# Rectangle tilings, connectivity and associated covariants

Shekhawat, Krishnendra

UNIVERSITÉ DE GENÈVE — FACULTÉ DES SCIENCES
Section de mathématiques — Prof. Daniel Coray

# Rectangle tilings, connectivity and associated covariants

THÈSE

présentée à la Faculté des Sciences de l'Université de Genève

pour obtenir le grade de

*Docteur ès Sciences de l'Université de Genève, mention Mathématiques*

par

Krishnendra SHEKHAWAT

(Inde)

Thèse Nº 4570

GENÈVE

Atelier d'impression ReproMail de l'Université de Genève

2013

# MOTIVATION

The present work discusses a problem in constructive geometry which was initially presented by architects. This problem, which concerns the allocation of living spaces in a building, was the theme of a research project *'Formalisation et sens du projet architectural'* [1]). The problem (in one of its forms) can be stated as follows:

Given a list of rooms and an allocation matrix (which defines the relative weights of the desired proximities among the rooms), all spaces must be enclosed within the shape of a cross made up of five rectangles. The constraints for allocation are that each arm of the cross must have at least one terrace and each room should be in contact with the outside world directly or through a terrace.

This problem has rarely been attacked by mathematical methods but we found it very interesting, challenging and deep because initially it was not at all clear:

1. how to define it mathematically (for example, first we tried to define it using linear programming methods),

2. which branch of mathematics should be studied and applied to solve the problem (at the beginning we tried to solve it by using some concepts given by Stiny [38], by applying the additive approach proposed by Tabor [27] Chapter 13 (see also Section 7.1), or by mimicking some calculations involved in the functioning of TEX, etc.,

3. in what form the solutions are required (geometrical or numerical).

In the end we defined it as a problem of geometry and optimization which we have solved by using some concepts of graph theory and by introducing computing algorithms and some optimization techniques.

---

[1]) *Formalization and meaning of the architectural project*. This project, supervised by Professors P. Pellegrino, D. Coray and G. Falquet, was funded by the Swiss National Science Foundation (subsidy no. K-12K1-120593). The objective was to develop a formalization of the architectural conception and to offer a computer-aided system of inference appropriate to the architectural process of composition. This research relates to the articulation between the sciences of architecture, semiotics of space, mathematics, and computing.

# ABSTRACT

In this work, we focus on *Tiling by rectangles*, *Connectivity of the corresponding tilings* and *Covariants associated with the corresponding tilings*.

Tiling by rectangles is the geometric problem of arranging a given family of rectangles in a larger specific frame. A primitive version is *rectangle tiling*, i.e., tiling rectangles inside a rectangular frame. Then the notion of rectangle tiling is extended to tiling rectangles inside some other shapes. This generalization is achieved through another object of a mathematical nature, namely the *allocation matrix*. This problem also involves the study of another mathematical concept the *extra spaces* which are introduced in the geometric distribution process. For this we shall use different algebraic and topological covariants.

When tiling by rectangles, we measure the degree of connectivity of the final tiling; an effort has been made to obtain a tiling which is best from the point of view of connectivity. Connectivity is defined in terms of adjacency for both the given rectangles and the extra spaces.

In addition, some methods have been defined to reduce the area of the obtained tilings in such a way that connectivity remains preserved. We have studied and developed several covariants related to tilings and their graph which are used to refine the number of solutions originating from the algorithm.

As far as the applications are concerned, the concept of tiling by rectangles has been applied to different fields, with special emphasis on architectural designing, where the aim is to obtain floor plans of some assigned shapes. Our mathematical theory leads to some interesting solutions, which have been tested in the framework of an interdisciplinary research project supported by the Swiss National Science Foundation (see motivation and related footnote for details about the project). The software *TOR* (tiling of rectangles), which we have produced on this occasion, generates the tiling by rectangles for various shapes, and computes the corresponding graph and a number of associated covariants, for a given set of data.

The presented research work opens a new field for applied mathematicians, in that it combines various aspects of geometry, topology and optimization

theory, towards the solution of a problem which is well known to architects, but which has rarely been attacked by mathematical methods. The covariants associated with graphs provide an innovative approach to understand and analyse the problem.

# RÉSUMÉ

Dans ce travail, nous nous intéressons aux *Pavages par des Rectangles*, en nous concentrant sur la *Connectivité* et *Covariants Associés*.

Le pavage par des rectangles est le problème géométrique qui consiste à organiser une famille de rectangles dans un cadre spécifié. Une version primitive est le pavage d'un rectangle, qui revient à disposer des rectangles à l'intérieur d'un cadre rectangulaire. Ensuite cette notion est étendue au pavage par des rectangles de certaines autres formes géométriques. Cette généralisation est obtenue par un autre objet de nature mathématique, à savoir la *matrice d'allocation*. Ce problème implique aussi l'étude d'un autre concept mathématique : les *espaces supplémentaires* qui sont introduits dans le processus de distribution géométrique. Pour cela nous utiliserons différents covariants algébriques et topologiques.

Lors d'un pavage par des rectangles, on mesure le degré de connectivité du pavage obtenu à la fin ; un effort a été fait pour obtenir un pavage qui soit optimal du point de vue de la connectivité. La connectivité est définie en termes de proximité à la fois pour les rectangles et pour les espaces supplémentaires. En outre, certaines méthodes ont été définies pour réduire l'aire des pavages de façon que la connectivité reste préservée.

Nous avons étudié et développé plusieurs covariants qui peuvent être associés aux pavages et à leur graphe et qui sont utilisés pour préciser le nombre de solutions obtenues à partir de l'algorithme.

En ce qui concerne les applications, le concept de pavage par des rectangles a été appliqué à différents domaines, et tout particulièrement à la conception architecturale, où le but est d'obtenir les plans d'étage de formes géométriques imposées. Notre théorie mathématique conduit à des solutions intéressantes, qui ont été testées dans le cadre d'un projet de recherche interdisciplinaire soutenu par le Fonds national suisse de la recherche (voir la motivation où la note en bas de page fournit les détails sur le projet). Le logiciel *TOR*, que nous avons composé à cette occasion, génère le pavage par des rectangles pour différentes formes géométriques, et calcule le graphe correspondant et un certain nombre de covariants associés, pour un ensemble de données fournies.

Ce travail de recherche ouvre un nouveau domaine pour les mathématiciens appliqués, en ce qu'il combine divers aspects de géométrie, de topologie et de théorie de l'optimisation, pour trouver la solution d'un problème qui est bien connu des architectes, mais qui a rarement été attaqué par des méthodes mathématiques. Les covariants associés aux graphes fournissent une approche innovante pour comprendre et analyser ce problème.

# ACKNOWLEDGEMENT

This work was conducted under the guidance of Professor Daniel Coray, University of Geneva, Switzerland. The research work was funded by the Faculty of Science of the University of Geneva and by the SNF (Swiss National Foundation) Project *formalization et sens du projet architectural.*

Professor Coray as a mentor offered his immense knowledge at every stage of the research. In addition, he always gave moral support and showed trust in my work and me. I am sure that without his assistance, it would have been impossible to even dream of this Dissertation.

Professor Pierre Pellegrino as a project leader helped a lot in arranging the funds for the research, which was very vital. His regular monitoring (in the form of scheduling, discussing, and giving short term goals) at every stage helped me to achieve the results in time.

I would also like to thank Professor Gilles Falquet (co-leader of the project), Dr. Mathieu Vonlanthen (project-colleague) who helped in the development of the software and in the formulation of the research tasks and Dr. Emmanuelle P. Jeanneret (project-colleague) who helped a lot in understanding the underlying architectural concepts and in making the research work fully interdisciplinary.

I would like to show my sincere gratitude toward Madam Lorenza Coray for her kind moral support and to my parents Mr. O. P. Shekhawat and Mrs. Sharda Shekhawat for generously providing everything for my education and for their dedicated support even from far away. I also appreciate my wife Komal Rathore, for reading the Thesis again and again and for providing encouragement at every small stage. At the end, I would like to thank my friends, especially in Geneva, for their support.

# INTRODUCTION

*"No clever arrangement of bad eggs ever made a good omelet"* said C. S. Lewis, a British scholar and novelist. Agreed, a clever arrangement can never make taste good what is bad, but surely it can make the omelet look pleasant.

Arrangements of objects of a set into patterns is a very routine and useful activity but surprisingly it remains unnoticed. Knowingly or unknowingly there are always some rules which need to be specified for arranging different objects. For example, a child arranges his colours, or a woman organizes the utensils in a kitchen, Fibonacci series is an arrangement of numbers in a sequence, etc. From a wider perspective, the problem of arranging rooms in a floor plan requires special skills in two-dimensional geometry.

An arrangement of objects which can be viewed as a branch of *constructive combinatorics*[2]) is fueled by problems and applications from many fields of study and that's why the problem of arrangement needs to be handled and solved properly. We deal with some of the problems that are faced in the arrangement of different items or objects.

An arrangement has two basic elements; first, its *geometry (position)*; the position of the objects is always kept in mind while organizing them. Examples of geometric arrangements are furniture organized in the most functional way, the famous Rubik Cube, or the arrangement of squares in the golden rectangle. The second essential element is *topology (similarity)* of the objects. Elements having similar shape, reaction, function, orientation or direction are often grouped together. For example, in graph theory, graphs are grouped together when they have some mathematical properties in common (planar graphs, bipartite graphs etc.); in the periodic table noble gases are arranged together on the basis of the chemical properties they share; in a house, the dining room and kitchen are placed together by architects.

---

[2]) *Combinatorics* is the study of arrangements: pairings and groupings, rankings and orderings, selections and allocations. *Constructive combinatorics* is the design and study of algorithms for creating arrangements with special properties [19].

In this work, we develop a mathematical model to solve the problem of arranging given rectangles (or rectangular tiles) having different sizes within a given shape. To solve this problem, we first group together the given tiles on the basis of some similarity (e.g., their functions or sizes) and then we define some methods to position them inside the required shape.

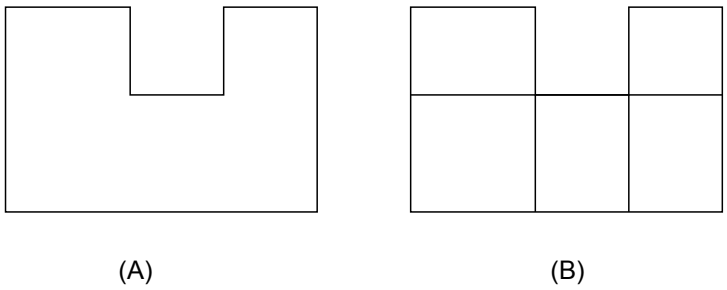


FIGURE 0
Given shape and its division into rectangles

To place $n$ rectangular tiles inside a given shape (like the one in Figure 0(A)), we consider the following procedure:

1. First we perform a geometric subdivision of the given shape into, say, $k$ rectangles. For example, the shape in Figure 0(A) is subdivided into $k = 5$ rectangles as shown in Figure 0(B).

2. We distribute the $n$ given tiles into $k$ logical groups according to some predefined rules (architectural or otherwise). The tiles which belong to a particular group are called *components* (or *members*) of that group.

3. Then we construct a rectangle tiling for each group by arranging its components in a way which will be discussed later, for instance along a spiral.

4. At the end, we arrange the $k$ rectangle tilings thus obtained in such a way that the complete tiling has some good connectivity properties and is mathematically satisfactory.

We refer the reader to Figure 4.9, which illustrates such a computer generated U-shape tiling.

This grouping of tiles is done on the basis of the relative importance which is attached to proximity among each pair of given tiles. This in turn may depend on the functionality of these tiles. For example in architecture, if each

tile represents a room then the relative importance of proximity of kitchen and dining room would be greater than that of kitchen and bedroom.

For the positioning of the given tiles into a rectangular composition, we consider mainly two criteria, namely *area* and *connectivity*, where connectivity is defined in terms of adjacency among the given tiles. Arrangements that are constructed in the present work are among the best arrangements for the given tiles in a rectangular frame from the point of view of connectivity.

A tile is certainly a geometrical shape but it can also represent a function of real numbers (for example time, cost etc.) to solve different problems using the concept of arrangement of tiles. Suppose there are different departments in an industry and each department has a certain number of employees. If the position of a tile represents the position of an employee where the corresponding employee works and the area of each tile represents the time for an employee to finish a certain task, we can say that two tiles share an edge if the corresponding employees need to finish some work collectively. Sometimes there is also a need to provide buffer time to some employees to complete the task. This buffer time gives an idea of those employees who should work together and for how long. This results in augmentation of efficiency while keeping an eye over cost. With the knowledge of an entire framework, the problem is to reduce cost and to augment efficiency. In terms of arrangement of tiles, the problem is to arrange tiles in such a way that the connectivity should be maximized while the area should be minimized.

In the above paragraph, the buffer time is mentioned, similarly in an arrangement we generate some extra spaces which can be referred as buffers; for example, in a house there are some extra spaces which are used as corridors, balconies, staircases or store rooms. Even in mathematics, when one integrates a function, one obtains a primitive of the function plus a constant. Here the problem is about the integration of different tiles where the extra space is generated like an constant. Therefore, in the arrangement of tiles the concept of extra spaces and effective methods to reduce the size of extra spaces are introduced.

To compare two arrangements or to characterize them, some numbers are needed. These numbers are said to be *covariants*. In this work, we develop and study various covariants associated with the obtained arrangements and their graph. These covariants have the following functions:

1. They help to study and understand the problem.

2. They characterize and measure the difference between different solutions.

3. They are used to obtain the best solution among numerous solutions and help in reducing their number.

In architectural terms, the concept of arrangement of tiles occurs in the form of *space allocation* or *floor plan generation.* As an application, we define some methods for obtaining floor plans. Also, it is exciting to visualize what has been done and to play around with the results by changing inputs. Therefore, we have developed a piece of software to understand, feel and envision the work done.

## SUMMARY

Chapter 1 is a general introduction to tiling by rectangles and to the connectivity of tilings. Also, we discuss the Fibonacci tiling and the arrangement of squares in a Fibonacci rectangle.

In Chapter 2, we define and obtain rectangle tiling for a given family of tiles and develop some associated invariants and covariants. In addition, we study the extra spaces and derive some techniques to reduce the size of the obtained rectangular composition in such a way that the connectivity remains preserved.

In Chapter 3, we define adjacency among rectangles of tilings and obtain the graph of a tiling. We also prove that the proposed algorithm gives one of the best rectangle tilings from the point of view of connectivity.

Chapter 4 is about generalizing rectangle tiling using an allocation matrix to obtain some other shape tilings. It also elaborates on the possibilities for different numbers of solutions for a given shape.

In Chapter 5 we define adjacency among rectangles in different shape tilings and we obtain the graph of the corresponding tilings. Afterwards, we derive various results and covariants related to the degree of connectivity of graphs.

In Chapter 6 we elaborate on graph invariants and covariants and obtain best solutions among many solutions on the basis of some of these covariants. At the end, we provide some methods to reduce the number of solutions.

Chapter 7 is concerned with the applications of rectangle tiling to different fields, particularly to architecture.

Chapter 8 discusses the construction of the software which is developed to visualize the tiling by rectangles.

We conclude with a discussion about future developments.

## NOTATIONS AND TERMINOLOGY

Notations and terms frequently used in the text are given as follows:

$a \asymp c$ : overlapping of walls $a$ and $c$ (see page 25)
$A_{initial}$ and $A_{final}$ : see page 52
$A_O$ and $A_I$ : see page 51
$A_T$ : allocation matrix (see page 71)
$E_S^2$ : the number of edges among two different groups (see page 90)
$E_S^P(n)$ : the number of edges in $G_S^P(n)$ (see page 91)
$E^R(n)$ and $E_S^R(n)$ : the number of edges in $G^R(n)$ and $G_S^R(n)$ respectively (see page 60)
$E^R(n) + S^R(n) = 3n + 1$ : Theorem 3.18 (see page 62)
$E_S^R(n) = 3n - 7$ : Theorem 3.13 (see page 60)
$G^R$ and $G_S^R$ : adjacency graph of $T^R$ and $T_S^R$ respectively (see page 59)
$\{k_1, k_2, k_3, k_4\}$ : adjacent numbers (see page 62)
$(L^1, H^1), (L^2, H^2), (L^3, H^3), (L^4, H^4), (L^5, H^5)$ : width and height of central, left, upper, right and lower $T_S^R$ respectively (see page 76)
$L_i$ and $H_i$ : width and height of a $T_S^R$ after drawing $i^{th}$ tile (see page 36)
$l_i$ and $h_i$ : width and height of $i^{th}$ tile (see page 36)
$R_A$ and $R_B$ : see page 78
$R_i$ : tile of width $l_i$ and height $h_i$ (see page 38)
$S^R$ : adjacent sum (see page 62)
$T^P(n)$ : plus-shape tiling of order $n$ (see page 72)
$T_S^P$ : spiral-based $T^P$ (see page 76)
$T^R$ : rectangle tiling (see page 35)
$T^R(n)$ : $T^R$ of order $n$ (see page 35)
$T_S^R$ : spiral-based $T^R$ (see page 37)

*adjacency graph* and *degree of connectivity of a shape tiling*: see page 28
*adjacency matrix*: see page 95
*adjacent side of a* $T^R$ : see page 61
*ascending order of allocation*: see page 46
*central, left, upper, right, lower* $T^R$ : see page 72
*circulation* and *terrace*: see page 112
*direct adjacency among rectangles*: see page 25
*direct adjacency* and *adjacency via virtual parts*: see page 56
*extra space*: see page 24

# TABLE OF CONTENTS

# CHAPTER 1

# TILING BY RECTANGLES

This chapter begins with the general definition of a tiling and related literature. Then we define rectangular tilings for given shapes and introduce the corresponding terminology. At the end, the arrangement of squares in a Fibonacci tiling is discussed. This notion will play a major role in most of this work.

## 1.1 INTRODUCTION AND RELATED WORK

In ancient human culture and in the natural world, traces of tesselations have been found. As a mathematical and scientific study, the history of tesselations is not very long. But since the beginning of the twentieth century, mathematicians have been interested in the concept of tiling plane figures. Some examples of tilings include Penrose tilings and real-life tilings such as ceilings and floors (see Papenfus [33]). The tilings by squares, triangles and hexagons are the simplest and are frequently seen in everyday life, for example, in chessboards and honeycombs.

DEFINITION 1.1. A *plane tiling* $T$ is a countable family of closed sets $T = \{T_1, T_2, \dots\}$ which covers the plane without any gaps or overlappings. Explicitly, the union of the sets $T_1, T_2, \dots$ (which are called the *tiles* of $T$) is to be the whole plane, and the interiors of the sets $T_i$ are to be pairwise disjoint.

DEFINITION 1.2. A *monohedral tiling* is a tiling in which every tile is congruent (directly or reflectively) to one fixed set. In simple terms, all the tiles have the same size and shape.

Some remarkable monohedral tilings are those which are of spiral form. Figure 1.1 describes one such tiling (see Grünbaum and Shephard [17], Chapter one).

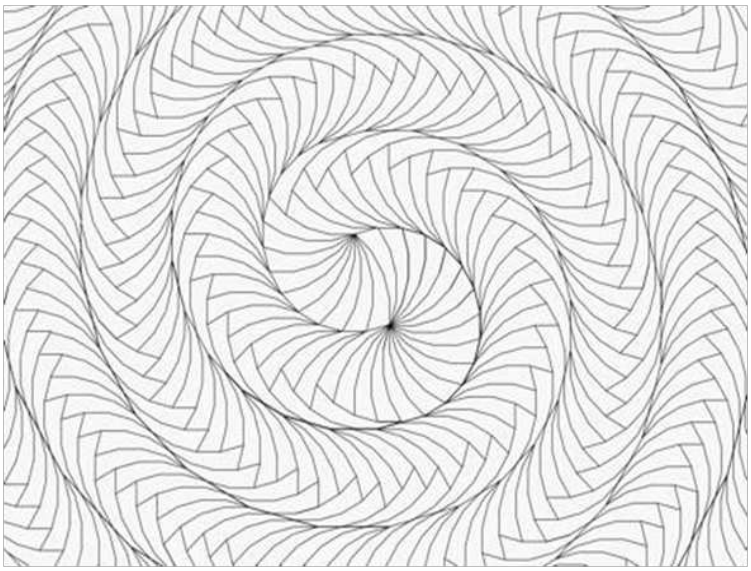

FIGURE 1.1

A monohedral tiling in spiral form (Grünbaum and Shephard).
Here all the tiles are identical in shape and are polygons with 9 sides.

DEFINITION 1.3. A rectangle which has been tiled into squares all of which have different sizes (areas) is called *perfect*.

DEFINITION 1.4. The number of squares or rectangles in the tiling of a rectangle is the *order of the tiling.*

The problem of tiling rectangles originated long ago. According to Singmaster, the history of tiling rectangles with squares began in 1902 with a puzzle titled *Lady Isabel's Casket*, written by Henry Ernest Dudeney (see Papenfus [33]). The problem is described as follows.

LADY ISABEL'S CASKET PUZZLE (see [11], page 67). *Find the dimensions of the top of a box on which there is a rectangular strip of gold, 10 inches by 1/4 inch; the rest of the surface is exactly inlaid with pieces of wood, each piece being a perfect square, and no two pieces of the same size.*

The solution of this puzzle was published by Dudeney ([11], page 191), and is shown in Figure 1.2. The rectangular strip can be regarded as an *extra space*, an important notion we shall elaborate on in Definition 1.7.

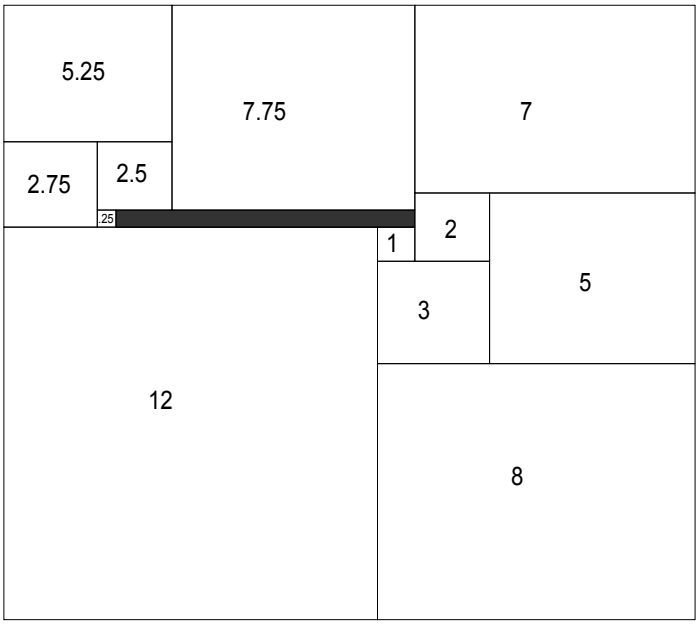


FIGURE 1.2
Solution for Lady Isabel's casket puzzle

Let $R(x, y)$ denotes a rectangle with width $x$ and height $y$. In 1903, Dehn [9] proved his famous result that $R(x, y)$ *can be tiled by (finitely many) squares if and only if* $y/x$ *is a rational number.* Hence it is noteworthy that Dehn's result applies only to finite tilings, e.g. the golden rectangle (Definition 1.16) is a union of countably many squares.

