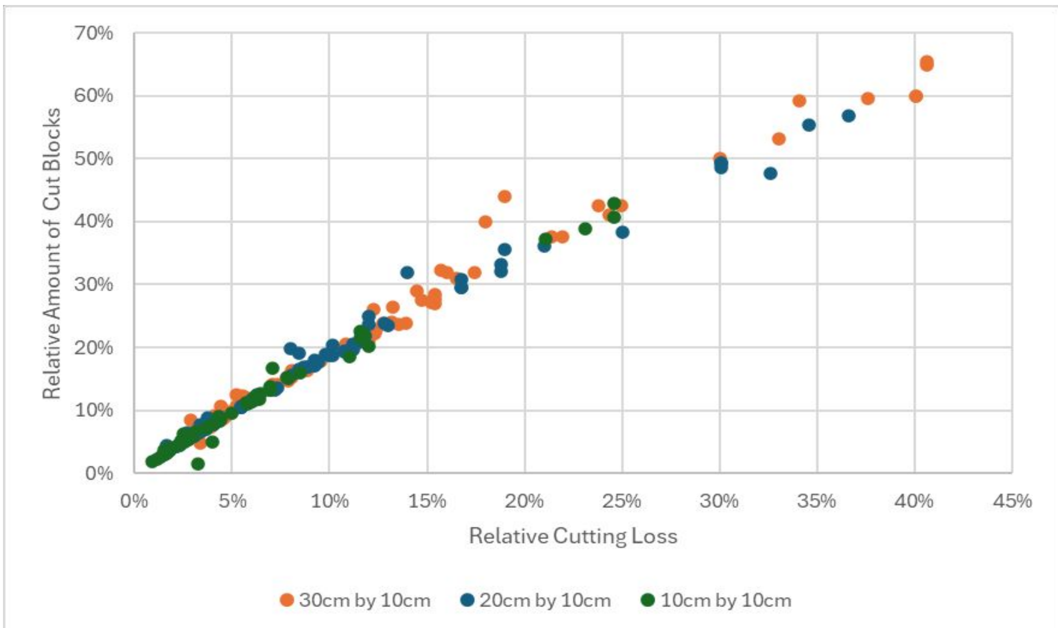


Figure 21. Relationship of cutting loss to cut blocks

Other design properties that are related to block placement is the amount of small cut blocks. A small cut block is defined as a cut block with less than 25% of the block area. It is recommended that small cut blocks should not be placed in the pavement since these pieces can be dislodged easily under traffic (Concrete Block Paving: Book 3–Specification and Installation 2008). Figure 22 shows that cutting loss and the amount of small cut blocks have a direct relationship as well, meaning that as long as the cutting loss is minimized, the amount of small cut blocks will also be minimized. Similar to the previous figure, smaller cut blocks would have a fewer amount of small cut blocks.