In 1925, Z. Moroń [30] gave the first example of the dissection of a $32 \times 33$ rectangle into 9 unequal squares, as shown in Figure 1.3.

Then in 1940, Brooks et al. [5] obtained a perfect squared rectangle till order 13 by associating with the tiling an electrical network consisting of currents, voltages, and resistance and using the well-known properties of such networks.

In a electrical circuit, two laws describe how the current must behave if the circuit is to be complete (*Kirchhoff's laws*).[3])

[3]) Kirchhoff's first law – The algebraic sum of the currents flowing to any of the terminals must be zero.

Kirchhoff's second law – The algebraic sum of the currents for the entire circuit must be zero.

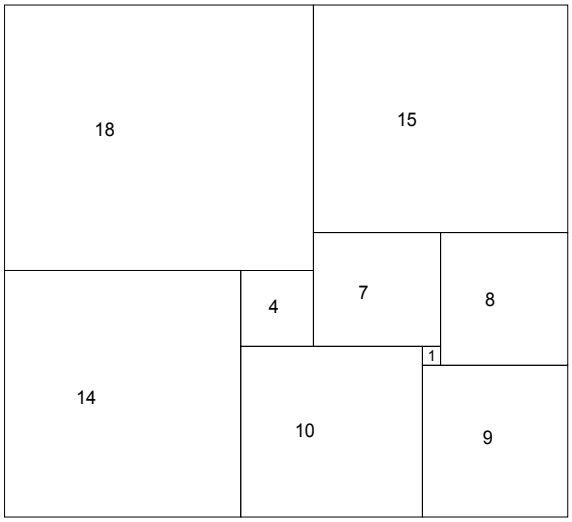


FIGURE 1.3
Dissection of a $32 \times 33$ rectangle into 9 unequal squares

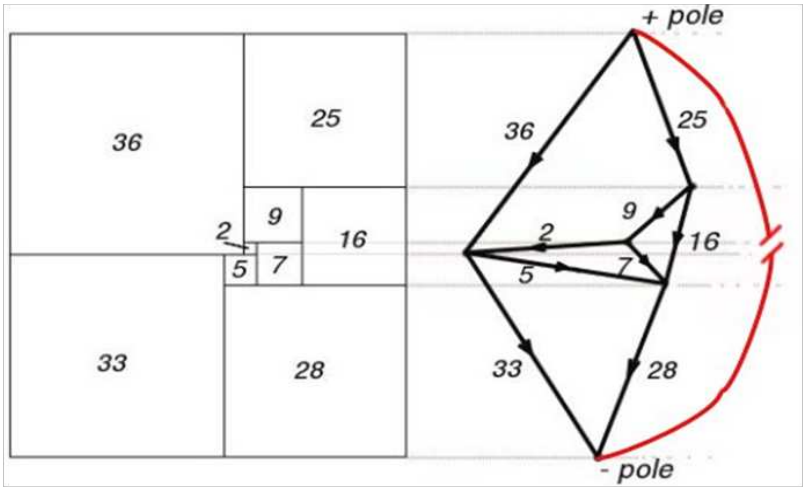


FIGURE 1.4
Electric circuit corresponding to a perfect squared rectangle

As an example, refer to Figure 1.4. In this figure each of the horizontal line segments of the squared rectangles is represented by a vertex (dot). Each vertex represents a terminal in the electrical circuit. An edge between two vertices represents the square that has two corresponding horizontal lines as boundaries.

It was C.J. Bouwkamp who first became interested in the problem of perfect squared rectangles during the early 1940's. He used computers and developed a code for finding all the possible solutions to a perfect squared rectangle of given dimensions.


FIGURE 1.5
A perfect squared square of order 21

Besides Bouwkamp, Duijvestijn [4] also discovered the very first simple perfect squared square making use of computers in March 1978. It has the lowest possible order, $n = 21$, and it is the only one of that order (Figure 1.5).

## 1.2 TILING BY RECTANGLES AND CONNECTIVITY

Here we discuss how to describe rectangular tilings for some more general shapes than rectangles.

DEFINITION 1.5. By *shape tiling* we refer to the arrangement of a collection of rectangles in the frame of a given shape, while satisfying a certain set of topological conditions which are stated and explained in Section 1.2.1. Each of the elementary rectangles is called *a tile*. The area of each tile is called the *size of the tile*, and the total number of tiles is called the *order of the tiling*.

We do not require that all the tiles should be of the same size. Neither are they supposed to form a covering of the given shape (see the definition of extra spaces below). Furthermore, shape tiling can be restricted to the frame of a given bounded shape or it can be used for a covering of the whole plane. Before discussing the topological conditions needed for shape tiling, we introduce some definitions which will be used throughout the present study.

DEFINITION 1.6. A *wall* is one of the four sides of a rectangle.

If $W$ is a wall of side length $j$, a *sub-wall* of $W$ is a proper connected part of $W$. Hence it is a closed interval of length $k$ strictly included in $W$, i.e. $0 < k < j$.

For example in Figure 1.6(a), one of the walls of rectangle A is a sub-wall of rectangle B.

DEFINITION 1.7. In shape tiling, an *extra space* is a rectangle generated because of the difference in either the width or the height of the tiles.

The extra spaces are not among the set of tiles, but they are generated automatically, in particular by Condition 5, which has to be satisfied for tiling the given shape (Section 1.2.1).

Consider a rectangle with horizontal sides $a$, $a'$ and vertical sides $b$, $b'$, and represent it as $R_{a,b,a',b'}$.

DEFINITION 1.8. Two rectangles $R_{a,b,a',b'}$ and $R_{c,d,c',d'}$ *share a wall* if any one of the following holds:

1. $a' \cap c = a'$ or $a' \cap c = c$,

2. $a \cap c' = a$ or $a \cap c' = c'$,
3. $b' \cap d = b'$ or $b' \cap d = d$,
4. $b \cap d' = b$ or $b \cap d' = d'$.

$R_{a,b,a',b'}$ *shares a wall* with $R_{c,d,c',d'}$ if any one of the following holds: $a' \cap c = a'$; $a \cap c' = a$; $b' \cap d = b'$; $b \cap d' = b$.

Let us say that walls $a$ and $c$ *overlap*, in symbols $a \asymp c$, if $|a \cap c| > 1$, i.e., if they intersect in more than one point.

$R_{a,b,a',b'}$ and $R_{c,d,c',d'}$ *share a sub-wall* if any one of the following holds:
1. $a' \asymp c$ & $a' \neq c$,
2. $a \asymp c'$ & $a \neq c'$,
3. $b' \asymp d$ & $b' \neq d$,
4. $b \asymp d'$ & $b \neq d'$.

$R_{a,b,a',b'}$ *shares a sub-wall* with $R_{c,d,c',d'}$ if any one of the following holds:
1. $a' \asymp c$ & $a' \not\subset c$,
2. $a \asymp c'$ & $a \not\subset c'$,
3. $b' \asymp d$ & $b' \not\subset d$,
4. $b \asymp d'$ & $b \not\subset d'$.

In particular, $R_{a,b,a',b'}$ and $R_{c,d,c',d'}$ *share a full wall* if any one of the following holds:
$a' = c$; $a = c'$; $b' = d$; $b = d'$.

DEFINITION 1.9. Two rectangles are *directly adjacent* if they share a *wall* or a *sub-wall*.

Figure 1.6 illustrates the concept of direct adjacency with the two rectangles A and B.

a) A and B share a *wall*. Or A shares a *wall* with B but B shares a *sub-wall* with A.

b) A and B share a *sub-wall*.

c) A and B share a *full wall*.

d) A and B share neither a *wall* nor a *sub-wall*.

In Figure 1.6, A is directly adjacent to B in the first three cases (a, b, c) but not in case (d).

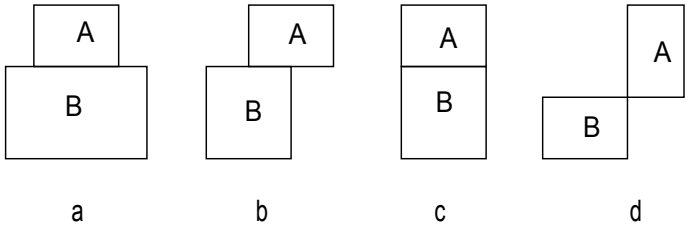


FIGURE 1.6
Four cases for defining the direct adjacency among rectangles

### 1.2.1 CONDITIONS FOR SHAPE TILING

The conditions are followed in the sequence as mentioned below:

*1. All the rectangles are normalized, i.e. they must have only horizontal and vertical sides*

For example, Figure 1.7(e) is not allowed.

*2. The intersection of any two distinct rectangles in the configuration should be an empty set i.e.* $R_{a,b,a',b'} \cap R_{c,d,c',d'} = \varnothing$

We consider rectangles as open subsets of the plane, therefore overlapping of boundaries (edges) of the rectangles is admitted but overlapping of the regions (areas) is not permitted. This means that in particular we do not consider the case of a rectangle inside another one. For example, Figures 1.7(a) and 1.7(b) are not allowed.

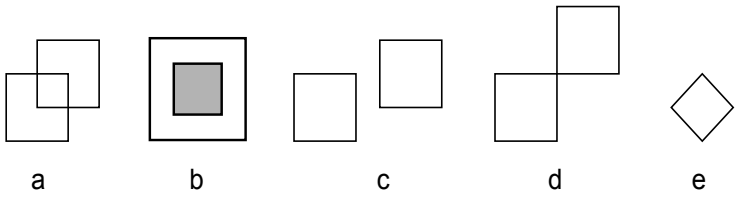


FIGURE 1.7

*3. An ordered procedure is used for arranging the given tiles*

At this stage we are not considering a specific order; the arrangement of the tiles depends on the order in which they come i.e. the tiles are arranged on *first come, first serve* basis.

*4. Each new tile must be adjacent to at least one of the previously drawn rectangles*

For example, in Figures 1.7(c) and 1.7(d), the two tiles are not adjacent therefore these figures are not allowed. But a new tile can be arranged at any position i.e. a new tile can be drawn above, below,left or right to any other tile.

*5. The composition of the tiles should always be rectangular*

To get a rectangular composition, after drawing a tile, at most two rectangular extra spaces can be drawn.

If the composition of the tiles is already rectangular then no extra space is required. For example, in Figure 1.8(a), an extra space is not required.

To draw an extra space, the width (or height) of the last drawn tile is compared with the width (or height) of the last obtained rectangular composition and thereafter at most two extra spaces are drawn to obtain a rectangular composition.

For example in Figure 1.8(b), after drawing the second tile, the composition of the two tiles is not rectangular. Therefore, two extra spaces are drawn as shown in Figure 1.8(c). In Figure 1.8(d) the heights of the two tiles are compared and two extra spaces are drawn as depicted in Figure 1.8(e).

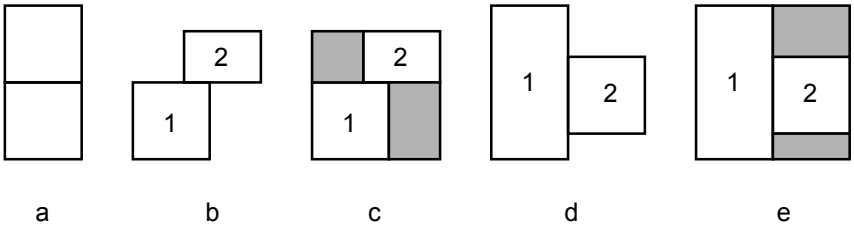


FIGURE 1.8
Drawing extra spaces

*6. An extra space can not be considered as a new tile*

An extra space can be merged into an existing tile (cf. Definition 3.1 where virtual and real parts are defined) but it can not be considered as a new tile. It means the number of tiles in a tiling always remains preserved.

### 1.2.2 CONNECTIVITY AND HOW TO MEASURE IT

A *graph G = (V, E)* is a mathematical structure consisting of two finite sets $V$ and $E$. The elements of $V$ are called *vertices* and the elements of $E$ are called *edges*. Each edge has a set of one or two vertices associated with it, which are called its *endpoints*.

A *simple graph* has neither self-loops nor multi-edges.

Two vertices $u$ and $v$ are called *adjacent* if an edge exists between them.

In an undirected graph $G$, two vertices $u$ and $v$ are *connected* if $G$ contains a path (or walk) from $u$ to $v$. A graph is *connected* if for every pair of vertices $u$ and $v$, there is a walk from $u$ to $v$.

For all these definitions, we refer to Jonathan and Yellen [16], Chapter 1.

DEFINITION 1.10. The *adjacency graph of a shape tiling* is a simple undirected graph obtained by representing each tile as a *vertex* and then drawing an *edge* between any two vertices if the corresponding tiles are *adjacent.*

The *degree of connectivity of a shape tiling* is defined by the adjacency among the different tiles and it is given in terms of the connectivity of the corresponding adjacency graph. Therefore, the comparison of the connectivity of different tilings is done by comparing the connectivity of the corresponding adjacency graphs.

Various measures are available to compare the connectivity of two adjacency graphs having the same number of vertices. For example, a comparison can be made on the basis of the number of edges, the diameter, average distance, number of cycles etc. In the present work, the *number of edges* in the adjacency graph is regarded as a measure of connectivity. For further details about these measures, see Rodrigue et al. ([36], Chapter 2, *measures and indices of graph theory*).

DEFINITION 1.11. If two connected adjacency graphs have the same number of vertices then the adjacency graph having more edges is considered to be *more connected.*

To be noted that the connectivity of adjacency graphs is independent of the order in which corresponding tiles are arranged.

## 1.3 FIBONACCI TILINGS

In this section, we describe the arrangement of squares which is associated with Fibonacci rectangles and with the golden rectangle. We also recall some well-known definitions.

### 1.3.1 THE GOLDEN RATIO

DEFINITION 1.14. The *golden number* is, in a sense, the most natural real number, since it can be written as:

$$\varphi = [\overline{1}],$$

without reference to any numbering system. This is standard abbreviation for the continued fraction expansion

$$\varphi = 1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \dots}}}.$$

From this we see that $\varphi - 1 = \frac{1}{\varphi}$, so that $\varphi$ is also equal to $\frac{1+\sqrt{5}}{2}$.

An obvious geometric interpretation of this number is that one can remove a square with side length one from a rectangle of sides $1 \times \varphi$ and obtain a new rectangle, with sides $\frac{1}{\varphi} \times 1$, which is similar to the original one. Hence, the construction can be repeated. See Walser [42], Chapter 3.

The golden number is also the limit of the sequence of *convergents*

$$\left(p_n/q_n\right)_{n=0}^{\infty} = \left(\frac{1}{1}, \frac{2}{1}, \frac{3}{2}, \frac{5}{3}, \frac{8}{5}, \frac{13}{8}, \frac{21}{13}, \dots\right),$$

which involves the successive Fibonacci numbers. This is not just one more occurrence of the Fibonacci numbers; in fact, a classical result (see Hardy & Wright [18], Chapter 10) is that the sequence of convergents is a sequence of *best approximations* to $\varphi$ by rational numbers, in a very specific sense. So, instead of saying that $\varphi = 1.61803\dots$, which is meaningful only in the decimal system, it is better to consider $\varphi$ as a sequence of convergents

$$\Phi = \left(\frac{1}{1}, \frac{2}{1}, \frac{3}{2}, \frac{5}{3}, \frac{8}{5}, \frac{13}{8}, \frac{21}{13}, \frac{34}{21}, \frac{55}{34}, \frac{89}{55}, \dots\right).$$

At any rate, this is how it appears when one attempts to interpret the work of many architects.

In fact, partly because of the necessity to work with standard units of measure and partly for conceptual reasons (the *taxis*), architects have often drawn their sketches on a regular orthogonal grid. In this way, the main building could fit in a rectangle of (say) 55 by 89 units, in which the various rooms could be represented by rectangles of sizes $8 \times 13$, $5 \times 8$, etc.

REMARK. If a positive real number $\alpha$ is of the form

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \dots}}},$$

where the $a_i$ are positive integers (the $a_i$ are called *partial quotients* of the continued fraction), one often writes

$$\alpha = [a_0, a_1, a_2, a_3, \dots].$$

This representation is periodic if and only if $\alpha$ is a quadratic irrationality. (The period is denoted by a bar above the sequence of partial quotients that are repeated.) In computations it is not possible to include infinitely many terms. Therefore, an actual calculation would end after some finite number of terms, $n$, and the continued fraction yields a rational approximation:

$$\alpha \cong [a_0, a_1, a_2, \dots, a_n].$$

See Hardy & Wright [18], Chapter 10.

Thus, the interpretation in terms of the continued fraction expansions can also be given for any ratio occurring in the literature. Such classical ratios have been used by artists and architects since the middle ages like

$$\alpha = [b_0; \overline{b_1}]$$

where $a_0 = b_0$ and each $a_i = b_1$, for $i > 0$. For example,

$$\alpha = \sqrt{2} = [1; \overline{2}],$$

which is the limit of the sequence of convergents $\left(\frac{1}{1}, \frac{3}{2}, \frac{7}{5}, \frac{17}{12}, \frac{41}{29}, \dots\right)$. Or

$$\alpha = \sqrt{5} = [2; \overline{4}],$$

which is the limit of $\left(\frac{2}{1}, \frac{9}{4}, \frac{38}{17}, \frac{161}{72}, \dots\right)$. Or, in fact, any positive real number. This removes some of the esoteric myths that have been associated with the

golden number, since architects could have used any system of proportions on a grid. They did not need to write irrational numbers like $\sqrt{5}$, because such numbers revealed themselves from the sequence of proportions used on the grid.

The ancient Greeks believed that a rectangle constructed in the proportions of the golden number was the most aesthetically pleasing of all rectangles; hence they incorporated this shape into many of their architectural designs.

However, the golden ratio has some other properties which have made it a favourite of some (though not all) architects: Brunelleschi, Palladio, Le Corbusier, Wright and others. This is where *spirals* have a role to play.

### 1.3.2 THE LOGARITHMIC SPIRAL

DEFINITION 1.15. The *logarithmic spiral* is centred at the point $O$ which is the intersection of the two diagonals $BD$ and $CE_1$ (see Figure 1.9). Its radius $r$ is reduced by a factor $\varphi$ each time the angle $\theta$ is decreased by $\pi/2$. So, its equation is of the form

$$r = r_0\, \varphi^{\frac{2}{\pi}\theta}\,,$$

where we can start with $\theta = 0$ for $r = r_0 = |AO| = \sqrt{2}\,\frac{\varphi}{\sqrt{1+\varphi^2}}$.

A basic property of the logarithmic spiral is that, at every point, its tangent vector makes a constant angle $\psi$ with the radius. This angle is specified by the formula

$$\tan\psi = \frac{r}{\left(\frac{dr}{d\theta}\right)} = \frac{\pi}{2\,\log\varphi}\,.$$

This also shows that the spiral is not tangent to the side $AD$ at the point $A$, since the angle between $OA$ and $AD$ is equal to $\frac{\pi}{4} + \arctan\frac{1}{\varphi} = 1.33897\ldots$, while $\psi = 1.2735\ldots$. The difference is of less than $4^\circ$ and we have drawn for comparison the quarter circle with centre $E_1$ inscribed in the first square.

### 1.3.3 THE ARRANGEMENT OF SQUARES IN A FIBONACCI RECTANGLE

DEFINITION 1.16. A rectangle whose ratio of width over height is equal to $\varphi$ is called a *golden rectangle*.

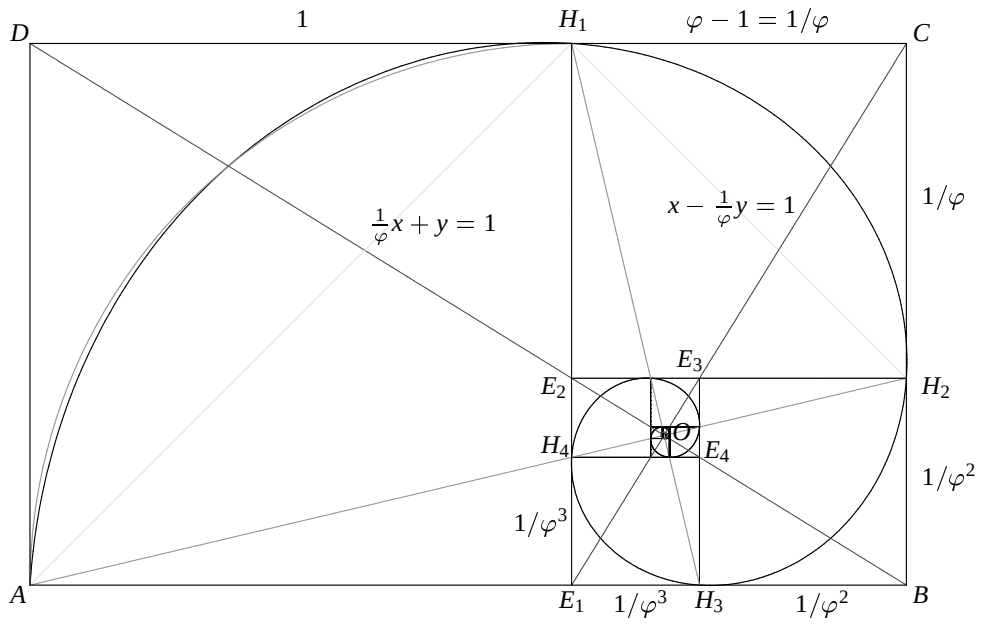


FIGURE 1.9
A golden rectangle and logarithmic spiral

DEFINITION 1.17. A *Fibonacci rectangle* is a rectangle whose side lengths are in proportion $F_{n+1}/F_n$, where the $F_n$ are the successive Fibonacci numbers: $F_0 = F_1 = 1$, $F_2 = 2$, ..., $F_{n+1} = F_n + F_{n-1}$.

The next lemma shows that a Fibonacci rectangle is naturally tiled by squares and therefore we can also speak of a *Fibonacci tiling.*

LEMMA 1.18. *The process of cutting off a square can be done by starting from any Fibonacci rectangle. The ratios of width over height for the successive residual rectangles run through all quotients of the form* $F_n/F_{n-1}$, $F_{n-1}/F_{n-2}$, *etc., until we reach the last square, which is of size* 1. *Then the residual rectangle is a square of side length* $F_1 = F_0 = 1$.

*Proof.* Removing a square of side length $F_n$ from a rectangle with sides $F_{n+1} \times F_n$ yields a rectangle with sides $F_n \times F_{n-1}$,since $F_{n+1} - F_n = F_{n-1}$. And we can repeat the process until we get a rectangle with sides $F_2 \times F_1 = 2 \times 1$. The next square to be cut off has side length $F_1 = 1$ and the residual rectangle is a square of side length 1.

For further details, refer to Figure 1.10 from (G) to (A) and see Walser [42], page 39. □

THEOREM 1.19. *One can construct a sequence of Fibonacci rectangles (and obtain a golden rectangle in the limit, as $n \to \infty$) by reversing the process described in Lemma 1.18. Starting from two unit squares one above another, one first adjoins a square of side length 2 to their right, so as to obtain a Fibonacci rectangle with sides $F_3 \times F_2$. Then one adjoins a square of side length $F_3$ below, so as to obtain a Fibonacci rectangle with sides $F_5 \times F_3$, etc. following a clockwise movement.*

*Proof.* The possibility of this construction of a sequence of Fibonacci rectangles follows from the relation $F_{n+1} = F_n + F_{n-1}$.

We know that

$$\lim_{n\to\infty} \frac{F_{n+1}}{F_n} = \varphi,$$

therefore, in the limit, as $n \to \infty$, the sequence of Fibonacci rectangles tends to a golden rectangle. For details refer to Figure 1.10 from (A) to (G), where the beginning of the sequence of Fibonacci rectangles is displayed. □

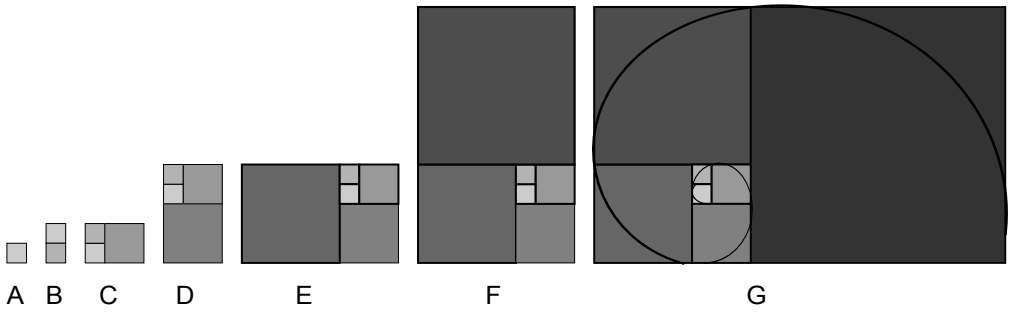


FIGURE 1.10
Sequence of Fibonacci rectangles and a Fibonacci spiral

DEFINITION 1.20. The logarithmic spiral can be replaced by its natural approximation by quarter circles in the successive squares appearing in this construction. We will call it the *Fibonacci spiral*.

EXAMPLE 1.21. The Fibonacci spiral is shown in Figure 1.10 (G).

REMARK. Theorem 1.19 provides a method for tiling squares in the frame of a rectangle, by arranging all the squares along a spiral.

As discussed earlier, many artists and architects have included in their works some rectangular components which have aspect ratios close to the golden ratio however there is rarely any evidence which would explain why the artists or architects used the golden ratio (see [26], Chapter *Proportion*).

In Chapter 3, the reason for using the golden rectangle so often is given in terms of connectivity. In the next chapter, we shall define rectangle tilings; then we shall use the method of arranging squares in a Fibonacci rectangle to obtain some particular rectangle tilings.

# CHAPTER 2

# SPIRAL-BASED RECTANGLE TILINGS

This chapter deals with shape tiling when the given shape is a rectangle. In most applications, like urbanism or architecture, disposing the tiles at random would certainly not be considered a good way. We suggest an explicit algorithm for disposing the tiles along a spiral, which often gives surprisingly good results. In addition, we study the extra spaces that this method introduces and we investigate some associated invariants and covariants.

## 2.1. AN ALGORITHM FOR RECTANGLE TILING

DEFINITION 2.1. We refer to *rectangle tiling* for shape tiling in the sense of Definition 1.5 when the given shape is a rectangle. Thus the tiles are also rectangular and the conditions prescribed in Section 1.2.1 have to be met. Such a tiling will be denoted by $T^R$.

If $T^R$ is of order $n$, i.e. if it consists of $n$ tiles, we shall also denote it by $T^R(n)$.

We aim to obtain a $T^R$ which is best from the point of view of connectivity.

### 2.1.1 SPIRAL-BASED SEQUENCE

As defined in Chapter 1, a Fibonacci rectangle follows the Fibonacci sequence for tiling a rectangle with squares. In this section we proceed in a similar way with a more general sequence. We begin with a purely number-theoretic definition:

DEFINITION 2.2. Suppose we are given two sets of positive real numbers: $\ell_i$ and $h_i$ $(i = 1, \ldots, n)$. Then we set $L_0 = 0$ and $H_0 = 0$ and define recursively two new sequences of numbers:

For $i = 1, 3, 5, \ldots$

$$L_i = L_{i-1} + \ell_i, \qquad H_i = \max(H_{i-1}, h_i)\,.$$

For $i = 2, 4, 6, \ldots$

$$L_i = \max(L_{i-1}, \ell_i), \qquad H_i = H_{i-1} + h_i\,.$$

The newly obtained sequence of pairs $(L_i, H_i)$ is called a *spiral-based sequence*.

Geometrically, we have a spiral-based sequence when we consider rectangular tiles. If $T^R$ is a rectangle tiling then

a. $\ell_i$ and $h_i$ are the width and height of the $i^{th}$ tile

b. $L_i$ and $H_i$ are the width and height of the partial tiling $T^R(i)$ which is obtained after the $i^{th}$ step.

EXAMPLE 2.3. To comprehend the concept of spiral-based sequence, consider Figure 2.1, where the shaded rectangles are extra spaces and the white rectangles are tiles.

In Figure 2.1(a), $i = 2$, therefore we compute $L_2$ and $H_2$ (the width and height of $T^R(2)$). Here $L_2 = \max(L_1, \ell_2) = \max(\ell_1, \ell_2) = \ell_2$ and $H_2 = H_1 + h_2 = h_1 + h_2$.

In Figure 2.1(b), a new tile is added to obtained $T^R(2)$, hence $i = 3$, $L_3 = L_2 + \ell_3 = \ell_2 + \ell_3$ and $H_3 = \max(H_2, h_3) = \max(h_1 + h_2, h_3) = h_1 + h_2$.

Similarly, in Figure 2.1(c), $i = 4$, $L_4 = \max(L_3, \ell_4) = \max(\ell_2 + \ell_3, \ell_4) = \ell_4$ and $H_4 = H_3 + h_4 = h_1 + h_2 + h_4$.

COROLLARY 2.4. *For $n$ even we have:*

$L_n = \max\left(\ell_n, \left(\ell_{n-1} + \max\left(\ell_{n-2}, \left(\ell_{n-3} + \cdots + \max(\ell_2, \ell_1)\right)\right)\right)\right)$

$H_n = h_n + \max\left(h_{n-1}, \left(h_{n-2} + \cdots + \max\left(h_3, h_2 + h_1\right)\right)\right)$ .

*For $n$ odd:*

$L_n = \ell_n + \max\left(\ell_{n-1}, \left(\ell_{n-2} + \cdots + \max\left(\ell_4, \ell_3 + \max(\ell_2, \ell_1)\right)\right)\right)$

$H_n = \max\left(h_n, \left(h_{n-1} + \max\left(h_{n-2}, \left(h_{n-3} + \cdots + \max\left(h_3, h_2 + h_1\right)\right)\right)\right)\right)$ .

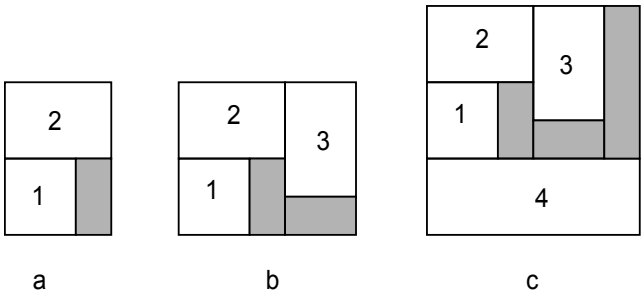


FIGURE 2.1

*Proof.* From the spiral-based sequence, for $n$ even we have $L_n = \max(L_{n-1}, \ell_n)$. Now for $L_{n-1}$, $n-1$ is odd therefore from the spiral-based sequence we have:

$$\begin{aligned} L_n &= \max\left(\ell_n, L_{n-1}\right) \\ &= \max\left(\ell_n, \ell_{n-1} + L_{n-2}\right) \\ &= \max\left(\ell_n, \ell_{n-1} + \max\left(\ell_{n-2}, L_{n-3}\right)\right) \\ &\vdots \\ &= \max\left(\ell_n, \left(\ell_{n-1} + \max\left(\ell_{n-2}, \left(\ell_{n-3} + \cdots + \max(\ell_2, L_1)\right)\right)\right)\right) \end{aligned}$$

and $L_1 = \ell_1$. The other assertions can be proved in the same fashion, by using the spiral-based sequence. □

### 2.1.2 SPIRAL-BASED ALGORITHM

In this section, we give an algorithm for the $T^R$ using a spiral-based sequence. This algorithm will be used in the upcoming chapters for developing the tilings of other given shapes because it provides one of the best $T^R$ from the point of view of connectivity (refer to Corollary 3.25).

DEFINITION 2.5. The algorithm (given below) to construct a $T^R$ will be termed *spiral-based algorithm.*

A *spiral-based rectangle tiling ($T^R_S$)* is a $T^R$ obtained using the spiral-based algorithm.

In the spiral-based algorithm there are three kinds of rectangles. Each type is given by its *position*, which is the upper left vertex, and its *size*, which

consists of the width and the height.

At each stage $i$ of the construction process there is a new *tile* $R_i$; we denote its position by $(x_i, y_i)$, and its size is the given pair $(\ell_i, h_i)$. $(x_i, y_i)$ is a point in a Cartesian frame as depicted in Figure 2.2.

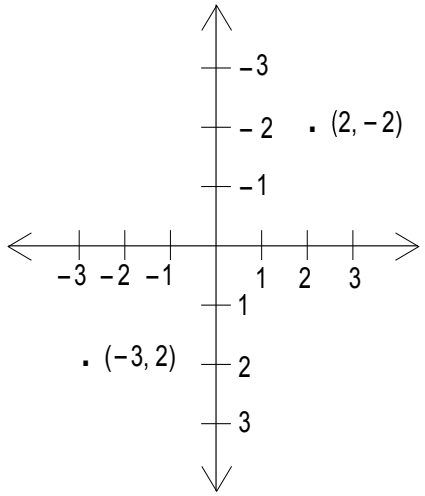


FIGURE 2.2
Cartesian plane for the position of rectangles

There is also an *extra space* $E_i$, which may be empty; we denote its position by $(t_i, u_i)$ and its size by the pair $(\lambda_i, \mu_i)$.

Finally there is a new *tiled rectangle* $T_S^R(i)$. Its position also depends on $i$; we denote it by $(X_i, Y_i)$, and the size of $T_S^R(i)$ is the pair $(L_i, H_i)$, which we already know how to compute (see Definition 2.2 and Example 2.3).

The process starts with $i = 0$, with all numbers set to $0$. Then the first tile is placed to the left of $(0, 0)$.

$i \equiv 1 \pmod 4$:
$(x_i, y_i) = (X_{i-1} - \ell_i, Y_{i-1}), \quad (X_i, Y_i) = (x_i, y_i)$

$i \equiv 2 \pmod 4$:
$(x_i, y_i) = (X_{i-1}, Y_{i-1} - h_i), \quad (X_i, Y_i) = (x_i, y_i)$

$i \equiv 3 \pmod 4$:
$(x_i, y_i) = (X_{i-1} + L_{i-1}, Y_{i-1}), \quad (X_i, Y_i) = (X_{i-1}, Y_{i-1})$

$i \equiv 0 \pmod 4$:
$(x_i, y_i) = (X_{i-1}, Y_{i-1} + H_{i-1}), \quad (X_i, Y_i) = (X_{i-1}, Y_{i-1})$

If $i$ is odd then

$(t_i, u_i) = (x_i, y_i + h_i)$ and $(\lambda_i, \mu_i) = (\ell_i, H_{i-1} - h_i)$ if $h_i < H_{i-1}$,

$(t_i, u_i) = (X_{i-1}, Y_{i-1} + H_{i-1})$ and $(\lambda_i, \mu_i) = (L_{i-1}, h_i - H_{i-1})$ if $h_i > H_{i-1}$.

If $i$ is even then

$(t_i, u_i) = (x_i + \ell_i, y_i)$ and $(\lambda_i, \mu_i) = (L_{i-1} - \ell_i, h_i)$ if $\ell_i < L_{i-1}$,

$(t_i, u_i) = (X_{i-1} + L_{i-1}, Y_{i-1})$ and $(\lambda_i, \mu_i) = (\ell_i - L_{i-1}, H_{i-1})$ if $\ell_i > L_{i-1}$.

REMARK. Let $R_i$ be drawn either above or below $T_S^R(i-1)$. In this case $E_i$ is drawn either to the right side of $R_i$ or to the right of $T_S^R(i-1)$.

Suppose $R_i$ is drawn either to the left or right of $T_S^R(i-1)$. Here $E_i$ is drawn either below $T_S^R(i-1)$ or below $R_i$.

EXAMPLE 2.6. In this example we explain the spiral-based algorithm for five given tiles.

1. We draw the first tile of width $\ell_1$ and height $h_1$ at position $(x_1, y_1)$, as shown in Figure 2.3.

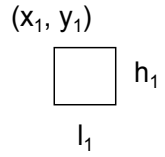


FIGURE 2.3
$T_S^R(1)$

2. The second tile is drawn above the first tile at position $(X_1, Y_1 - h_2)$ in such a way that its lower left vertex is the upper left vertex of $T_S^R(1)$ (Figure 2.4 (A)). On the basis of the difference in width of the first and the second tile, three cases are possible. An extra space is drawn if either $\ell_2 > L_1$ or $\ell_2 < L_1$, since in both cases the composition of tiles is not rectangular. To obtain a $T_S^R(2)$ an extra space $t$ is drawn as shown in Figure 2.4(B).

The position of $t$ in these two cases is $(x_2 + \ell_2, y_2)$ and $(X_1 + L_1, Y_1)$ respectively.

3. The third tile is drawn to the right of $T_S^R(2)$ in such a way that its upper left vertex is the upper right vertex of $T_S^R(2)$. The third tile with an extra space $t$ is shown in Figure 2.5.

As it is not feasible to illustrate all the possible cases, in Figure 2.5 it is assumed that $\ell_2 = \ell_1$. For all these cases, the position of the third tile is

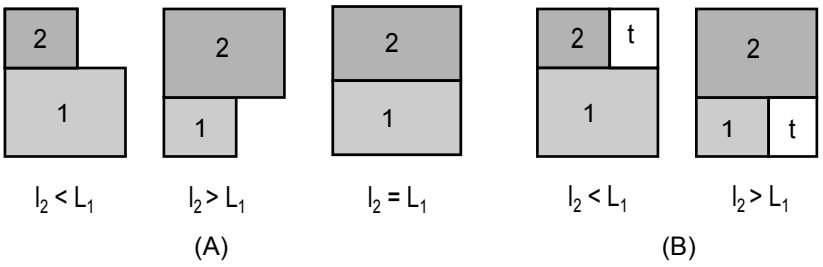


FIGURE 2.4
$T_S^R(2)$

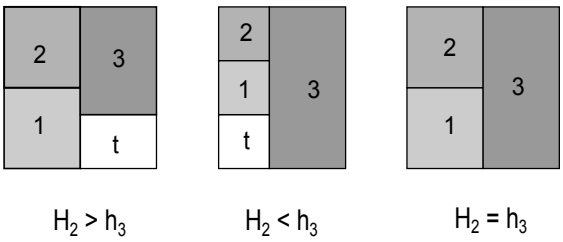


FIGURE 2.5
$T_S^R(3)$

$(X_2 + L_2, Y_2)$. Also, in the first two shapes of this figure, the position of $t$ is $(X_2, Y_2 + H_2)$ and $(x_3, y_3 + h_3)$ respectively.

4. The fourth tile is drawn below $T_S^R(3)$ such that its upper left vertex is the lower left vertex of $T_S^R(3)$. The fourth tile with an extra space $t$ is shown in Figure 2.6. In this figure, it is assumed that $\ell_2 = \ell_1$ and $H_2 = h_3$; however, other possible scenarios also exist. For all these scenarios, the position of the fourth tile is $(X_3, Y_3 + H_3)$. In the first two shapes of this figure, the position of $t$ is $(X_3 + L_3, Y_3)$ and $(x_4 + \ell_4, y_4)$ respectively.

5. The fifth tile is drawn to the left of $T_S^R(4)$ with its upper right vertex as the upper left vertex of $T_S^R(4)$. The fifth tile with an extra space $t$ is shown in Figure 2.7. In Figure 2.7, for demonstration it is considered that $\ell_2 = \ell_1$, $H_2 = h_3$ and $L_3 = \ell_4$ but there are other possible cases also. For all these cases, the position of the fifth tile is $(X_4 - \ell_5, Y_4)$. In the first two shapes of this figure, the position of $t$ is $(X_4, Y_4 + H_4)$ and $(x_5, y_5 + h_5)$ respectively.

It is clear from the steps of the above example that the tiles are arranged

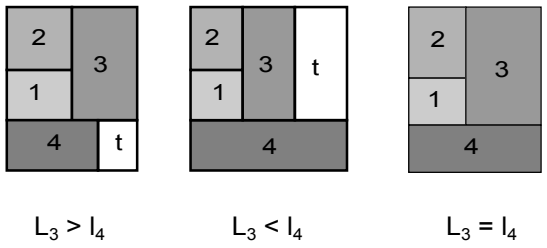


FIGURE 2.6
$T_S^R(4)$

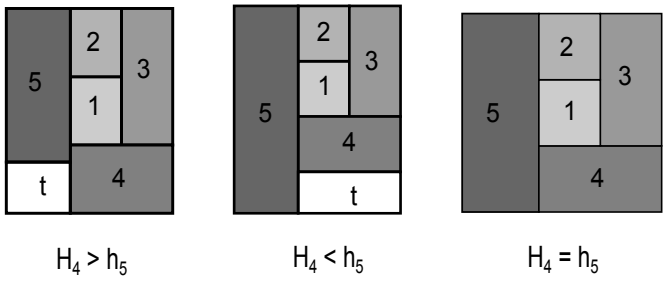


FIGURE 2.7
$T_S^R(5)$

along a spiral and that is why we refer to the algorithm as the "spiral-based algorithm".

### 2.1.3 INVARIANTS AND COVARIANTS

The concept of an *invariant* appeared naturally in geometry in response to the need for the classification of figures. In concrete terms, what distinguishes the following three sub-figures?

In Figure 2.8, the first sub-figure is a *straight line*, the second sub-figure a *rectangle* and the third one a *spiral*. A general line in the same plane intersects these figures in at most 1 and 2 points respectively for the first two figures, while it meets the spiral in infinitely many points. Although the numbers 1, 2 and $\infty$ do not fully describe these figures, yet they characterize and distinguish them from other geometrical shapes. Interestingly, these numbers remain unchanged if we apply certain geometric transformations, like isometries or scale changes. Thus the *number of intersection points* is an *invariant* with respect to a restricted set of transformations, which can be specified.

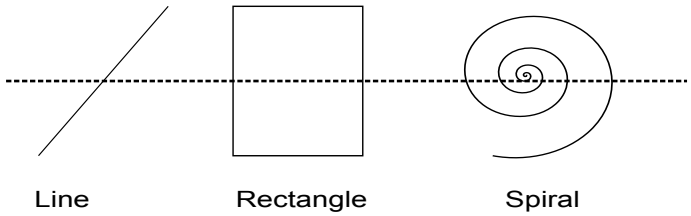


FIGURE 2.8

DEFINITION 2.7. If a number or a mathematical object associated with a geometric configuration remains unchanged with respect to a certain group of transformations then this number or mathematical object is said to be an *invariant* (with respect to the specified group of transformations).

For the shapes in Figure 2.8, the angles of intersection are certainly not invariant under *rotation*, but we can say how they behave. Therefore, the angles of intersection are *relative invariants* or *covariants*, with respect to *rotation*.

DEFINITION 2.8. If we know precisely how a number or a mathematical object associated with some geometric figure behaves with respect to a specified set of transformations, we call it a *covariant*.

Invariants and covariants can be used to understand the nature of a problem and to characterize its solutions.

COROLLARY 2.9. *A* $T_S^R(n)$ *has at most* $n-1$ *extra spaces.*

*Proof.* From the spiral-based algorithm, after drawing each tile except the first one, only one extra space may need to be drawn, therefore there can be at most $n-1$ extra spaces in the obtained $T_S^R(n)$. □

REMARK. From the spiral-based algorithm and Corollary 2.9, with respect to changes in the widths and heights of the tiles, $n$ is an *invariant* associated with the $T_S^R(n)$. However, depending on the difference in the width (or height) of the tiles, the number of extra spaces can be either $n-1$ or less than $n-1$. Therefore, the number of extra spaces varies in a known way as a function

of the sequence of tile sizes: it is a *covariant* with respect to modifications of the tile sizes.

### 2.1.4 EIGHT DIFFERENT SPIRALS

In this section, we provide a method for obtaining seven more $T_S^R$, which utilizes the direction (along which squares are arranged in a Fibonacci rectangle) of seven different Fibonacci spirals.

THEOREM 2.10. *Seven more sequences of Fibonacci rectangles and the golden rectangle (in the limit, as $n \to \infty$) can be obtained by making the following changes in Theorem 1.19:*

*By considering the anticlockwise movement as a replacement for clockwise movement; by interchanging position of the first and second squares and then following either the clockwise movement or the anticlockwise movement.*

*By considering the first and second squares side by side and then following either the clockwise or the anticlockwise movement; by interchanging position of the first and second squares and then following either the clockwise or the anticlockwise movement.*

*Proof.* Proof of this theorem follows from the proof of Theorem 1.19 and Figure 2.9.

In Figure 2.9, first consider the Fibonacci rectangle *spiral*1 which has the clockwise movement. The next Fibonacci rectangle is obtained by following the anticlockwise movement (*spiral*7). The next two Fibonacci rectangles are attained by interchanging position of the first and second square and then following the clockwise and the anticlockwise movement respectively (*spiral*3 and *spiral*5). The next Fibonacci rectangle is given by *spiral*2 where the first and second rectangle are arranged side by side. Similarly, from *spiral*2, the other 3 Fibonacci rectangles can be obtained (*spiral*4, *spiral*6 and *spiral*8). □

COROLLARY 2.11. *One can obtain seven more $T_S^R$ by using the same concept as used in theorem 2.10.*

*Proof.* By making all the changes (as given in theorem 2.10) in the spiral-based algorithm, we can obtain seven other $T_S^R$. □

Furthermore, the eight spirals will be mentioned with the names as given in Figure 2.9.