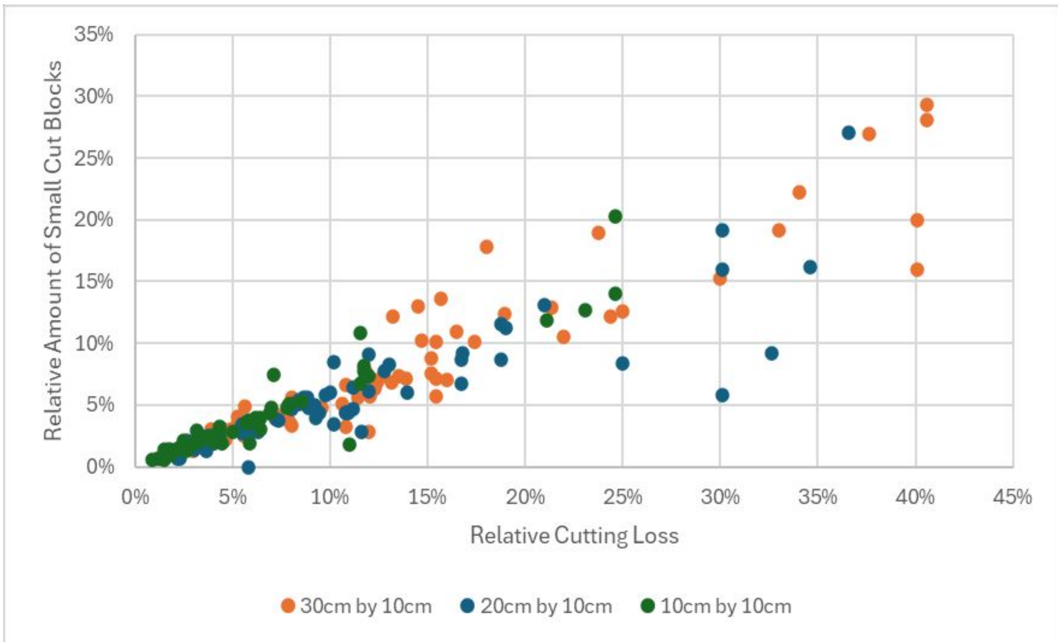
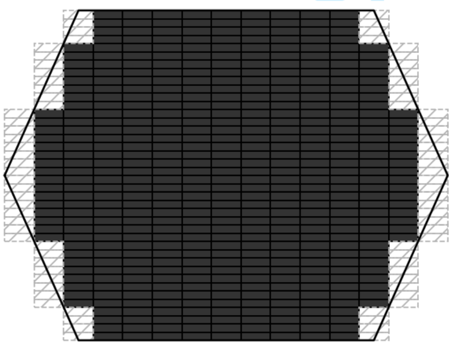
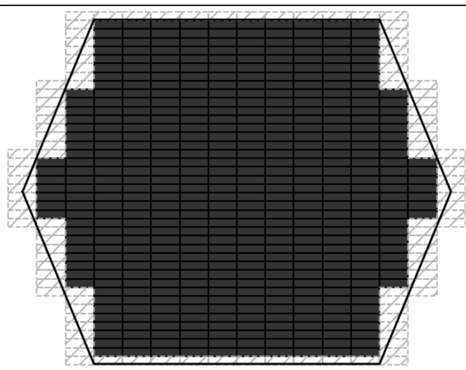
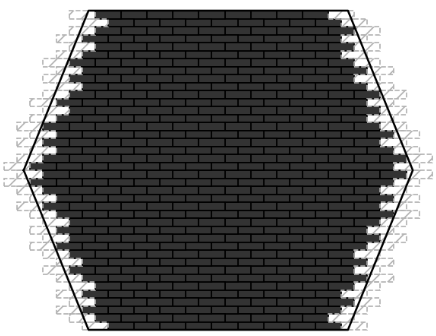
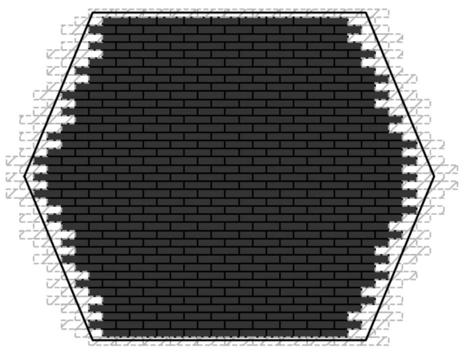


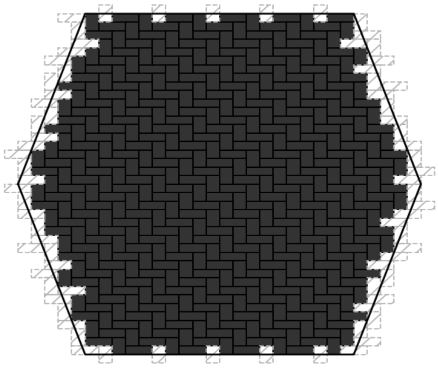
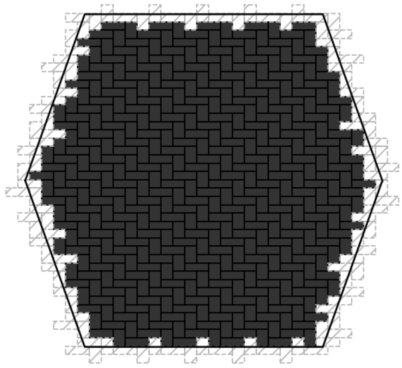
Figure 22. Relationship of cutting loss to small cut blocks