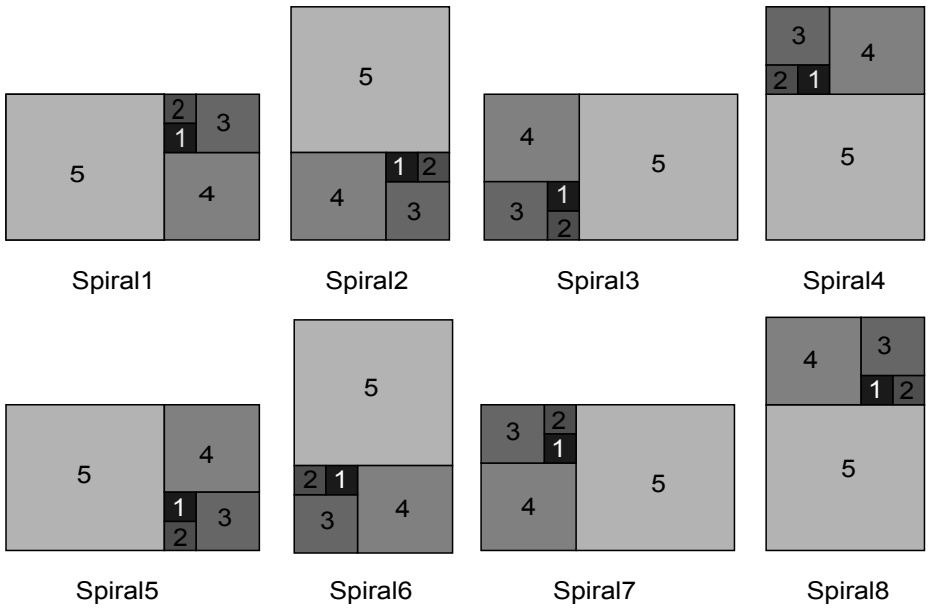


FIGURE 2.9
Eight sequences of the Fibonacci rectangles

REMARK. Associated with the $T_S^R$, with respect to the change of spirals, position of the tiles, position and area of these extra spaces are *covariants*.

For example, consider *spiral*4 and *spiral*6 in Figure 2.9. For both spirals, the position of the first and second tiles is the same but the position of the third tile is different. As we see, the position of the tiles is itself a covariant which depends on the choice of a spiral.

EXAMPLE 2.12. In this example, for a given data, 8 different $T_S^R$ are obtained and their graph is shown.

Given:

1. Five tiles with their areas $6, 24, 48, 48, 72$.
2. The ratio between their widths and heights, i.e., height of $i^{th}$ tile = $1.618\times$ (width of $i^{th}$ tile).

The widths and heights of given tiles are:
R1(1.93, 3.11), R2(3.86, 6.23), R3(5.45, 8.8), R4(5.45, 8.8), R5(6.67, 10.79).

Using the concept (or direction) of 8 spirals, eight computer generated $T_S^R$ are demonstrated in Figure 2.10.

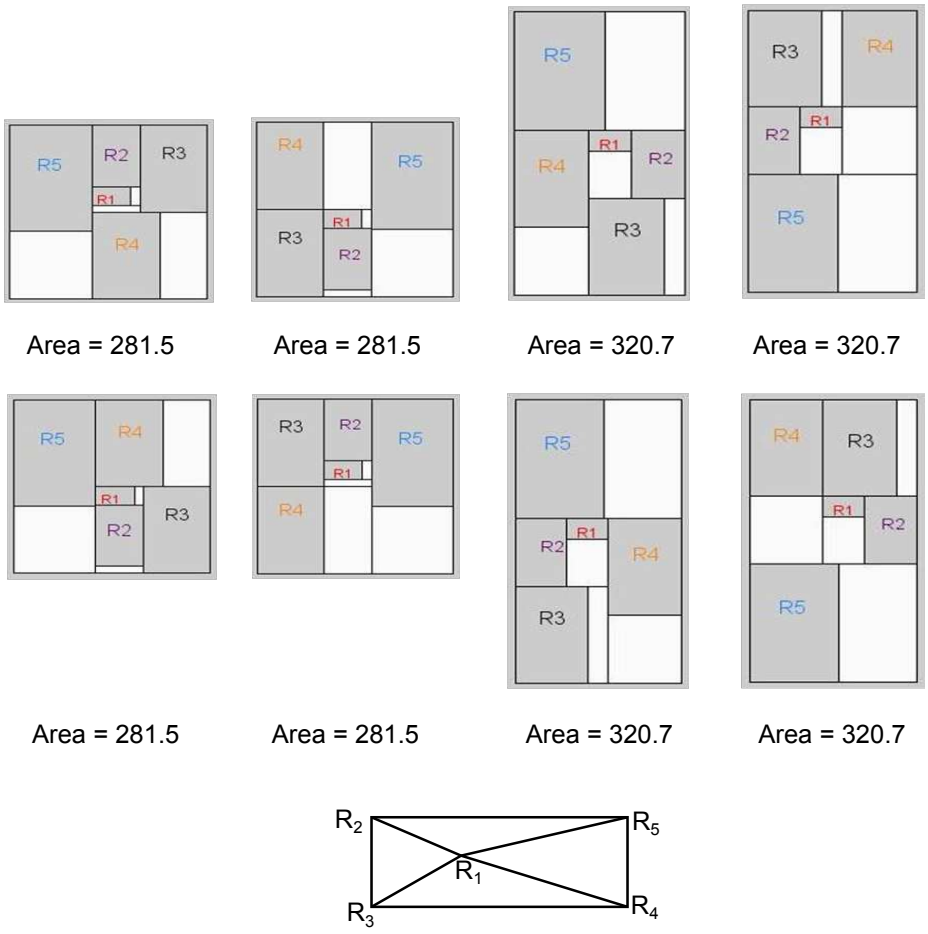


FIGURE 2.10
Eight different $T_S^R$ with their graph and respective areas

## 2.2. REDUCING AREAS OF THE EXTRA SPACES

In the preceding section, the obtained $T_S^R$ has some extra spaces also. For the connectivity and other purposes, an extra space seems to be helpful; for example, in a house, a corridor is needed as an extra space. But to acquire a compact shape, we may wish to reduce the sizes of these extra spaces.

In a $T_S^R$, the areas of the tiles always stay unchanged. The extra spaces are generated automatically, but their areas can be adjusted, i.e., it is possible to reduce or to enhance the total area of the $T_S^R$. To reduce the areas of these extra spaces in a $T_S^R$, the following two methods will be used:

*1. Choosing a different order of allocation,*

*2. Swapping the width and height of some tiles.*

REMARK. Only those methods are used for reducing the areas of the extra spaces by which the connectivity will remain preserved (refer to Corollary 3.14).

For example, it is possible to cross out all the extra spaces (refer to Section 2.2.3), however, as far as the applications are concerned, the extra spaces are valuable. Hence, the method explained in Section 2.2.3 won't be added to the spiral-based algorithm.

### 2.2.1 ORDER OF ALLOCATION

DEFINITION 2.13. In the spiral-based algorithm, the order in which the tiles are allocated is called the *order of allocation*.

In the spiral-based algorithm, $n!$ $T_S^R$ are obtained by changing the order of allocation of the tiles. Here each $T_S^R$ may have a different area because of the differences in areas of the extra spaces. Therefore, it is possible to pick one having least area among the $n!$ $T_S^R$.

REMARK. The total area of the extra spaces is a covariant of the $T_S^R(n)$, with respect to changes in the order of allocation.

#### 2.2.1.1 ARRANGING THE TILES IN ASCENDING ORDER AFTER THEIR AREAS

We are not ready to give a least area order of allocation for any $T^R$ but we are going to prove that the ascending order of allocation always gives "lesser area $T_S^R$" as compared to the descending order of allocation. Before moving to this result, it is important to describe the ascending order of allocation.

DEFINITION 2.14. The order of allocation in which all the tiles are arranged in the order of increasing areas is called the *ascending order of allocation*.

Before applying the spiral-based algorithm to the given tiles, the tiles are arranged in the ascending order.

EXAMPLE 2.15. Given three tiles $(R_1, R_2, R_3)$ with their respective areas(width and height) $9(3 \times 3)$, $24(6 \times 4)$ and $15(5 \times 3)$. For constructing

$T_S^R(3)$, one way is to use the same order as given initially (Condition A); another way is to arrange them in ascending order, i.e., 9, 15 and 24 (Condition B). The two $T_S^R(3)$ obtained with these conditions are shown in Figure 2.11. We see that Area(condition A) $>$ Area(condition B).

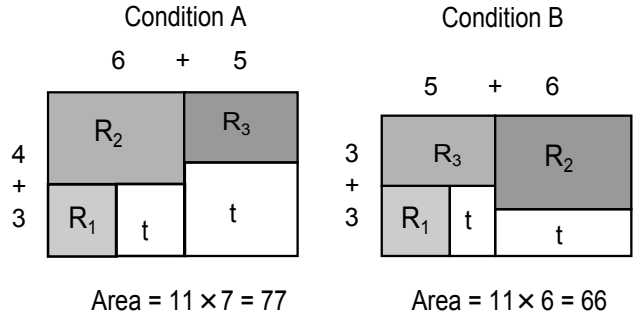


FIGURE 2.11
Explaining the ascending order of allocation

LEMMA 2.16. *If, for every even $i \le n$, we have $\ell_i \le L_{i-1} = \ell_{i-1} + L_{i-2}$ (in particular if the sequence $\{\ell_i\}$ is non-increasing) then we have, for $n$ even:*

$$L_n = \ell_{n-1} + \ell_{n-3} + \cdots + \ell_1 ,$$

*and for $n$ odd:*

$$L_n = \ell_n + L_{n-1} = \ell_n + \ell_{n-2} + \ell_{n-4} + \cdots + \ell_1 .$$

*If, for every odd $i \le n$ ($i \ne 1$), we have $h_i \le H_{i-1} = h_{i-1} + H_{i-2}$ (in particular if the sequence $\{h_i\}$ is non-increasing) then we have, for $n$ odd:*

$$H_n = h_{n-1} + h_{n-3} + \cdots + h_2 + h_1 ,$$

*and for $n$ even:*

$$H_n = h_n + H_{n-1} = h_n + h_{n-2} + h_{n-4} + \cdots + h_2 + h_1 .$$

*Proof.* We prove the result by mathematical induction.

Initially for $n = 1$, $L_1 = \ell_1$ and $H_1 = h_1$.

For $n = 2$, from the spiral-based sequence $L_2 = \max(L_1, \ell_2) = \max(\ell_1, \ell_2) = \ell_1$ and $H_2 = H_1 + h_2 = h_1 + h_2$.

For $n = 3$, $L_3 = L_2 + \ell_3 = \ell_1 + \ell_3$ and $H_3 = \max(H_2, h_3) = \max(h_1 + h_2, h_3) = h_1 + h_2$.

Hence the result is true for $n = 2$ and $n = 3$.

Let the result be true for all $n' < n$. Now we prove the result for $n$. If $n$ is odd, for even $n-1$ we have

$L_{n-1} = \ell_{n-2} + \ell_{n-4} + \cdots + \ell_1$ and

$H_{n-1} = h_{n-1} + h_{n-3} + h_{n-5} + \cdots + h_2 + h_1$.

From the spiral-based sequence for odd $n$, we have

$L_n = L_{n-1} + \ell_n = \ell_n + \ell_{n-2} + \ell_{n-4} + \cdots + \ell_1$ and

$H_n = \max(H_{n-1}, h_n)$

$= \max(h_{n-1} + h_{n-3} + h_{n-5} + \cdots + h_2 + h_1, h_n)$

$= h_{n-1} + h_{n-3} + h_{n-5} + \cdots + h_2 + h_1$ (because $\{h_i\}$ is a non-increasing sequence).

If $n$ is even, for odd $n-1$ we have

$L_{n-1} = \ell_{n-1} + \ell_{n-3} + \ell_{n-5} + \cdots + \ell_1$ and

$H_{n-1} = h_{n-2} + h_{n-4} + \cdots + h_2 + h_1$.

From the spiral-based sequence for even $n$, we have

$L_n = \max(L_{n-1}, \ell_n)$

$= \max(\ell_{n-1} + \ell_{n-3} + \ell_{n-5} + \cdots + \ell_1, \ell_n)$

$= \ell_{n-1} + \ell_{n-3} + \ell_{n-5} + \cdots + \ell_1$ (because $\{\ell_i\}$ is a non-increasing sequence) and

$H_n = H_{n-1} + h_n = h_n + h_{n-2} + h_{n-4} + \cdots + h_2 + h_1$.

Hence by mathematical induction, the result is true for $n$. □

THEOREM 2.17. *Suppose we have $n$ tiles, given by their areas and the assumption that the ratio between their widths and heights is fixed, i.e. the same for all the tiles. Then the area of $T_S^R(n)$ for the ascending order of allocation is at most equal to the area of $T_S^R(n)$ for the descending order of allocation.*

*Proof.* Let $R_i$ be the $i^{th}$ tile for $i = 1, 2, \ldots, n$, the area of $R_i$ is $S_i$ and the ratio of $h_i$ over $\ell_i$ is a fixed number $k$, i.e., $h_i = k \cdot \ell_i$ for every $i$. Also consider $S_1 \leq S_2 \leq \cdots \leq S_n$, which implies that $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_n$ and $h_1 \leq h_2 \leq \cdots \leq h_n$. The proof is structured according to the parity of $n$.

*1. Calculating the area in the ascending order for $n$ even and $n$ odd*

We have $A_n = H_n \cdot L_n$, where $H_n$ and $L_n$ are given by Corollary 2.4.

*2. Calculating the area in the descending order for $n$ even and $n$ odd*

For this order we have $A'_n = H'_n \cdot L'_n$, according to the following description.

Let $h'_i = h_{n-i+1}$ and $\ell'_i = \ell_{n-i+1}$. Now the sequences $\{\ell'_i\}$ and $\{h'_i\}$ are non-increasing, therefore using Lemma 2.16 we obtain the following results:

2.1 For even $n$, we have

$$H'_n = h'_n + h'_{n-2} + h'_{n-4} + \cdots + h'_2 + h'_1 = h_n + h_{n-1} + h_{n-3} + \cdots + h_1$$

and

$$L'_n = \ell'_{n-1} + \ell'_{n-3} + \cdots + \ell'_1 = \ell_n + \ell_{n-2} + \ell_{n-4} + \cdots + \ell_2 .$$

2.2 For odd $n$, we have

$$H'_n = h'_{n-1} + h'_{n-3} + \cdots + h'_2 + h'_1 = h_n + h_{n-1} + h_{n-3} + \cdots + h_2$$

and

$$L'_n = \ell'_n + \ell'_{n-2} + \ell'_{n-4} + \cdots + \ell'_1 = \ell_n + \ell_{n-2} + \ell_{n-4} + \cdots + \ell_1 .$$

Now we are going to prove that $A_n \le A'_n$. For $n$ even, we will do it by proving that $H_n \le H'_n$ and $L_n \le L'_n$. But for $n$ odd, this argument would not work since $L_n \ge L'_n$; in this case we will prove the result by proving $1/k \cdot H_n \le L'_n$ and $L_n \le 1/k \cdot H'_n$.

*3. $n$ is even*

3.1 To show that $H_n \le H'_n$ when $n$ is even;
From Corollary 2.4, we have

$$H_n = h_n + \max\left(h_{n-1}, \left(h_{n-2} + \max\left(h_{n-3}, \left(h_{n-4} + \cdots + \max(h_3, h_2 + h_1)\right)\right)\right)\right) ,$$

which implies that $H_n$ can have any one of the following values:

$h_n + h_{n-1}$
$h_n + h_{n-2} + h_{n-3}$
$h_n + h_{n-2} + h_{n-4} + h_{n-5}$
$\vdots$
$h_n + h_{n-2} + h_{n-4} + \cdots + h_1$

From 2.1, $H'_n = h_n + h_{n-1} + h_{n-3} + \cdots + h_1$ .

The maximum number of terms in $H_n$ and $H'_n$ is $\frac{1}{2}n + 1$. It is easy to see that, for each $i$, the $i^{th}$ term of $H_n$ is less than or equal to the $i^{th}$ term of $H'_n$, which implies that $H_n \le H'_n$.

3.2 To show that $L_n \le L'_n$ when $n$ is even;
From Corollary 2.4, we have

$$L_n = \max\left(\ell_n, \left(\ell_{n-1} + \max\left(\ell_{n-2}, \ell_{n-3} + \ldots \max(\ell_4, \ell_3 + \ell_2)\right)\right)\right)$$

and from 2.1, $L'_n = \ell_n + \ell_{n-2} + \ell_{n-4} + \cdots + \ell_2$ .

The maximum number of terms in $L_n$ and $L'_n$ is $\frac{1}{2}n$. It is easy to see that, for each $i$, the $i^{th}$ term of $L_n$ is less than or equal to the $i^{th}$ term of $L'_n$, which implies that $L_n \leq L'_n$.

*4. n is odd*

We know that each $h_i = k \cdot l_i$. From Corollary 2.4, we have

$1/k \cdot H_n = \max\left(\ell_n, \left(\ell_{n-1} + \max\left(\ell_{n-2}, \left(\ell_{n-3} + \cdots + \max\left(\ell_3, \ell_2 + \ell_1\right)\right)\right)\right)\right)$
and $L_n = \ell_n + \max\left(\ell_{n-1}, \left(\ell_{n-2} + \cdots + \max\left(\ell_4, \ell_3 + \ell_2\right)\right)\right)$.

From 2.2, we have

$1/k \cdot H'_n = \ell_n + \ell_{n-1} + \ell_{n-3} + \cdots + \ell_2$
and $L'_n = \ell_n + \ell_{n-2} + \ell_{n-4} + \cdots + \ell_1$.

4.1 To show that $1/k \cdot H_n \leq L'_n$ when $n$ is odd;

The maximum number of terms in $1/k \cdot H_n$ and $L'_n$ is $\frac{1}{2}(n-1)+1$. It is easy to see that, for each $i$, the $i^{th}$ term of $1/k \cdot H_n$ is less than or equal to the $i^{th}$ term of $L'_n$, which implies $1/k \cdot H_n \leq L'_n$.

4.2 To show that $L_n \leq 1/k \cdot H'_n$ when $n$ is odd;

The maximum number of terms in $L_n$ and $1/k \cdot H'_n$ is $\frac{1}{2}(n-1)+1$. It is easy to see that, for each $i$, the $i^{th}$ term of $L_n$ is less than or equal to the $i^{th}$ term of $1/k \cdot H'_n$, which implies $L_n \leq 1/k \cdot H'_n$.

When $n$ is either even or odd, it is proved that $A_n \leq A'_n$ which proves the theorem. □

REMARK. For each $T_S^R(n)$, there are $n!$ orders of allocation. To attain one with the least area, we compute the area of each $T_S^R$ and then compare all the areas to select a $T_S^R$ (out of $n!$ $T_S^R$) having the least area. However, this comparison is feasible only for small values of $n$. To reduce the complexity and to obtain a superior solution (not the best), we apply the ascending order of allocation to the given tiles before applying the spiral-based algorithm.

EXAMPLE 2.18. Consider 3 tiles $R_1, R_2, R_3$. First we allocate the given tiles in ascending order, i.e., $\{R_1, R_2, R_3\}$ and then allocate the same tiles in a different order, i.e., $\{R_1, R_3, R_2\}$.

*Let the width and height of the tiles be:* $\ell_1 = 3$, $h_1 = 1$, $\ell_2 = 2$, $h_2 = 4$, $\ell_3 = 3$, $h_3 = 4$.

For the ascending order, the area of $T_S^R(3)$ is 30 while for the other order the area is 25 (cf. Figure 2.12). Clearly, Area($1^{st}$ order) $>$ Area($2^{nd}$ order) which shows:

1. The ascending order of allocation is not the least area order of allocation for all $T_S^R$,

2. The area of a $T_S^R$ is a *covariant* with respect to change in an order of allocation.

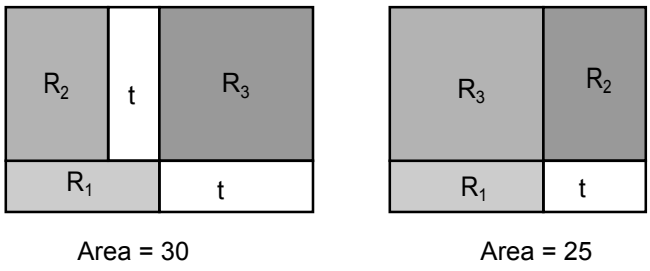


FIGURE 2.12
Comparing the areas for two different orders of allocation

As the extra spaces have areas which depend largely on those of the tiles, it is difficult to imagine an order of allocation which would give a least area $T_S^R$ for any values of areas of the tiles i.e. *a least area order of allocation defined universally for any* $T_S^R$. This leads to an open problem which is described as follows:

*Open problem 1: Does there exist a least area order of allocation valid universally for any* $T_S^R$ *? If yes, how can one describe it ?*

### 2.2.2 SWAPPING THE WIDTH AND HEIGHT OF A TILE

After applying ascending order of allocation to the given tiles, the areas of the extra spaces can be reduced by interchanging the width and height of some tiles.

DEFINITION 2.19. $A_O$ is the area of the extra space drawn after drawing $i^{th}$ tile ($i > 1$) with width $\ell_i$ and height $h_i$ (i.e., as given initially).

$A_I$ is the area of the extra space drawn after drawing $i^{th}$ tile ($i > 1$) with width $h_i$ and height $\ell_i$ (i.e., the width and height are swapped).

In the spiral-based algorithm, for interchanging the width and height of a tile, consider each tile one by one. Suppose $i^{th}$ tile has to be drawn with width $\ell_i$ and height $h_i$. Calculate $A_O$ and $A_I$ for the $i^{th}$ tile. If $A_O > A_I$ then interchange the width and height of the $i^{th}$ tile, otherwise not.

The dimension of the extra spaces in the spiral-based algorithm can be reduced by swapping the width and height of the tiles. But if all the combinations of different tiles except first one (whose width and height would be swapped) are taken into account, there are $2^{n-1}$ combinations.

Picking a combination having the least area among all the possible combinations is feasible for the smaller values of $n$ but as the value of $n$ increases it become complex exponentially. Therefore, to reduce the complexity, we first compute the value of $A_O$ and $A_I$ corresponding to each tile and if $A_O > A_I$, we swap its width and height (see Example 2.21).

DEFINITION 2.20. For constructing a $T_S^R$, if the widths and heights of all the tiles are considered as given initially, then the area of the $T_S^R$ will be denoted by $A_{initial}$.

And if the widths and heights of the tiles are interchanged (under condition "if required" defined below), then the area of the $T_S^R$ will be denoted by $A_{final}$.

*If required* condition means that the width and height of a tile are interchanged only when $A_O > A_I$ for that tile.

EXAMPLE 2.21. Consider 3 tiles $R_1, R_2, R_3$. Let the areas of given tiles in the ascending order be $9(3\times 3)$, $15(5\times 3)$ and $24(6\times 4)$. We construct the first $T_S^R(3)$ without swapping the width and height of any tile. To construct other $T_S^R(3)$, we compare $A_O$ and $A_I$ for each tile. For the $2^{nd}$ and $3^{rd}$ tile, $A_O > A_I$, therefore, we swap the width and height of the $2^{nd}$ and $3^{rd}$ tile, i.e., the width and height will be $3\times 3$, $3\times 5$ and $4\times 6$ (cf. Figure 2.13).

The above case demonstrate that $A_{initial}$ is not always greater than $A_{final}$.

In many examples $A_{initial} > A_{final}$, therefore, the width and the height of the tiles will be interchanged, if required.

REMARK. We modify the spiral-based algorithm by adding the following two steps:

1. Before drawing the tiles, arrange them in the ascending order according to their areas.

2. Before drawing each tile, interchange its width and height, if required.

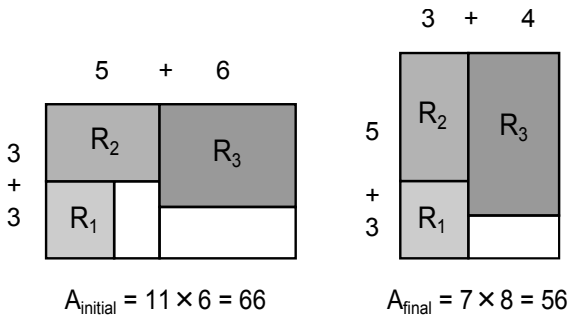


FIGURE 2.13
Comparing $A_{initial}$ and $A_{final}$

### 2.2.3 NO EXTRA SPACE

If extra spaces are not at all essential, they can be removed by adjusting the ratio between the width and height of each tile.

EXAMPLE 2.22. Consider 3 tiles with widths and heights $2 \times 3$, $3 \times 3$ and $3 \times 5$. For these tiles, a $T_S^R(3)$ is shown in Figure 2.14 (A) having area 36. By altering the ratio between the width and height of the tiles, the modified widths and heights are $2 \times 3$, $2 \times 4.5$ and $2 \times 7.5$. For the newly obtained widths and heights, a $T_S^R(3)$ is given in Figure 2.14 (B) having area 30. Clearly, Figure 2.14 (B) does not have any extra space.

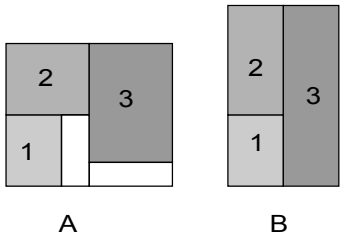


FIGURE 2.14
No extra space

REMARK. The size of extra spaces can also be reduced by picking a $T_S^R$ (out of eight $T_S^R$) having minimum area.

## 2.3. Related Work

DEFINITION 2.23. Given $n$ rectangles, the *rectangle packing problem* consists of finding a rectangle of *minimum area* that contains all the given rectangles without any overlapping.

If only the arrangement is considered, then the rectangle packing problem has some similarity with the $T^R$.

The rectangle packing problem seeks to acquire an enclosing rectangle of minimum area. After obtaining a best enclosing $T^R$ (from the point of view of connectivity), some techniques which are used in rectangle packing problem can be utilized to reduce the size of some extra spaces such that the *number of extra spaces* and *connectivity* remains preserved.

Korf [24] introduced a method for reducing the sizes of extra spaces by *slicing* them.

If merely rectangles are considered, then the $T^R$ has some similarity to *Shape grammar* also.

DEFINITION 2.24. *Shape grammar* is a procedure for generating different geometric shapes.

Shape grammars have been studied in different areas particularly in computer-aided architectural design, as a method for providing a formalism to create new designs. A shape grammar consists of the shape rules and a generation engine that selects and processes the rules. A shape rule defines how an existing (part of a) shape can be transformed from one to another (See Stiny [38]).

From shape grammars, some ideas can be used to obtain different rectangular and other shaped tilings which should be best from the point of view of connectivity.

In this chapter, eight different $T_S^R$ have been obtained. In addition some methods have been introduced to reduce the size of extra spaces. This concept of obtaining the $T_S^R$ can be further extended to attain different shaped tilings but in the next chapter, our main focus will be on proving that the $T_S^R$ is best from the point of view of connectivity.

# CHAPTER 3

# CONNECTIVITY OF A RECTANGLE TILING

This chapter focuses on proving that a $T_S^R$ is one of the best $T^R$ from the point of view of connectivity.

## 3.1 VIRTUAL PARTS AND ADJACENCY

The notion of adjacency among rectangles has already been defined in Chapter 1. In this section we discuss a more general concept of adjacency among tiles via extra spaces.

DEFINITION 3.1. A *virtual part* of a tiling (or a tile) is an extra space which shares a full wall with that tiling (or tile).

If a *virtual part* of a tile is considered to be merged into that tile then we will say that the *virtual part* has been made a *real part* of the tile.

EXAMPLE 3.2. For the first shape of Figure 3.1, $t_1$, $t_2$ and $t_3$ are extra spaces and each extra space shares a full wall with the second, third and fourth tiles respectively. Hence, $t_1$, $t_2$ and $t_3$ are the virtual parts of the second, third and fourth tiles respectively.

REMARK. A virtual part can be made real by adjusting the ratios $\ell_i/h_i$ (cf. Section 2.2.3). In other words, the extra spaces can be either shrunk or expanded by inserting some springs (cf. Kalay [23], Chapter 14, *Physically based space allocation*).

Consider a tile having width $a$ and height $b$ and its virtual part having width $c$ and height $d$, which we denote by $R_{a,b}$ and $V_{c,d}$ respectively. The transformation of a virtual part of a tile into a *real part* is done by one of the following operations:

$$R_{a,b} + V_{c,b} = R_{a+c,b}$$

$$R_{a,b} + V_{a,c} = R_{a,b+c}\,.$$

EXAMPLE 3.3. In the second shape of Figure 3.1, $t_1$ and $t_3$ have been transformed into real parts of the second and fourth tiles respectively, while keeping $t_2$ as a virtual part of the third tile.

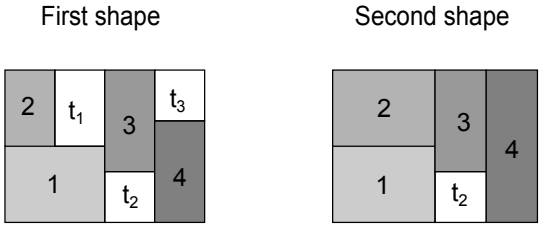


FIGURE 3.1
Making virtual parts real

DEFINITION 3.4. A tile (say $K$) is said to be *directly adjacent* to a tile (say $R$) if $K$ shares a wall or a sub-wall with $R$.

A tile $K$ is said to be *adjacent* to a tile $R$ *via virtual parts* if any of the following holds:

1. Either the virtual part of $K$ is directly adjacent to $R$ or the virtual part of $R$ is directly adjacent to $K$

2. The virtual part of $K$ is directly adjacent to the virtual part of $R$.

Adjacency between two virtual parts or adjacency between a virtual part and a tile is same as direct adjacency between two rectangles (for the definition of direct adjacency refer to Definition 1.9).

For example, in the first shape of Figure 3.1, $t_1$ is adjacent to the third tile, consequently, we also consider the second tile to be adjacent to the third tile.

In Chapter 1, the notion of adjacency among two rectangles has already been defined. Since in a $T^R$ an extra space is always a rectangle, the adjacency between an extra space and a tile is the same as the adjacency between two tiles. But whenever an extra space is present between two tiles, further discussion is required.

Consider there are $n$ tiles, $R_{a_i,b_i}$ is the $i^{th}$ tile and $E_{c_i,d_i}$ is the required extra space. The following 4 cases define the adjacency among these tiles via extra spaces:

1. *When $E_{c_i,d_i}$ is a virtual part of $R_{a_i,b_i}$*

EXAMPLE 3.5. In Figure 3.2, $t$ is a virtual part of the third tile; $t$ is adjacent to the first tile which implies that the third tile is adjacent to the first one.

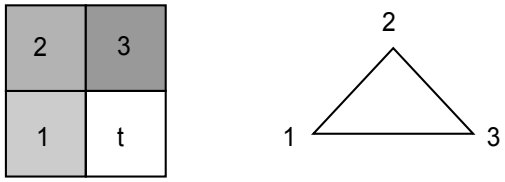


FIGURE 3.2
When $E_{c_i,d_i}$ is a virtual part of $R_{a_i,b_i}$

2. *When $E_{c_i,d_i}$ is not a virtual part of $R_{a_i,b_i}$ but is a virtual part of $R_{a_j,b_j}$*

EXAMPLE 3.6. In Figure 3.3, $t$ is drawn after drawing second tile but it is a virtual part of the first tile. And $t$ is adjacent to the third tile, this implies that the first tile is adjacent to the third tile.

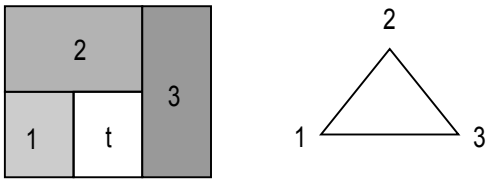


FIGURE 3.3
When $E_{c_i,d_i}$ is not a virtual part of $R_{a_i,b_i}$ but of $R_{a_j,b_j}$

3. *When $E_{c_i,d_i}$ is a virtual part of more than one tile*

Here there exist $k > 1$ tiles such that either $c_i = a_1 + a_2 + \cdots + a_k$ or $d_i = b_1 + b_2 + \cdots + b_k$. For this condition, we divide the extra space $E_{c_i,d_i}$ between $k$ parts such that $E_{c_i,d_i}$ will be a virtual part of the $k$ tiles.

EXAMPLE 3.7. In the first shape of Figure 3.4, $t$ is not sharing a *full wall* with any of the tiles. Whereas in the second shape, where $t$ is divided into 2 parts ($t_1$ and $t_2$), it shares a *full wall* with the first and second tiles. Hence $t_1$ and $t_2$ are the virtual parts of the second tile and first tile respectively.

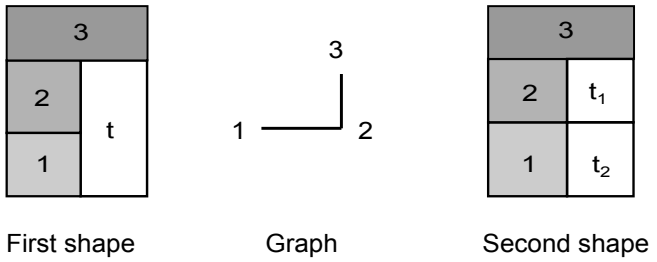


FIGURE 3.4
When $E_{c_i,d_i}$ is a virtual part of more than one tile

DEFINITION 3.8. If $E_{c_i,d_i}$ shares a full wall with $E_{c_j,d_j}$ and $E_{c_j,d_j}$ is a virtual part of $R_{a_k,b_k}$ then we say $E_{c_i,d_i}$ is a virtual part of $R_{a_k,b_k}$.

4. *When an extra space $E_{c,d}$ shares a full wall with $r \geq 0$ tiles and $s > 0$ extra spaces*

In this case, either $c = (a_1 + a_2 + \cdots + a_r) + (c_1 + c_2 + \cdots + c_s)$ or $d = (b_1 + b_2 + \cdots + b_r) + (d_1 + d_2 + \cdots + d_s)$. Also, a tile can have more than one virtual parts.

For this condition, we divide the extra space $E_{c,d}$ between $r + s$ parts such that $E_{c,d}$ will be a virtual part of the $r + s$ tiles.

EXAMPLE 3.9. In the first shape of Figure 3.5, $t''$ is not sharing a *full wall* with any of the tiles or extra spaces. Whereas in the second shape, where $t''$ is divided in 3 parts ($t''_1$, $t''_2$ and $t''_3$), it shares a *full wall* with $t$, $t'$ and the first tile. Hence $t''_1$, $t''_2$ and $t''_3$ are the virtual parts of the third tile, first tile and second tile respectively. Clearly, second and third tiles have two virtual parts.

REMARK. In the first shape of Figure 3.4 and 3.5, $t$ and $t''$ are virtual parts of the $T^R(2)$ and the $T^R(3)$ respectively.

Also, for a $T^R_S$, $E_{c_i,d_i}$ is a virtual part of either $R_{a_i,b_i}$ or $T^R_S(i-1)$.

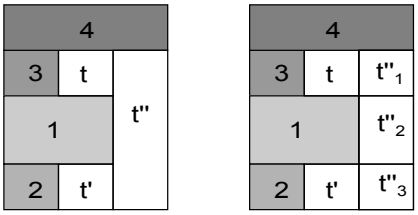


FIGURE 3.5
When $E_{c_i,d_i}$ shares a full wall with tiles and extra spaces

COROLLARY 3.10. *The adjacency among the tiles remains unaffected even when virtual parts are transformed into real parts.*

*Proof.* As defined above, if a virtual part of a tile $K$ would be adjacent to a tile $R$ or virtual part of $R$, then $K$ would become adjacent to $R$. Here, $K$ is adjacent to $R$ by means of a virtual part. If a virtual part of $K$ is made a real part of $K$ then $K$ would be directly adjacent to $R$. It means that in both cases, $K$ would be adjacent to $R$.

For example, in the first shape of Figure 3.1, the second tile is adjacent to the third tile via $t_1$ however in the second shape of Figure 3.1, the second tile is directly adjacent to the third tile. □

We denote the adjacency graph of $T^R$ and $T_S^R$ by $G^R$ and $G_S^R$ respectively.

COROLLARY 3.11. *There always exist a path between any two vertices of $G^R$; hence $G^R$ is connected.*

*Proof.* From condition 4 of Section 1.2.1, each new tile must be adjacent to at least one of the previously drawn rectangle. Now each new tile would be adjacent to at least one of the previously drawn tile either directly or via an extra space. Therefore by induction, each time a new tile is added to say $T^R(i-1)$, at least one edge gets added to a connected adjacency graph $G^R(i-1)$. Hence the new adjacency graph $G^R(i)$ is connected. □

COROLLARY 3.12. *For a $T_S^R(n)$ when $n > 4$, the new $n^{th}$ tile $R_n$ is adjacent to $R_{n-1}$, $R_{n-3}$ and $R_{n-4}$.*

*Consider virtual parts as real parts of the corresponding tiles.*

*If $R_n$ is drawn either to the left or to the right of $T_S^R(n-1)$ then*

$$h_n = h_{n-1} + h_{n-3} + h_{n-4}$$

*If $R_n$ is drawn either above or below $T_S^R(n-1)$ then*

$$l_n = l_{n-1} + l_{n-3} + l_{n-4}\,.$$

*Proof.* For a $T_S^R(n)$ when $n > 4$, $R_n$ is adjacent to $R_{n-1}$, $R_{n-3}$ and $R_{n-4}$ either directly or by considering its virtual part real. For example, in all $T_S^R(5)$ of Figure 2.10, $R_5$ is adjacent to $R_1$, $R_2$ and $R_4$. Clearly, every new tile is adjacent to three previously drawn tiles.

If an extra space is required after drawing $R_n$ (either to the left or to the right of $T_S^R(n-1)$), then it will be drawn either below $R_n$ or below $T_S^R(n-1)$. In both cases, if the virtual parts are transformed into real parts of the corresponding tiles, then $R_n$ will share a full wall (for the definition of sharing a full wall refer to Definition 1.8) only with $R_{n-1}$, $R_{n-3}$ and $R_{n-4}$ because $R_n$ is only adjacent to $R_{n-1}$, $R_{n-3}$ and $R_{n-4}$ when $n > 4$. This implies

$$h_n = h_{n-1} + h_{n-3} + h_{n-4}$$

Similarly, if $R_n$ is drawn either above or below $T_S^R(n-1)$, then

$$l_n = l_{n-1} + l_{n-3} + l_{n-4}. \qquad \square$$

Let $E^R(n)$ and $E_S^R(n)$ be the number of edges for the adjacency graphs $G^R(n)$ and $G_S^R(n)$ respectively.

THEOREM 3.13. *If $n > 3$ then $E_S^R(n) = 3n-7$ and the number $D_S^R(n)$ of triangles in $G_S^R(n)$ is equal to $D_S^R(n) = 2n-6$.*

*Proof.* From the point of view of adjacency, it does not matter whether the extra spaces are virtual or real parts (cf. Corollary 3.10).

In every shape of Figure 2.10, $E_S^R(n) = 3n-7$. If $n$ is increased by one, then from Corollary 3.12, a new tile would be adjacent to 3 existing tiles, and hence $E_S^R(n+1) = 3(n+1)-7$. By induction, for $n > 3$ we have $E_S^R(n) = 3n-7$.

Also, for all the shapes of Figure 2.10, the number of triangles[5]) in the corresponding $G_S^R(n)$ is $2n-6$. If we add a new tile to $T_S^R(n-1)$, then $G_S^R(n)$

[5]) A *triangle* is a cycle of length 3.

will be $G_S^R(n-1)$ with 3 more edges, because the new tile will be adjacent to 3 existing tiles. These 3 edges always form 2 new triangles and therefore by induction for $n > 3$, the number of triangles is $2n-6$.

It is noteworthy that for $n = 3$, $E_S^R(n) = 3n-6$ and the number of triangles is $2n-5$. □

REMARK. The adjacency among the tiles remains unchanged even by transforming the virtual parts into real parts, therefore with respect to modification of the tile sizes, $G^R$ is an *invariant*.

COROLLARY 3.14. *$G_S^R(n)$ remains preserved even after applying any one of the 2 methods given in Section 2.2 and the method given in Section 2.2.3.*

*Proof.* By applying any one of the two methods as given in Section 2.2, the sizes of the virtual parts get either increased or decreased. But if the virtual parts are turned into real parts, then from the point of view of adjacency, it does not matter whether the sizes of the virtual parts are changed or not. Therefore, after applying these two methods, the $G_S^R(n)$ remains preserved.

By applying the third method, there would not be any extra space in a $T_S^R$; obviously the $G_S^R(n)$ remains preserved. □

## 3.2. ADJACENT SIDES, ADJACENT NUMBERS AND ADJACENT SUM

The results given in this section will be further used to prove an important result in the next section (see Theorem 3.24 and Corollary 3.25).

A $T^R$ consists of rectangles and $T^R$ is also a rectangle having four sides.

DEFINITION 3.15. Every side of a $T^R$ is called an *adjacent side*.

Suppose $a$ and $b$ are the width and height of a $T^R$ while $\ell_i$ and $h_i$ denote as usual the width and the height of a tile $R_i$. If the virtual parts are made into real parts of the corresponding tiles then we can write

$a = a_1 + a_2 + \cdots + a_k = a'_1 + \cdots + a'_{k'}$ and $b = b_1 + b_2 + \cdots + b_l = b'_1 + \cdots + b'_{l'}$, where the $a_i$ and the $a'_i$ are subsets of the $\ell_i$, and the $b_i$ and the $b'_i$ are subsets of the $h_i$. In each case the two decompositions correspond to opposite sides of the rectangle $T^R$.

The values of $k$ and $k'$, or $l$ and $l'$ can be either the same or different. These numbers $k$, $k'$ and $l$, $l'$ are called the *adjacent numbers* associated with each adjacent side.

To fix the notation, let $k_1$, $k_2$, $k_3$ and $k_4$ be the adjacent numbers of a $T^R$ corresponding to the left, upper, right and lower adjacent sides respectively.

EXAMPLE 3.16. Refer to Figure 3.6 where the adjacent numbers of all the adjacent sides of $T^R(4)$ are calculated.

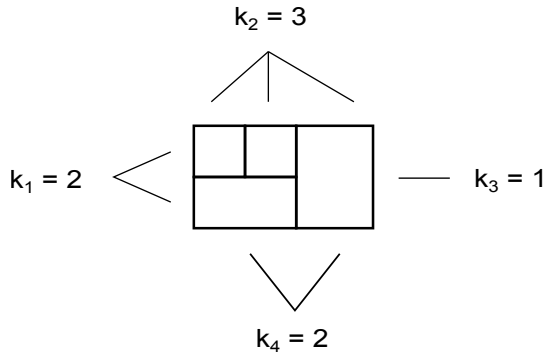


FIGURE 3.6
Adjacent numbers for a $T^R(4)$

DEFINITION 3.17. The total number of sides of the tiles of a $T^R$ facing the outer world are collectively called the *adjacent sum*, i.e., $S^R = k_1+k_2+k_3+k_4$.

THEOREM 3.18. *For a* $T^R(n)$, *the following equality holds:*

$$E^R(n) + S^R(n) = 3n + 1\,.$$

*Proof.* This result is proved by induction. First, we check that it holds for $n = 1$, then we shall assume that it is true for $n$ and deduce it for $n+1$.

For $n = 1$, the result is obvious as $E^R(1) = 0$ and $S^R(1) = 4 = 3n + 1$. Now suppose $E^R(k) + S^R(k) = 3k + 1$.

If $R_{k+1}$ is added to any adjacent side of $T^R(k)$ (say *left*), then $R_{k+1}$ will be adjacent to all the tiles on the left side. Therefore, by adding $R_{k+1}$, the following changes occur (refer to Figure 3.7 where virtual part of $R_{k+1}$ if exist, is considered real):

1. $k_1$ becomes one

Due to this change, $E^R(k)$ gets increased by $k_1$, i.e., $E^R(k+1) = E^R(k)+k_1$.

2. $k_2$ and $k_4$ are increased by one

Because of this change, $S^R(k)$ gets increased by 2 and reduced by $(k_1-1)$, i.e., $S^R(k+1) = S^R(k) + 2 - (k_1 - 1)$.

3. $k_3$ remains unchanged.

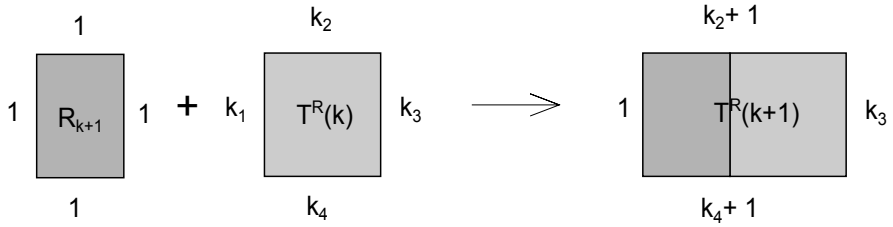


FIGURE 3.7

Changes in the adjacent numbers when a tile is added to the left of $T^R(k)$

Thus, $E^R(k+1) + S^R(k+1) = E^R(k) + k_1 + S^R(k) + 2 - (k_1 - 1) = E^R(k) + S^R(k) + 3 = 3k + 1 + 3 = 3(k+1) + 1$, as required. □

LEMMA 3.19. *For a* $T_S^R(n)$ *when* $n > 3$, $S^R(n) = 8$ *and* $\{k_1, k_2, k_3, k_4\}$ *is a cyclic permutation of* $\{3, 2, 2, 1\}$.