4.2 Adherence to Construction Guidelines

As seen in the sensitivity analysis is that the relative cutting loss could be as much as 40% of the pavement area, even if the construction guidelines in Seoul only allows an additional 8% of the estimated quantity to account for breaking and cutting blocks. To show the practical application of the model, the optimized layout was determined on a $10m^2$ pavement, using 20cm by 10cm blocks in different patterns. According to the guidelines, the estimated quantity of blocks with the additional 8% allowance is 540 blocks. Both the model results and inefficient layout are shown in Table 4, with the inefficient layout defined as those not adherent to the allowable construction guidelines in Seoul. The percentage included inside the parenthesis is the additional allowance permitted by the guidelines. The model was able successfully provide a layout that would follow the allocated blocks. It is also possible that without optimizing the layout would result in more blocks being needed. This highlights the importance of optimizing the block layout because by applying the model, it was able to determine a result that would be able to follow the guidelines, and it can also be used as a theoretical basis to estimate the number of blocks needed to cover a particular area.

Table 4. Comparison of the Empirical and Theoretical Block Pavement Layout

Construction Guideline (Empirical)	**Model Results (Theoretical)**	**Inefficient Layout**
540 blocks (8%)	**Stack Bond Pattern**	
	 0.88m^2 cutting loss with 536 total blocks placed (7.2%)	 1.00m^2 cutting loss with 560 total blocks placed (12.0%)
	Stretcher Pattern	
	 0.80m^2 cutting loss with 540 total blocks placed (8.0%)	 1.80m^2 cutting loss with 556 total blocks placed (11.2%)
	Herringbone Pattern	

	 0.74m² cutting loss with 537 total blocks placed. (7.4%)	 1.40m² cutting loss with 554 total blocks placed (10.8%)

5. Conclusion

In this article, a model was presented to optimize the minimum cutting loss of block pavement of any shape. Most studies are limited to having only rectangle bounding areas or not having any patterns regarding the placement of the blocks. The model is subjected to several constraints such as amount, geometric boundary, overlap, and pattern constraint. The non-linear model was solved using harmony search algorithm which was able to reasonably solve and converge the cutting loss to a minimum value. By minimizing the cutting loss, the amount of cut blocks can also be minimized which can save on costs and placement time. The sensitivity analysis showed using a one-way ANOVA with a significance value of 0.05 showed that block size, pavement area and convexity would have significant effect on the cutting loss. Using smaller block sizes and a lower convexity pavement area would decrease cutting loss and that with a pavement area greater than $15m^2$,the relative cutting loss tends to converge to less than 5% of the pavement area regardless of the convexity of the bounding area. It also showed utilizing the model can serve as a theoretical basis of estimating the total required number of blocks necessary for a certain pavement and furthermore can comply to the required number of blocks set in construction guidelines. Future studies can improve upon this study by using other methods such as machine learning to further improve the results or by using non-rectangular blocks and other patterns. The framework of this model relies on the fact that this is currently a non-linear optimization based on the constraint equations that were developed and that HSA is a meta-heuristic algorithm used to solve the problem. Additional improvement for future research is to simplify the constraints or employ machine learning methods to determine the global optimum solution.

Acknowledgements

This research was supported by the Chung-Ang University Young Scientist Scholarship in 2021 and the BK21 Graduate Student Scholarship. The authors gratefully acknowledge their financial support.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CrediT authorship contribution statement

Ivan Jan Urbino: Methodology, Software, Formal Analysis, Investigation, Data Curation, Writing-Original **Hyunsik Hwang:** Validation**,** Formal Analysis, Writing-Review & Editing **YoonHo Cho:** Conceptualization, Validation, Writing-Review & Editing, Visualization, Supervision, Project administration

Data availability

Data generated or analyzed during this study are available from the corresponding author upon reasonable request.

References

Abate, M. K. (1993). Concrete paving blocks: an overview.

Arenales, M., & Morabito, R. (1995). An and/or-graph approach to the solution of two-dimensional non-guillotine cutting problems. *European Journal of Operational Research, 84*(3), 599-617. doi:https://doi.org/10.1016/0377-2217(95)00026-m

Askarzadeh, A., & Rashedi, E. (2017). *Harmony Search Algorithm. Recent Developments in Intelligent Nature-Inspired Computing.*

Beasley, J. E. (2004). A population heuristic for constrained two-dimensional non-guillotine cutting. *European Journal of Operational Research*, 601-627.