*Proof.* From Theorem 3.13 we have $E_S^R(n) = 3n - 7$ when $n > 3$. From Theorem 3.18 we have $E^R(n) + S^R(n) = 3n + 1$. It follows that for a $T_S^R(n)$ we have $S^R(n) = (3n+1) - E_S^R(n) = (3n+1) - (3n-7) = 8$ when $n > 3$.

In each $T_S^R$ of Figure 2.10, the adjacent numbers for the adjacent sides are given by the set $\{3, 2, 2, 1\}$. Also, from Corollary 3.12, a new tile is adjacent to 3 existing tiles. Therefore, if a new tile is added, the following transformations occur in the set $\{3, 2, 2, 1\}$ :

a. 3 becomes 1 because a new tile has been added to this side

b. 2 becomes 3 because this side is adjacent to the side to which the new tile has been added

c. The next 2 remains unchanged because this side is opposite to the side to which the new tile has been added

d. 1 becomes 2 because this side is adjacent to the side to which the new tile has been added.

*These four points are explained using Figure 3.7.*
Hence, after adding a new tile the same set is established, i.e., $\{3,2,2,1\}$. □

REMARK. Associated with a $T_S^R$, from Lemma 3.19, $S^R$ and the set $\{3,2,2,1\}$ are *invariants* with respect to the addition of a new tile, however the adjacent number for each adjacent side changes when a new tile is added (cf. Figure 3.7), therefore each adjacent number is a *covariant*.

## 3.3. BEST $T^R$ ON THE BASIS OF CONNECTIVITY

In this section, we proceed with some properties of planar graphs which will be used to prove further results.

### 3.3.1 PLANAR GRAPHS

The theory of planar graphs is an extensive field to explore and has many interesting applications in various other fields.

DEFINITION 3.20. A *planar graph* is a graph that can be embedded in the plane in such a way that no two edges cross each other. If a graph $G$ is embedded in this way, then the points of the plane not on $G$ are partitioned into open sets called *faces*.

Euler discovered the basic relationship between the numbers of vertices, edges and faces.

THEOREM 3.21 (Euler's Formula). *Let $G$ be a connected planar graph with $n$ vertices, $m$ edges, and $l$ faces. Then*

$$n - m + l = 2$$

.

*Proof.* For a proof refer to Diestel [10], Theorem 4.2.9. □

From Euler's formula, we have the following lemma.

LEMMA 3.22. *If $G$ is a (connected) simple, finite planar graph with $n$ vertices ($n \geq 3$), then $G$ has at most $3n-6$ edges. Furthermore, if $G$ contains no triangles, then $G$ has at most $2n-4$ edges.*

*Proof.* For a proof refer to Diestel [10], Corollary 4.2.10. □

DEFINITION 3.23. Let $e$ be an edge with endpoints $\{u, v\}$ in a graph $G$. *Subdividing the edge $e$* means that a new vertex $w$ is added to $V_G$, and that edge $e$ is replaced in $E_G$ by an edge $e'$ with endpoints $\{u, w\}$ and an edge $e''$ with endpoints $\{w, v\}$.

Two graphs $G$ and $H$ are *homeomorphic* if there is an isomorphism from a subdivision of $G$ to a subdivision of $H$ (Jonathan and Yellen [16], Chapter 7).

From Lemma 3.22, it follows that the two graphs $K_5$ and $K_{3,3}$ are non-planar. Kuratowski proved that these two graphs are the only barriers to planarity.

THEOREM (Kuratowski's Theorem [1930]). *A graph is planar if and only if it contains no sub-graph homeomorphic to $K_5$ or to $K_{3,3}$.*

*Proof.* For a proof of this theorem and details about planar graphs we refer to Jonathan and Yellen [16], Chapter 7 and Theorem 7.4.1. □

### 3.3.2 BEST RECTANGLE TILING

By using some of the results of Sections 3.1, 3.2 and 3.3.1, we prove that the spiral-based algorithm provides one of the best $T^R$ from the point of view of connectivity.

THEOREM 3.24. *For any $T^R(n)$, we have $E^R(n) < 3n-6$ provided that $n > 3$.*

*Proof.* $G^R$ is always planar because overlapping among the tiles is not admitted. It is simple (because of symmetry, i.e., if $A$ is adjacent to $B$ then $B$ is adjacent to $A$) and also connected (cf. Corollary 3.11).

From here on, if a planar graph is mentioned, it means a simple, connected, planar graph and if an edge is mentioned it means an edge in the corresponding $G^R$.

Since $G^R(n)$ is planar, it follows from Lemma 3.22 that $E^R(n) \leq 3n-6$. Therefore, it is enough to prove that $E^R(n) \neq 3n-6$.

We proceed by induction starting from the case where $n = 4$. In this case the only graph having $3n - 6 = 6$ edges is the complete graph $K_4$ illustrated in Figure 3.8.

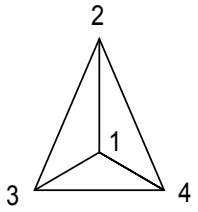


FIGURE 3.8
Graph having 6 edges

Thus we are led to prove that $G^R(4)$ is not isomorphic to $K_4$. If it were, then each tile would be adjacent to 3 other tiles (as shown in Figure 3.8). Now, if the first tile is adjacent to the remaining three tiles, there are three possibilities:

1. *Three walls of the first tile are shared*

Clearly, two of the other tiles must be situated on opposite sides of the first tile. Hence these tiles cannot touch each other and they are not adjacent (see Figure 3.9).

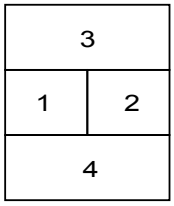


FIGURE 3.9
Two tiles on opposite sides of the first tile

**Note:** *For our purpose in all the figures the extra spaces are considered as real parts of the corresponding tiles.*

2. *Only two walls of the first tile are shared*

As in the preceding case, we can assume that these walls are not opposite. Then we can assume, without loss of generality, that two of the tiles are

situated on the right of the first tile $R_1$, one above the other (say $R_3$ on top of $R_4$), and that the remaining tile $R_2$ lies below $R_1$ or to the left of $R_1$ (as in Figure 3.10). Then $R_2$ is at a certain distance from $R_3$.

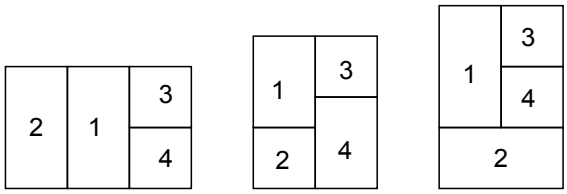


FIGURE 3.10
$T^R(4)$ where only two walls of the first tile are being shared

3. *Only one wall of the first tile is shared*

In this case, there is only one possibility, as shown in Figure 3.11. Clearly, two of the tiles are too distant from each other to be adjacent.

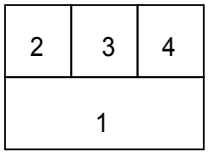


FIGURE 3.11
$T^R(4)$ where only one wall of the first tile is being shared

Thus we have shown that $E^R(4) \neq 6$ for any rectangle tiling of order 4. We now make the induction hypothesis that $E^R(k) \neq 3k-6$ i.e. $E^R(k) \leq 3k-7$ for any rectangle tiling of order $k$ (where $k \geq 4$). We need to prove that $E^R(k+1) \neq 3(k+1)-6$.

Suppose $E^R(k) = 3k-7$. Then to have $E^R(k+1) = 3(k+1)-6$, four edges need to be added to $G^R(k)$.
From Theorem 3.18 we have $E^R(k) + S^R(k) = 3k+1$ implies $S^R(k) = (3k+1)-(3k-7) = 8$ i.e. $k_1+k_2+k_3+k_4 = 8$.
If we add 4 edges to $G^R(k)$, the adjacent number for at least one adjacent side should be 4. Let's say $k_1 = 4$ (cf. Conditions 5 and 6, Section 1.2.1).

Now, $S^R(k) = 8$ and $k_1 = 4$ implies $k_2+k_3+k_4 = 4$, i.e., $\{k_2, k_3, k_4\} = \{2, 1, 1\}$. This set is possible only when $k = 2$, but in our induction hypothesis

we are assuming that $k \geq 4$. Hence it is not possible to add 4 edges to $G^R(k)$ to obtain $G^R(k+1)$, when $G^R(k)$ has $3k-7$ edges.
If $E^R(k)$ is even smaller, say $E^R(k) = 3k-7-j$, then $k_1$ get increased by $j$ while the set $\{2,1,1\}$ remains unchanged, i.e., $\{k_2,k_3,k_4\} = \{2,1,1\}$ as before. This means that it is not possible to add 4 or more edges to $G^R(k)$ to acquire $G^R(k+1)$. Hence, $E^R(k+1) \neq 3(k+1)-6$, as asserted. □

REMARK. From Theorem 3.24 we have $E^R(n) \leq 3n-7$ when $n > 3$, and in this case there does exist a tiling for which $E^R(n) = 3n-7$ (cf. Theorem 3.13).

For $n = 3$, there exists a tiling for which $E^R(3) = 3n-6 = 3$ (refer to Figure 3.12). For $n = 1$ and $n = 2$, there is only one possibility, namely $E^R(1) = 0$ and $E^R(2) = 1$.

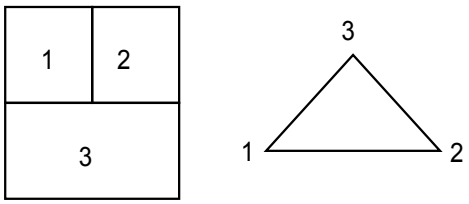


FIGURE 3.12
$T^R(3)$ having 3 edges

COROLLARY 3.25. *A $T_S^R$ is one of the best $T^R$ from the point of view of connectivity.*

*Proof.* From Theorem 3.13, for a $T_S^R(n)$ we have $E_S^R(n) = 3n-7$ when $n > 3$ and from Theorem 3.24 we have $E_S^R(n) \leq 3n-7$ implies that the spiral-based algorithm gives one of the best $T^R(n)$ for $n > 3$ from the point of view of connectivity.

The cases where $n \leq 3$ are obvious (see above Remark). □

COROLLARY 3.26. *The tiling by squares in the golden rectangle is one of the best tilings of a rectangle by squares from the point of view of connectivity.*

*Proof.* In a $T_S^R$, the arrangement of rectangles is same as the arrangement of squares in the golden rectangle. Also if $n > 3$, then the golden rectangle has $3n-7$ edges. Therefore, from Corollary 3.25, the tiling by squares in the

golden rectangle is one of the best tilings by squares from the point of view of connectivity. □

REMARK. In the spiral-based algorithm, sometimes there are extra spaces which help to maximize the connectivity; however in the golden rectangle there is no extra space and it is one of the best tilings from the point of view of connectivity. This may be one of the reasons for using the golden ratio so often.

*Open problem 2: Does there exist a family of* $T^R(n)$ *which is best from the point of view of connectivity i.e. having* $3n - 7$ *edges?*

In this chapter, many covariants associated with a $T^R$ and $T^R_S$ have been provided but the focus was on proving that a $T^R_S$ is one of the best $T^R$ from the point of view of connectivity. In the next chapter the idea behind the spiral-based algorithm will be extended to obtain other shaped tilings.

# CHAPTER 4

# GENERALIZING $T_S^R$ USING AN ALLOCATION MATRIX

In this chapter, we introduce *allocation matrices*, a mathematical object which behaves as a set of constraints. The purpose of an allocation matrix is to control the quality and number of possible tilings, according to the wishes of architects, etc. Allocation matrices are also used to obtain some tilings other than $T^R$. As an illustration, a plus-shape tiling has been developed by putting together five different $T_S^R$ (because $T_S^R$ is one of the best $T^R$ from the point of view of connectivity, cf. Corollary 3.25).

## 4.1. First Definitions

### 4.1.1 Allocation Matrix

The relative weights of the desired proximities among the tiles are determined by considering the relationships between the activities they house. For example the *number of trips* between each pair of tiles is a relation between the activities they house. On this basis, in a hospital, the room of a nurse is closer to the room of a patient in comparison to the room of a surgeon. The trips can be further weighed on the basis of the size of the tiles. The obtained results are then recorded in the form of a matrix, which lists the weight of the connection between each pair of tiles. This matrix is called an *allocation matrix* (Kalay [23], Chapter 13 and Figure 13.7).

An allocation matrix of order $n \times n$ is represented by $A_T(n)$. In $A_T$, the numbers vary from 0 to 10 and the matrix is always symmetric. For example, one of the $A_T(16)$ is given in Figure 4.2. In a sense, the numbers in $A_T$ represent the probability of two tiles being adjacent. For example, the number 10 corresponds to the maximum probability for the tiles to be adjacent whereas the number 0 stands for the lowest probability for the tiles to be adjacent.

Occasionally it may be necessary to overlook a higher number and consider a lower number in order to obtain a tiling. Therefore, $A_T$ is not a constraint

which needs to be entirely satisfied but it represents a set of constraints. This nature of $A_T$ leads to an open problem as follows:

*Open problem 3: How to interpret $A_T$ and which interpretation leads to the best result for the grouping of the tiles?*

In the next section, a precise method to evaluate $A_T$ is specified, which comes handy for obtaining the initial adjacency pairs.

### 4.1.2 PLUS-SHAPE TILING

DEFINITION 4.1. We refer to *plus-shape tiling* for shape tiling in the sense of Definition 1.5 when the given shape is a cross (Figure 4.1(A)). We denote it by $T^P$.

A $T^P$ of order $n$ is denoted by $T^P(n)$. To obtain a $T^P$, the plus shape is first divided into 5 rectangles (we shall denote them by *central, left, upper, right* and *lower* $T^R$) as shown in Figure 4.1(B) and all these $T^R$ can have either the same or different areas. Then all the tiles are grouped into 5 subsets and for every group, a $T^R_S$ is constructed. At last, the 5 obtained $T^R_S$ are adjoined and extra spaces (if required) are drawn to encompass a $T^P$.

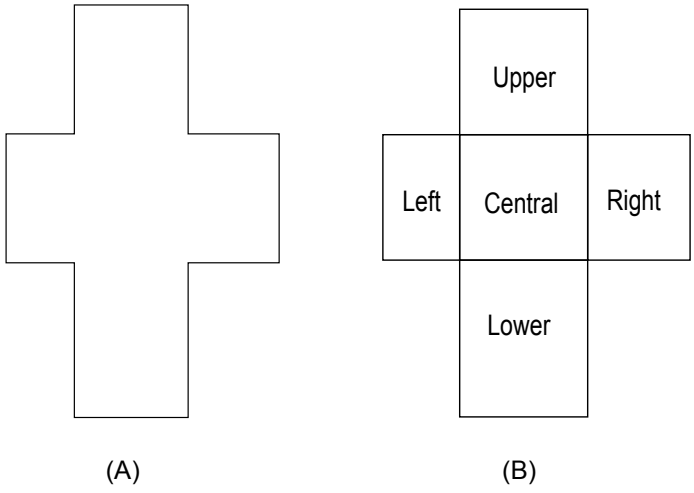


FIGURE 4.1
A plus shape and its parts

## 4.2. INITIAL ADJACENCY PAIRS

For categorizing the tiles into 5 groups, we need to define what we call *initial adjacency pairs*.

DEFINITION 4.2. An *adjacency pair* consists of 2 tiles which are adjacent. The *initial adjacency pairs* are obtained from $A_T$ as follows:

Let $I_p$ be a set of tiles forming the initial adjacency pairs and initially $I_p = \varnothing$. Let $R_i$ be a tile corresponding to the $i^{th}$ row of allocation matrix $A_T(n)$ and $A_T(n)$ is represented by $[a_{ij}]_{n\times n}$. $M = \max\{a_{ij}\}$ gives the highest number of all rows and we have $i = 1, ...n$; $j = 1, ..., n$.

We start by considering all the pairs of tiles corresponding to $M$ as initial adjacency pairs; for example, for $A_T(16)$ in Figure 4.2 where $M = 10$, $(R_5, R_8)$, $(R_6, R_7)$, $(R_{10}, R_{11})$, $(R_{12}, R_{14})$, $(R_{13}, R_{14})$ are the initial adjacency pairs corresponding to the number 10.

| | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 | R10 | R11 | R12 | R13 | R14 | R15 | R16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **R1** | 0 | 8 | 6 | 6 | 8 | 6 | 9 | 6 | 4 | 5 | 3 | 2 | 2 | 2 | 8 | 6 |
| **R2** | 8 | 0 | 6 | 6 | 8 | 6 | 9 | 6 | 4 | 5 | 3 | 2 | 2 | 2 | 8 | 6 |
| **R3** | 6 | 6 | 0 | 8 | 6 | 8 | 6 | 9 | 4 | 4 | 3 | 6 | 6 | 4 | 4 | 6 |
| **R4** | 6 | 6 | 8 | 0 | 6 | 8 | 6 | 9 | 4 | 4 | 3 | 4 | 4 | 4 | 4 | 6 |
| **R5** | 8 | 8 | 6 | 6 | 0 | 6 | 9 | 10 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 2 |
| **R6** | 6 | 6 | 8 | 8 | 6 | 0 | 10 | 9 | 6 | 2 | 2 | 2 | 2 | 2 | 4 | 2 |
| **R7** | 9 | 9 | 6 | 6 | 9 | 10 | 0 | 6 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 2 |
| **R8** | 6 | 6 | 9 | 9 | 10 | 9 | 6 | 0 | 6 | 6 | 4 | 4 | 4 | 4 | 4 | 6 |
| **R9** | 4 | 4 | 4 | 4 | 2 | 6 | 2 | 6 | 0 | 8 | 6 | 6 | 6 | 6 | 6 | 9 |
| **R10** | 5 | 5 | 4 | 4 | 2 | 2 | 2 | 6 | 8 | 0 | 10 | 4 | 4 | 4 | 6 | 4 |
| **R11** | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 4 | 6 | 10 | 0 | 2 | 2 | 2 | 4 | 4 |
| **R12** | 2 | 2 | 6 | 4 | 2 | 2 | 2 | 4 | 6 | 4 | 2 | 0 | 8 | 10 | 2 | 9 |
| **R13** | 2 | 2 | 6 | 4 | 2 | 2 | 2 | 4 | 6 | 4 | 2 | 8 | 0 | 10 | 2 | 9 |
| **R14** | 2 | 2 | 4 | 4 | 2 | 2 | 2 | 4 | 6 | 4 | 2 | 10 | 10 | 0 | 2 | 4 |
| **R15** | 8 | 8 | 4 | 4 | 4 | 4 | 4 | 4 | 6 | 6 | 4 | 2 | 2 | 2 | 0 | 6 |
| **R16** | 6 | 6 | 6 | 6 | 2 | 2 | 2 | 6 | 9 | 4 | 4 | 9 | 9 | 4 | 6 | 0 |

FIGURE 4.2
An $A_T(16)$

Now we reduce $M$ by 1 and consider every row of $A_T$ one by one. If $R_i \in I_p$ we skip the $i^{th}$ row, otherwise we select all the pairs of tiles corresponding to $M$ from $A_T$ and place them in $I_p$. For $A_T(16)$ we skip rows 5, 6, ... and 14 except row 9. All the adjacency pairs corresponding to the

number 9 in the other rows of $A_T$ are $(R_1, R_7)$, $(R_2, R_7)$, $(R_3, R_8)$, $(R_4, R_8)$, $(R_9, R_{16})$.

The process is repeated until $\bigcup_{i=1}^{n} R_i = I_p$ i.e., until all tiles have been exhausted. For $A_T(16)$ only one tile is not covered, namely $R_{15}$. All the adjacency pairs for number 8 in the row corresponding to $R_{15}$ are $(R_{15}, R_1)$, $(R_{15}, R_2)$.

For further details, refer to the *initial adjacency pair algorithm* (Section A.3).

## 4.3. GROUPING OF THE TILES

Here we describe a procedure for extracting the groups from the initial adjacency pairs.

We start by considering $R_1$ as the first member of the first group.

The formation of a group $G_i$ starts by looking for a tile $R_j$ which is not yet covered in any of the created groups. Each adjacency pair has two tiles, this implies that $R_j$ is adjacent to at least one other tile.

After attaining two members of $G_i$, we search for other members of $G_i$. For convenience we denote the tiles of an adjacency pair by $\{R_a, R_b\}$. If $R_a$ is common between members of $G_i$ and any other adjacency pair, then we consider $R_b$ as a member of $G_i$. By following this procedure for all initial adjacency pairs, we obtain the groups for the given tiles.

For further details refer to the first four steps of the algorithm given in Section A.4 and coming example.

EXAMPLE 4.3. For example consider the initial adjacency pairs obtained in Section 4.2 for $A_T(16)$ in Figure 4.2. For these pairs, we obtain groups as follows:

1. Obtaining the members of the first group

From the initial adjacency pairs, $R_1$ is adjacent to $R_7$ and $R_{15}$, $R_{15}$ is adjacent to $R_2$, and $R_7$ is adjacent to $R_2$ and $R_6$. Therefore, the members of the first group are $R_1, R_2, R_6, R_7,$ and $R_{15}$.

2. Obtaining the members of the second group

$R_3$ is not a member of the first group. Let $R_3$ be the first member of the second group. By referring to the initial adjacency pairs $(R_3, R_8)$, $(R_4, R_8)$ and $(R_5, R_8)$, the members of the second group are $R_3, R_4, R_5,$ and $R_8$.

3. Obtaining the members of the third group

$R_9$ is not a member of either the first or the second group, thus consider $R_9$ as first member of the third group. From the initial adjacency pairs, the members of the third group are $R_9$ and $R_{16}$.

3.4 Obtaining the members of the fourth group

Here, $R_{10}$ is not among the members of the first, second and third groups. By referring to the initial adjacency pairs, the members of the fourth group are $R_{10}$ and $R_{11}$.

3.5 Obtaining the members of the fifth group

Now, $R_{12}$ is not a member of any obtained groups. From the initial adjacency pairs, the members of the fifth group are $R_{12}, R_{13}$ and $R_{14}$.

LEMMA 4.4. *The number of groups can be increased by one; by splitting the largest group (say $G$) into 2 groups (say $G_1$ and $G_2$). The members of $G_1$ and $G_2$ are obtained by redistributing the members of $G$ between $G_1$ and $G_2$ on the basis of comparison of proximity (as given in $A_T$) of each member of $G$ with every other member.*

*Proof.* To increase the number of groups, the largest group $G$ (which has the maximum number of members) is obtained. Then using $A_T$, the pair $(R_i, R_j)$ having the minimum weight in $G$ is selected (the one with minimum weight is selected because lesser is the weight, less important it becomes to put $R_i, R_j$ together in the same group). Consider $R_i$ and $R_j$ to be the first members of $G_1$ and $G_2$ respectively and $R_k$ be another member of $G$. Now if the weight of the pair $(R_i, R_k)$ (as given in $A_T$) is greater than the weight of the pair $(R_j, R_k)$ then $R_k$ will become a member of $G_1$ otherwise consider it as a member of $G_2$.

At the end of this process, the number of groups will be increased by one. For further details refer to step 6 of the algorithm given in Section A.4. □

LEMMA 4.5. *The number of groups can be reduced by one by coalescing the two smallest groups.*

*Proof.* The proof follows by obtaining the two groups having the least number of members and then forming a new group with all the members of these 2 groups. For further details, refer to step 7 of the algorithm given in Section A.4. □

## 4.4. A SPIRAL-BASED PLUS-SHAPE TILING

From the given $A_T$, we attain the initial adjacency pairs using the algorithm given in Section 4.2. Subsequently, we obtain the groups of tiles using the algorithm in Section 4.3. And if the acquired number of groups is less than (or greater than) five then we can apply Lemma 4.4 (or Lemma 4.5 respectively). After this, we construct a $T_S^R$ for every group to obtain a $T^P$.

LEMMA 4.6. *A $T^P$ is generated from five $T^R$, by positioning a $T^R$ in the centre and then positioning the remaining $T^R$ around the one in the centre. After allocating each $T^R$ (other than the central one), an extra space is drawn, if required.*

*Proof.* The steps of the spiral-based plus-shape algorithm given below discusses all the possibilities for constructing a $T^P$ which concludes the proof. □

### 4.4.1 A SPIRAL-BASED PLUS-SHAPE ALGORITHM

In this section, a $T^P$ is developed from 5 groups using a *spiral-based plus-shape algorithm*. A $T^P$ obtained from this algorithm is called *spiral-based* $T^P$ and we shall denote it by $T_S^P$. The other shape tilings can be generated by making a slight change in the same algorithm.

Step 1: *Obtaining initial adjacency pairs from $A_T$ for the given tiles*
Step 2: *Grouping the tiles using initial adjacency pairs*
Step 3: *Fixing the position of the groups*

In a $T^P$, there are 5 positions as shown earlier in Figure 4.1(B). The allocation of any group at any position is done in such a way that no two groups can have the same position.

Step 4: *Selecting a spiral for every group*

The construction of each group is accomplished with the help of one of the eight spirals as specified in Section 2.1.4.

Step 5: *Constructing each $T_S^R$*

After fixing position and spiral for each group, a $T_S^R$ is generated. To fix the notation, we define the width and height of the central, left, upper, right and lower $T_S^R$ as $(L^1, H^1)$, $(L^2, H^2)$, $(L^3, H^3)$, $(L^4, H^4)$ and $(L^5, H^5)$ respectively.

Step 6: *Positioning the central* $T_S^R$

The central $T_S^R$ is positioned at say $(x, y)$. Assigning a $T^R$ at position $(x, y)$ means $(x, y)$ is the upper left corner of this $T^R$.

Step 7: *Positioning the left* $T_S^R$ *and drawing an extra space*

7.1 The left $T_S^R$ is positioned at $(x - L^2, y)$ in such a way that its upper right corner coincides with the upper left corner of the central $T_S^R$.

7.2 Drawing an extra space

For achieving a $T^P$, in the upcoming steps, either the width or the height of the various $T_S^R$ is compared and an extra space is drawn, if required.

To draw an extra space, the following two possible cases are listed:

*i*. If $H^2 > H^1$, we draw an extra space below the central $T_S^R$ at position $(x, y + H^1)$ having width $L^1$ and height $H^2 - H^1$.

*ii*. If $H^2 < H^1$, we draw an extra space below the left $T_S^R$ at position $(x - L^2, y + H^2)$ having width $L^2$ and height $H^1 - H^2$.

Step 8: *Positioning the upper* $T_S^R$ *and drawing an extra space*

The upper $T_S^R$ is positioned at $(x, y - H^3)$. If $L^1 > L^3$ we draw an extra space at position $(x + L^3, y - H^3)$ having width $L^1 - L^3$ and height $H^3$ . If $L^3 > L^1$ we draw an extra space at position $(x + L^1, y)$ having width $L^3 - L^1$ and height $\max(H^1, H^2)$.

To fix the notation, let $G_1, G_2, G_3, G_4, G_5$ be the central, left, upper, right and lower $T_S^R$ respectively. For better understanding of upcoming step, refer to Figure 4.3 where coloured rectangles are extra spaces.

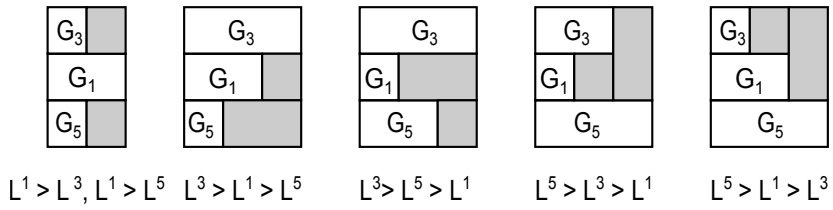


FIGURE 4.3
Comparing $L^1, L^3, L^5$

Step 9: *Positioning the right* $T_S^R$ *and drawing an extra space*

The right $T_S^R$ is positioned at $(x + \max(L^1, L^3, L^5), y)$ (see Figure 4.3). If $H^4 > \max(H^1, H^2)$ we draw an extra space at position $(x - L^2, y + \max(H^1, H^2))$

having width $L^2 + \max(L^1, L^3, L^5)$ and height $H^4 - \max(H^1, H^2)$. If $H^4 < \max(H^1, H^2)$ we draw an extra space at position $(x + \max(L^1, L^3, L^5), y + H^4)$ with width $L^4$ and height $\max(H^1, H^2) - H^4$.

Step 10: *Positioning the lower $T_S^R$ and drawing an extra space*

The lower $T_S^R$ is positioned at $(x, y + \max(H^1, H^2, H^4))$. If $L^5 < \max(L^1, L^3)$ we draw an extra space at position $(x + L^5, y + \max(H^1, H^2, H^4))$ with width $\max(L^1, L^3) - L^5$ and height $H^5$. If $L^5 > \max(L^1, L^3)$ we draw an extra space at position $(x + L^1, y - H^3)$ with width $L^5 - \max(L^1, L^3)$ and height $\max(H^1, H^2) + H^3$.

### 4.4.2 THE AREA OF A SPIRAL-BASED PLUS-SHAPE TILING

Consider a $T_S^P$ is made up of two rectangles say $R_A$ and $R_B$. $R_A$ is made up of the central, lower and upper $T^R$ and $R_B$ is made up of the central, left and right $T^R$, refer to Figure 4.1(B). If the areas of $R_A$ and $R_B$ are added then the area of the central $T^R$ is counted twice. Therefore the total area of a $T_S^P$ is obtained by adding the areas of $R_A$ and $R_B$ and subtracting the area of the central $T^R$.

$L_A = \text{width of } R_A = \max(L^1, L^3, L^5)$

$H_A = \text{height of } R_A = H^3 + H^5 + \max(H^1, H^2, H^4)$

$L_B = \text{width of } R_B = L^2 + L^4 + \max(L^1, L^3, L^5)$

$H_B = \text{height of } R_B = \max(H^1, H^2, H^4)$

*Total area of* $T_S^P = L_A \times H_A + L_B \times H_B - \max(L^1, L^3, L^5) \times \max(H^1, H^2, H^4)$.

REMARK. The area of $T_S^P$ is a *covariant* with respect to changes of the spirals and the group positions.

## 4.5. AN EXAMPLE FOR A SPIRAL-BASED PLUS-SHAPE TILING

This example is of great benefit for understanding the concept of tilings in the preceding and upcoming sections and chapters.

EXAMPLE 4.7. Step 1: *Obtaining the initial adjacency pairs from* $A_T(16)$

We have given 16 tiles $R_i$ $(i = 1, \ldots, 16)$ having areas 48, 48, 48, 48, 24, 24, 6, 6, 144, 72, 48, 48, 48, 72, 72, 12 respectively; the ratio between width and height of each tile (1.618).

To proceed further, we consider the same $A_T(16)$ given in Figure 4.2. The initial adjacency pairs corresponding to $A_T(16)$ were obtained in Section 4.2.

Step 2: *Extracting five groups from the initial adjacency pairs*

For the five groups, refer to Example 4.3.

Step 3: Suppose the position for each group is:

1. *Central group:* $R_1, R_2, R_7, R_6, R_{15}$
2. *Left group:* $R_3, R_4, R_8, R_5$
3. *Upper group:* $R_9, R_{16}$
4. *Right group:* $R_{10}, R_{11}$
5. *Lower group:* $R_{12}, R_{13}, R_{14}$

Step 4: Consider *spiral*8, *spiral*6, *spiral*4, *spiral*2 and *spiral*1 are the spirals for the central, left, upper, right and lower groups respectively.

Step 5: The five computer generated $T_S^R$ are shown in Figure 4.4.

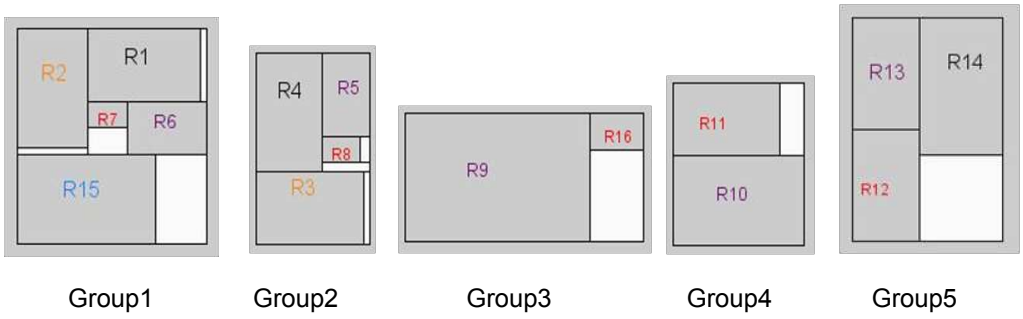


FIGURE 4.4
Five $T_S^R$ for obtaining $T_S^P$

Step 6: The central $T_S^R$ is positioned at $(x, y)$.

Step 7: Allocate the left $T_S^R$. Since $H^2 < H^1$, we draw an outer extra space below the left $T_S^R$ (just to categorize the extra spaces (see Section 5.1.1), an outer extra space is mentioned).

Step 8: Allocate the upper $T_S^R$ and since $L^3 > L^1$ we draw an outer extra space to the right of the central $T_S^R$.
Step 9: Position the right $T_S^R$ and since $H^4 < \max(H^1, H^2)$, we draw an outer extra space below the right $T_S^R$.
Step 10: Allocate the lower $T_S^R$ and here also we draw an outer extra space to the right of the lower $T_S^R$ because $L^5 < \max(L^1, L^3)$.

For further details, refer to Figure 4.5 where a computer generated $T_S^P$ is shown.

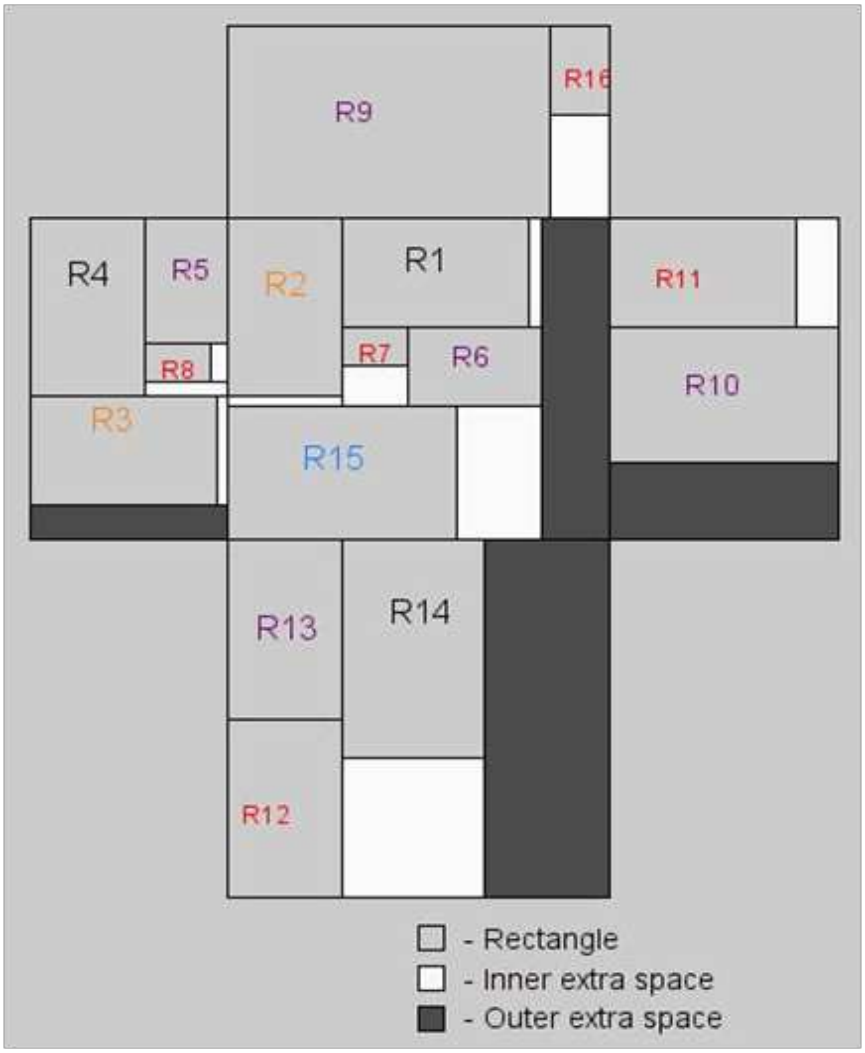


FIGURE 4.5
Obtained $T_S^P$

Step 11 : *Obtaining the area of obtained* $T_S^P$

In Figure 4.5, the widths and heights of the central, left, upper, right and lower $T_S^R$ are $(14.8, 15.9), (9.3, 14.3), (17.9, 9.5), (10.7, 12.1), (12.1, 17.6)$ respectively.

Using Section 4.4.2, the area of $R_A = \max(14.8, 18, 12.1) \times \{9.5 + 17.6 + \max(16, 14.3, 12.1)\} = 18 \times 43 = 775.8$.

The area of $R_B = \{9.3 + 10.7 + \max(14.8, 18, 17.6)\} \times \max(16, 14.3, 12.1) = 37.9 \times 16 = 606.4$.

The area of central $T^R = \max(14.8, 18, 12.1) \times \max(16, 14.3, 12.1) = 18 \times 16 = 288$.

The area of $T_S^P = 775.8 + 606.4 - 288 = 1094.2$.

## 4.6. THE TOTAL NUMBER OF SOLUTIONS PROVIDED

In this section, we compute the total number of $T_S^P$ that can be generated by using the spiral-based plus-shape algorithm.

*1. The number of solutions obtained by changing spiral for each group*

As discussed earlier, five groups are used to generate a $T_S^P$ but each group is drawn using one of the eight spirals. Therefore for a given set of data the number of different $T_S^P$ that can be obtained by changing the spiral for each group is $8^5 = 32,768 \approx 2^5 \times 10^3$.

*2. The number of solutions obtained by changing the position of each group*

In a $T_S^P$, a group can be assigned any of the five positions shown in Figure 4.1(B) such that no two groups have the same position. Therefore for a given set of data by changing position for each group, $5! = 120$ different $T_S^P$ can be obtained.

So for the given set of data, in total there are $8^5 \times 120 = 3,932,160 \approx$ 4 million different $T_S^P$ can be obtained.

## 4.7. OTHER SHAPE TILINGS

In this section we present two other shape tilings developed in analogy with the spiral-based plus-shape algorithm. This is done to demonstrate that numerous shape tilings can be developed using simple modifications of the same algorithm.

In Section 4.4, a $T_S^P$ was constructed from five groups. Just by reducing the number of groups by either 1 or 2, some other interesting shape tilings can easily be obtained; for example, we will obtain tilings for *vertically flipped* **T** (see Figure 4.6(A)) and *horizontally flipped* **L** (see Figure 4.6(B)) shapes. These 2 shape tilings are achieved by considering the number of groups as 4 and 3 respectively.

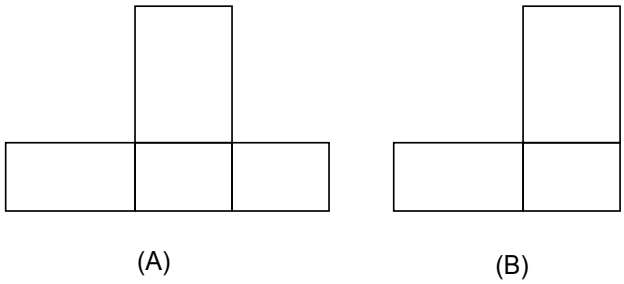


FIGURE 4.6
The vertically flipped T and horizontally flipped L shapes

EXAMPLE 4.8. Consider the same list of tiles and $A_T(16)$ as given in Example 4.7.

Using Section 4.2, we obtain the initial adjacency pairs from the given $A_T$ and then using Section 4.3, we extract four groups.

Suppose position and spiral for the obtained 4 groups are as follows:

1. *Central group:* $R_3, R_4, R_8, R_5$ *with spiral* 7.
2. *Left group:* $R_1, R_2, R_7, R_6, R_{15}$ *with spiral* 5.
3. *Upper group:* $R_{12}, R_{13}, R_{14}$ *with spiral* 4.
4. *Right group:* $R_9, R_{16}, R_{10}, R_{11}$ *with spiral* 8.

For these four groups, a computer generated spiral-based vertically flipped T-shape tiling is shown in Figure 4.7.

To obtain a horizontally flipped L-shape tiling, only 3 groups are required. Their position and spiral are as follows:

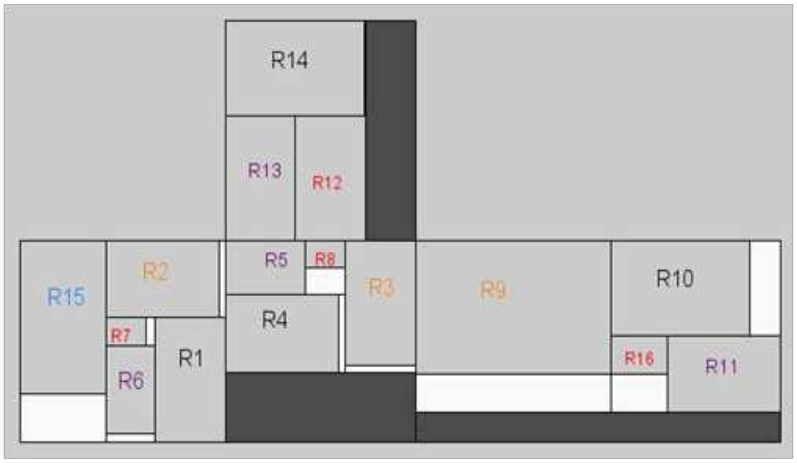


FIGURE 4.7
A spiral-based vertically flipped T-shape tiling

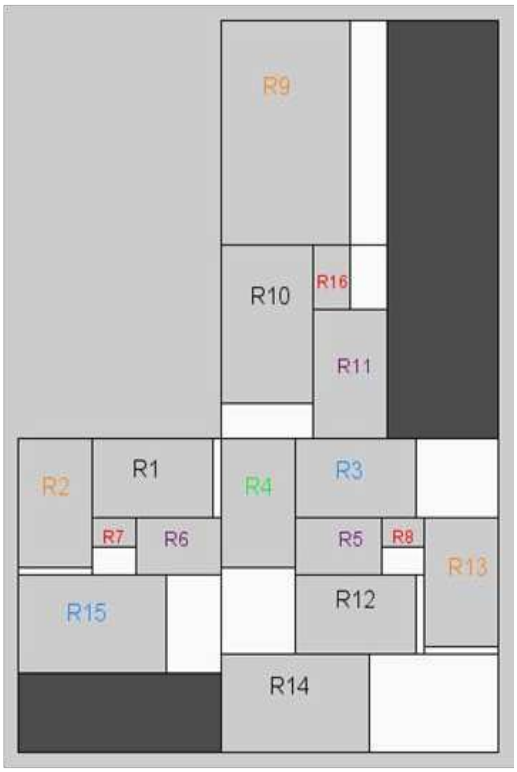


FIGURE 4.8
A spiral-based horizontally flipped L-shape tiling

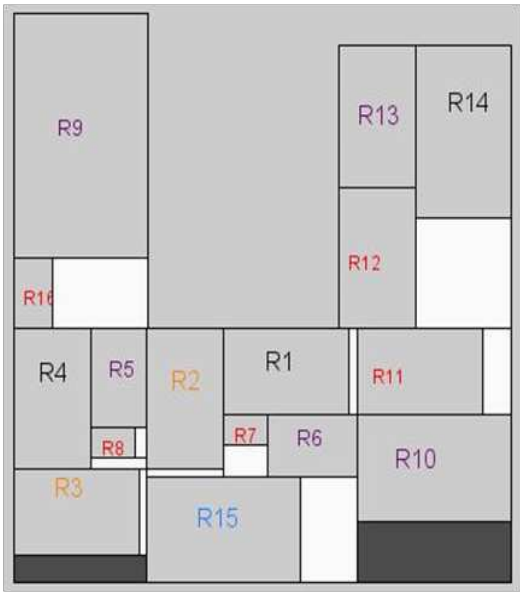


FIGURE 4.9
A spiral-based U-shape tiling

1. *Central group:* $R_3, R_4, R_8, R_5, R_{12}, R_{13}, R_{14}$ *with spiral*7.
2. *Left group:* $R_1, R_2, R_7, R_6, R_{15}$ *with spiral*8.
3. *Upper group:* $R_9, R_{16}, R_{10}, R_{11}$ *with spiral*2.

From these three groups, a computer generated spiral-based horizontally flipped L-shape tiling is shown in Figure 4.8.

For another example, refer to Figure 0(B). For this figure, we consider the same list of tiles, $A_T(16)$ and groups as given in Example 4.7. Suppose position and spiral for each group are:

1. *Central group:* $R_1, R_2, R_7, R_6, R_{15}$ *with spiral*8.
2. *Left group:* $R_3, R_4, R_8, R_5$ *with spiral*6.
3. *Upper left group:* $R_9, R_{16}$ *with spiral*6.
4. *Right group:* $R_{10}, R_{11}$ *with spiral*5.
5. *Upper right group:* $R_{12}, R_{13}, R_{14}$ *with spiral*1.

For these groups, we have constructed a computer generated spiral-based U-shape tiling which is shown in Figure 4.9.