Bourke, P. (1997). Polygons and meshes.

Brinkhoff, T., Kriegel, H. P., Schneider, R., & Braun, A. (1995). Measuring the Complexity of Polygonal Objects. *ACM-GIS, 109*.

Chan, H., & Markov, I. L. (2004). Practical slicing and non-slicing block-packing without simulated annealing. *14th ACM Great Lakes symposium on VLSI* (pp. 282-287). New York: ACM.

Concrete Block Paving: Book 3–Specification and Installation (5th ed.). (2008). Midrand: Concrete Manufacturers Association.

Dowsland, K. A. (1987). An exact algorithm for the pallet loading problem. *European Journal of Operational Research, 31*(1), 78-84.

Fang, J., Rao, Y., & Shi, M. (2023). A deep reinforcement learning algorithm for the rectangular strip packing problem. *Plos one, 18*(3). doi:e0282598

Gavrilova, M., & Rokne, J. G. (2000). Reliable line segment intersection testing. *Computer-Aided Design, 32*(12), 737-745.

Haessler, R. W., & Sweeney, P. E. (1991). Cutting stock problems and solution procedures. *European Journal of Operational Research, 54*(2), 141-150.

He, T., Wang, H., & Yoon, S. W. (2018). Comparison of four population-based meta-heuristic algorithms on pick-and-place optimization. *Procedia Manufacturing, 17*, 944-951.

Huang, E., & Korf, R. E. (2013). Optimal rectangle packing: An absolute placement approach. *Journal of Artificial Intelligence Research, 46*, 47-87.

Innocente, M., & Sienz, J. (2021). Particle swarm optimization: Development of a general-purpose optimizer. From arXiv preprint arXiv:2101.09835

Jamshidi, A., Kurumisawa, K., White, G., Nishizawa, T., Igarashi, T., Nawa, T., & Mao, J. (2019). State-of-the-art of interlocking concrete block pavement technology in Japan as a post-modern pavement. *Construction and Building Materials, 200*, 713-755.

Leung, J. Y., Tam, T. W., Wong, C. S., Young, G. H., & Chin, F. Y. (1990). Packing squares into a square. *Journal of Parallel and Distributed Computing, 10*(3), 271-275.

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation, 188*(2), 1567-1579.

Mampearachchi, W. K., & Senadeera, A. (2014). Determination of the most effective cement concrete block laying pattern and shape for road pavement based on field performance. *Journal of materials in civil engineering, 26*(2), 226-232.

Mikusova, M., Abdunazarov, J., Zukowska, J., & Jagelcak, J. (2020). Designing of parking spaces on parking taking into account the parameters of design vehicles. *Computation, 8*(3), 71.

Mudiyono, R., Nor, H. M., Hainin, M. R., & Ling, T. C. (2006). *Performance of concrete paving blockspavement on sloped road section.* Universiti Teknologi Malaysia.
Pitombeira-Neto, A. R., & Murta, A. H. (2022). A reinforcement learning approach to the stochastic cutting stock problem. *EURO Journal on Computational Optimization, 10*. doi:100027
Seoul Metropolitan Government and Seoul Facilities. (2020). Seoul sidewalk construction design and construction manual.
Soutsos, M. N., Tang, K., Khalid, H. A., & Millard, S. G. (2011). The effect of construction pattern and unit interlock on the structural behaviour of block pavements. *Construction and Building Materials, 25*(10), 3832-3840.
To, P. A. (2006). Tetris-packing problem with maximizing filled grid squares. From http://hdl.handle.net/10211.3/116668
Urbino, I. J., Lin, W. G., Seunghyun, R., & Ho, C. Y. (2023). Block Layout Optimization Using the Harmony Search and Particle Swarm Algorithms. *International journal of highway engineering, 25*(5), 69-77.
Yuan, Z., Song, X., & An, D. (2022). Heuristic Approach for Packing Identical Rectangular Tiles in an Irregular Marble Slab. *Mathematical Problems in Engineering, 2022*(1). doi:9038402