This chapter describes the construction of a $T_S^P$ and how by using this concept, many different shape tilings can be generated. For example, three different shape tilings are obtained in Section 4.7. Interestingly, about 4 million

$T_S^P$ can be produced for a given set of data. In general, for a particular shape, one can have many choices but it is quite impossible to go for each solution one by one. Therefore some covariants and invariants need to be defined and derived which would help to reduce the number of solutions. In the next chapter, adjacency among the tiles of $T_S^P$ will be defined, which will be used to obtain some new covariants and invariants.

# CHAPTER 5

# ADJACENCY GRAPH FOR SHAPE TILINGS

In this chapter, we focus on generating the adjacency graph of a $T_S^P$ and we compute degree of connectivity of the adjacency graphs (for degree of connectivity refer to Definition 1.10).

## 5.1. ADJACENCY VIA EXTRA SPACES

The concept of adjacency among the components of a group has already been discussed in Chapter 3. Now we proceed by defining adjacency among the members of two different groups. Before expanding the concept of adjacency; we go through the different types of extra spaces.

### 5.1.1 INNER AND OUTER EXTRA SPACES

DEFINITION 5.1. An extra space positioned inside a group, i.e. belonging to a group is called an *inner extra space*. Each inner extra space is a virtual part of some group members(tiles) and is generated due to the difference in width (or height) of the components of a group.

An extra space which is situated outside a group is called an *outer extra space*. Such an extra space is a virtual part of a group (tiling) and is generated due to the difference in width (or height) of the groups.

EXAMPLE 5.2. The inner and outer extra spaces are mentioned in Figure 4.5.

### 5.1.2 ADJACENCY AMONG TILES OF DIFFERENT GROUPS

To further define the adjacency among the tiles of different groups, we consider the following cases:

*1. When there is no extra space between two tiles belonging to two different groups*

If both tiles share a wall or a sub-wall, then they are directly adjacent otherwise not (cf. Definitions 1.8 and 1.9, where direct adjacency among tiles and related terms are defined).

EXAMPLE 5.3. In Figure 4.5, R14 is directly adjacent to R15 because they share a sub-wall. Also, R1 is directly adjacent to R9 because it shares a wall with R9.

*2. When there are either one or two inner extra spaces between two tiles belonging to two different groups*

If the two tiles are separated by one inner extra space, all the possible cases have already been discussed in Section 3.1. For more than one inner extra space, adjacency is discussed as follows:

Suppose the inner extra spaces $I_1$ and $I_2$ are virtual parts of the tiles $R_1$ and $R_2$ respectively. If $I_1$ is adjacent to $I_2$ then $R_1$ is adjacent to $R_2$.

EXAMPLE 5.4. Refer to Figure 4.5 where R3 is adjacent to R15 because the inner extra space between them is a virtual part of R3 and this inner extra space is adjacent to R15. Also, R2 is adjacent to R3 because the inner extra spaces which are virtual parts of R2 and R3 are adjacent to one another.

*3. When there are either one or two outer extra spaces between two tiles belonging to two different groups*

If by removing the outer extra spaces two tiles are adjacent, then we consider these tiles to be adjacent.

EXAMPLE 5.5. In Figure 4.5, with an outer extra space between R6 and R10, R6 is adjacent to R10.

*4. When there are at most two inner and at most two outer extra spaces between two tiles belonging to two different groups*

The adjacency in this case is defined by combining the definitions of adjacency in the two cases given above such that we first apply case 3 and then case 2.

EXAMPLE 5.6. In Figure 4.5, with both an outer extra space and an inner extra space between R1 and R11, R1 is adjacent to R11, where the inner extra space is a virtual part of R1.

We denote the adjacency graph of $T^P$ and of $T_S^P$ by $G^P$ and $G_S^P$ respectively.

EXAMPLE 5.7. For the example in Figure 4.5, we get the following adjacency graph.

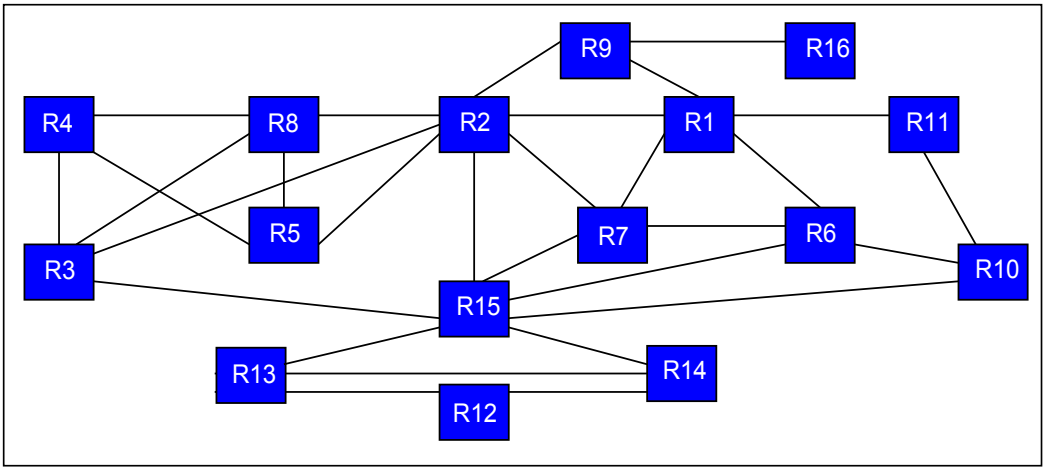


FIGURE 5.1
$G_S^P$ for the $T_S^P$ in Figure 4.5

## 5.2. THE NUMBER OF EDGES

Once we know the adjacency graph we can easily count its degree of connectivity. In this section, all the possibilities for the number of edges in a $G_S^P$ is discussed which will be used further to reduce the number of solutions (cf. Section 6.4).

The core result of this section is given by Corollary 5.12 which states that a $G_S^P(m)$ can have at most $3m-19$ edges. To prove this result, some lemmas are required, which are also interesting for their own sake and provide some new covariants.

REMARK. In a $T_S^P$, the members of left, upper, right and lower $T_S^R$ can only be adjacent to members of central $T_S^R$ and vice-versa.

LEMMA 5.8. *In a $T_S^P$, at most 3 members of a $T_S^R$ (other than the central $T_S^R$) can be adjacent each to at most 3 members of the central $T_S^R$.*

*Proof.* A $T_S^P$ is made up of five $T_S^R$ and the components of the left, upper, right or lower groups can be adjacent only to the components of the central group. From Lemma 3.19, for a $T_S^R$ we have $\{k_1, k_2, k_3, k_4\} = \{3, 2, 2, 1\}$, i.e. the maximum value of an adjacent number is 3. This implies that at most 3 members of a $T_S^R$ can be adjacent to at most 3 members of any other $T_S^R$. □

Consider that the virtual parts of tiles are made real. We denote by $a_1, a_2, a_3$ the width (resp. the height) of adjacent sides of the components of the upper and lower $T_S^R$ (resp. the left and right $T_S^R$). We shall also denote by $b_1, b_2, b_3$ the width (resp. the height) of adjacent sides of the components of the central $T_S^R$, as depicted in Figure 5.2. The sizes of $a_1$, $a_2$, $a_3$, $b_1$, $b_2$ and $b_3$ are usually different.

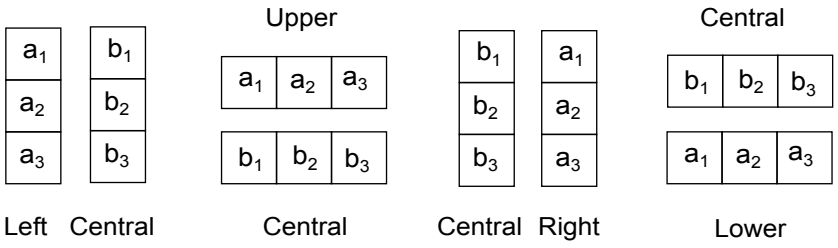


FIGURE 5.2
The width or height of adjacent sides of the components of $T_S^R$

Let $E_S^2$ be the number of edges connecting the members of the central group to those of another group of a $T_S^P$.

**Note:** *For the upcoming lemma, $a_i$ adjacent to $b_j$ means that the corresponding group members are adjacent; for computation the inner extra spaces are made real.*

LEMMA 5.9. $E_S^2 \leq (2+j)$ *when at most* $0 < j < 4$ *members of the central $T_S^R$ are adjacent to at most three members of a $T_S^R$.*

*Proof.* When at most 3 members of the central $T_S^R$ are adjacent to at most 3 members of a $T_S^R$, we have two cases:

If $b_1$ is adjacent to $a_1, a_2, a_3$ then $b_2, b_3$ can only be adjacent to $a_3$ (cf. Figure 5.3(A)).

If $b_1$ is adjacent to $a_1, a_2$ then $b_2$ can be adjacent to $a_2, a_3$ and $b_3$ can be adjacent to $a_3$ (cf. Figure 5.3(B)).

Hence, we have $E_S^2 \leq (i + j - 1) = 5$.

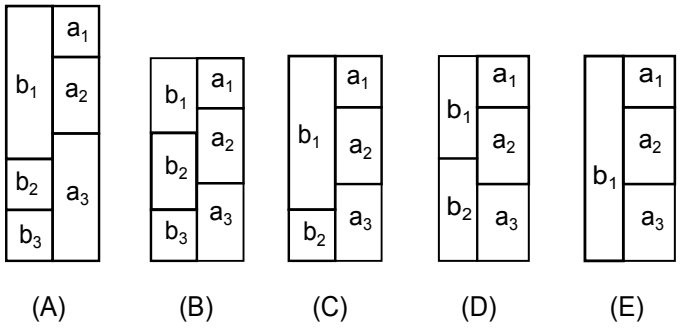


FIGURE 5.3
Computing $E_S^2$

When at most 2 members of the central $T_S^R$ are adjacent to at most 3 members of a $T_S^R$ we have following two cases:

If $b_1$ is adjacent to $a_1, a_2, a_3$ then $b_2$ can only be adjacent to $a_3$ (cf. Figure 5.3(C)).

If $b_1$ is adjacent to $a_1, a_2$ then $b_2$ can only be adjacent to $a_2, a_3$ (cf. Figure 5.3(D)).

Hence, we have $E_S^2 \leq (i + j - 1) = 4$.

By same argument, when one member of the central $T_S^R$ is adjacent to at most 3 members of a $T_S^R$ we have $E_S^2 \leq (i+j-1) = 3$ (cf. Figure 5.3(E)). □

We denote the number of edges in a $G_S^P(m)$ by $E_S^P(m)$. Let $p_1(m)$ be the number of edges in a $T_S^P(m)$ if adjacency is restricted to the members of each group and $p_2(m)$ is for the edges which come uniquely from adjacency among the members of different groups of a $T_S^P(m)$.

LEMMA 5.10. $p_1(m) = 3m - 35$ *provided that each group has at least four members.*

*Proof.* A $T_S^P(m)$ is obtained by combining five $T_S^R$. From Theorem 3.13, for each $n_i > 3$ we have $E_S^R(n_i) = 3n_i - 7$, where $i = 1, \ldots, 5$. Therefore, if adjacency is restricted to the members of each group, $p_1(m) = \sum_{i=1}^{5}(3n_i - 7) = 3m - 35$. □

LEMMA 5.11. $p_2(m) \leq 16$.

*Proof.* From Lemma 5.8, at most 3 members of a $T_S^R$ (other than the central $T_S^R$) can be adjacent to at most 3 members of the central $T_S^R$.

From Lemma 3.19, for a $T_S^R$ we have $\{k_1, k_2, k_3, k_4\} = \{3, 2, 2, 1\}$ implies 3 members of a $T_S^R$ can be adjacent to 3 (or 2 or 2 or 1) members of central $T_S^R$ respectively. From Lemma 5.9 we have $E_S^2 \leq 5$( or 4 or 4 or 3) respectively. Hence we have $p_2(m) \leq 5 + 8 + 3 = 16$. □

COROLLARY 5.12. $E_S^P(m) = p_1(m) + p_2(m) \leq 3m - 19$ *if each* $T_S^R$ *has at least 4 members.*

*Proof.* The proof follows from Lemmas 5.10 and 5.11. □

REMARK. With respect to changes in the position of groups and change of the spiral for each group, $E_S^P(m)$ is a *covariant.*

In context of Corollary 5.12, some more results are given below.

LEMMA 5.13. *In a* $T_S^P(m)$,

*1. If a* $T_S^R$ *(other than central* $T_S^R$*) has* 2 *or* 3 *members, then* $p_2(m)$ *in Corollary* 5.12 *gets reduced by* 1

*2. If a* $T_S^R$ *(other than central* $T_S^R$*) has only one member, then* $p_2(m)$ *gets reduced by* 2

*3. If central* $T_S^R$ *has* 3 *or* 2 *or* 1 *members, then* $p_2(m)$ *gets reduced by* 1 *or* 2 *or* 4 *respectively*

*4. Let* $p_1(n) = (3n - 35)$. *In* $p_1(n)$ *if any* $n_i = 3$ *or* $n_i = 2$ *or* $n_i = 1$, *then the value* $(3n_i - 7)$ *in* $p_1(n)$ *becomes* $(3n_i - 6)$ *or* $(3n_i - 5)$ *or* $(3n_i - 3)$ *respectively where* $n_i$ *is the number of members of* $i^{th}$ *group.*

*Proof.* 1. If a $T_S^R$ has 2 or 3 members, then the set $\{3, 2, 2, 1\}$ for adjacent numbers will become $\{2, 2, 1, 1\}$. Hence, at most 2 members of a $T_S^R$ can be adjacent to at most 3 members of the central $T_S^R$. Therefore, when adjacency is considered among the members of different groups, $p_2(m) = 16$ will get reduced by one.

2. If a $T_S^R$ has only 1 member, then the set $\{3,2,2,1\}$ for adjacent numbers will become $\{1,1,1,1\}$. Hence, only one member of a $T_S^R$ can be adjacent to at most 3 members of the central $T_S^R$. Therefore, $p_2(m) = 16$ will get reduced by 2.

3. This result is proved using the same reasoning as given in the proof for the first two points.

4. The proof follows from the fact that for $n_i = 3$, $n_i = 2$ and $n_i = 1$, $E_S^R(3) = 3(3n_i - 6)$, $E_S^R(2) = 1(3n_i - 5)$ and $E_S^R(1) = 0(3n_i - 3)$ respectively. □

REMARK. Interestingly, changing the spiral for a group, leaves the number of edges *invariant* for $G_S^R$ (cf. Theorem 3.13) but it is a *covariant* for $G_S^P$ (cf. Corollary 5.12).

### 5.3. RACING CARS

The result given by Lemma 5.9 is generalized using the concepts of game theory. For better understanding refer to the game explained below.

Consider that a stretch of road is subdivided into $N$ parts, $b_1, \ldots, b_N$, separated by $N-1$ traffic lights and there are $m$ potential players $(a_1, \ldots, a_m)$, who wish to drive a Ferrari over some distance each, one after another.

Each player has to pay 1 dollar for the privilege of getting into the car. In addition, a driver has to pay a one dollar fine for each traffic light he/she passes without stopping (but no fine if he/she stops at the traffic light and surrenders the car to the next player).

Now we calculate the amount of money (say $E_S^2$) the organisers will cash at the end of the game. This problem is seen as "what is the maximum value of $E_S^2$ when at most $N$ members of the central $T_S^R$ are adjacent to at most $m$ members of a $T_S^R$".

As discussed above, there will be at most 1 dollar for each potential player, and at most $N-1$ dollars in fines; hence the organisers will cash at most $m+N-1$ dollars. For better understanding, suppose $N \leq 3$ and $m \leq 3$. Then we have $E_S^2 \leq 5$ (cf. Lemma 5.9).

In Chapter 3, the maximum number of edges for a $G^R$ and a $G_S^R$ was obtained. In this chapter, the maximum number of edges for a $G_S^P$ has been calculated.

But, at this stage it is not known that $T_S^P$ is best from the point of view of connectivity or not, which leads to an open problem as follows:

*Open problem 4: What is the maximum number of edges for a* $G^P$ *?*

In the next chapter, we derive new covariants and discuss some methods for reducing the number of solutions.

# CHAPTER 6

# INVARIANTS AND COVARIANTS

This chapter focuses on obtaining the best solution and reducing the number of solutions by means of *invariants* and *covariants* (cf. Section 2.1.3). For $T_S^R$ and $T_S^P$ a lot of associated invariants and covariants have already been derived in the previous chapters.

## 6.1. Invariants and Covariants associated with a $G_S^P$

In this section we study some graph invariants and covariants which are related to $G_S^P$.

For the definitions related to graphs in the upcoming subsections, refer to Jonathan and Yellen [16].

### 6.1.1 Adjacency matrix and Degree of vertices

The power of linear algebra is applied to graph theory through representation of graphs by matrices. An adjacency matrix is the first graph invariant to be elaborated. Adjacency among the different vertices of $G_S^P$ has already been discussed. The adjacency matrix is obtained straightforwardly from the adjacency pairs.

Definition 6.1. The *adjacency matrix of a simple graph G*, denoted by $A_G$, is the symmetric matrix whose rows and columns are indexed by $V_G$ (in the same order) and is such that

$$A_G[u,v] = \begin{cases} 1, & \text{if } u \text{ and } v \text{ are adjacent} \\ 0, & \text{otherwise.} \end{cases}$$

For a simple graph with no self-loops, the adjacency matrix must have $0$'s on the diagonal.

DEFINITION 6.2. The *degree (valency)* of a vertex $v$ in a graph $G$, denoted by $\deg(v)$, is the number of vertices adjacent to $v$.

We obtain the degree of each vertex by adding the number of 1's in the corresponding row (or column) of the adjacency matrix.

EXAMPLE 6.3. For the $G_S^P$ in Figure 5.1, the degrees of all the vertices, which correspond to all the tiles $R_i$ $(i = 1, \ldots, 16)$, are (5, 7, 4, 3, 3, 4, 4, 4, 3, 3, 2, 2, 3, 3, 7, 1).

To study the distribution of degrees, one uses methods similar to those commonly used in probability and statistics, namely mean, standard deviation, measures of dispersion, minimum and maximum.

For real numbers $\{x_1, x_2, \ldots, x_n\}$, we denote the *arithmetic mean* by $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

The *standard deviation* $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}$ gives the variation (or dispersion) that exists from the corresponding mean. A low value of the standard deviation shows that the data points tend to be very close to the mean, whereas a high value of the standard deviation can sometimes indicate that the data points are spread out over a large range of values.

DEFINITION 6.4. The *measure of statistical dispersion* is a non-negative real number that is zero if all the data have the same value and increases if the data become more diverse. In percentage terms, it is given as $\rho = (\sigma/\overline{x}) \times 100$.

In this subsection, we calculate the measure of dispersion of the degree around its mean.

EXAMPLE 6.5. For all the degrees obtained in Example 6.3, $\overline{x} = 3.62$, $\sigma = 1.58$, $\rho = 43.48\%$, maximum=7, minimum = 1.

If the degrees of all the graph vertices are given, then the corresponding graph is obtained using the algorithm given in Section A.1.

### 6.1.2 DISTANCE AND SHORTEST PATH

The classic example of a real life graph is a network of roads connecting cities. One of the important algorithmic problems on such road networks is finding the shortest path between any two given vertices.

DEFINITION 6.6. For a graph $G$, the *distance* $d(x, y)$ between two vertices $x$ and $y$ is the length of the shortest path from $x$ to $y$, considering all possible paths in $G$ from $x$ to $y$.

The distance between any node and itself is 0 and if there is no path from $x$ to $y$, then $d(x, y)$ is infinity.

It is to be noted that for the distance and shortest path, we consider only non-weighted graphs.

DEFINITION 6.7. The *distance matrix* is a matrix (two-dimensional array) containing the distances, taken pairwise, of the set of vertices. This matrix has order $n \times n$, where $n$ is the number of vertices.

Once we know the adjacency matrix, we can obtain a shortest path (and its length) between any two vertices of $G_S^P$ and the distance matrix by using *Floyd's Algorithm* (see Section A.2 and Pemmaraju and Skiena [34], Chapter 8).

EXAMPLE 6.8. For the $G_S^P$ in Figure 5.1, we have the following results:

The distance between $R_1$ and $R_4$ is 3 and the distance between $R_5$ and $R_{12}$ is 4.

A shortest path between $R_1$ and $R_4$ is given by:

$R_1 \to R_2 \to R_3 \to R_4$.

Also a shortest path between $R_5$ and $R_{12}$ is:

$R_5 \to R_2 \to R_{15} \to R_{13} \to R_{12}$.

### 6.1.3 ECCENTRICITY, RADIUS, DIAMETER AND CENTRE

In this section we discuss several graph invariants related to the all-pairs shortest-path matrix.

DEFINITION 6.9. The *eccentricity* of a graph vertex $v$ in a graph $G$, denoted $\text{ecc}(v)$ is the distance from $v$ to a vertex farthest from $v$. That is,

$$\text{ecc}(v) = \max_{x \in V_G} \{d(v,x)\}$$

DEFINITION 6.10. The maximum eccentricity is the graph *diameter*, denoted by $\text{diam}(G)$. That is,

$$\text{diam}(G) = \max_{x \in V_G} \{\text{ecc}(x)\} = \max_{x,y \in V_G} \{d(x,y)\}$$

DEFINITION 6.11. The minimum graph eccentricity is called the graph *radius*, denoted by $\text{rad}(G)$. That is,

$$\text{rad}(G) = \min_{x \in V_G} \{\text{ecc}(x)\}$$

DEFINITION 6.12. A *central vertex* of a graph $G$ is a vertex with minimum eccentricity. Thus $\text{ecc}(v) = \text{rad}(G)$.

The *centre* of the graph is the set of all vertices with minimum eccentricity.

EXAMPLE 6.13. For the $G_S^P$ in Figure 5.1, the eccentricities of all the vertices corresponding to all the tiles are (4, 3, 3, 4, 4, 3, 3, 4, 4, 4, 4, 5, 4, 4, 3, 5) with radius = 3, diameter = 5 and centre = $\{R_2, R_3, R_6, R_7, R_{15}\}$.

### 6.1.4 CUT VERTEX AND CUT PAIR

DEFINITION 6.14. A *cut vertex* of a connected graph $G$ is a vertex whose removal renders $G$ disconnected. Any graph with no cut vertices is said to be *biconnected*.

Biconnectivity is an important property due to many reasons. For example, *Menger's Theorem* implies that any graph with at least 3 vertices is biconnected if and only if there are at least 2 vertex disjoint paths between any pair of vertices. If looked from the perspective of communication networks, biconnected networks are more fault-tolerant since blowing away any single node does not cut off communication for any other node.

DEFINITION 6.15. A *cut pair* in a connected graph $G$ is a pair of vertices whose removal renders $G$ disconnected.

To obtain cut vertices, we remove each vertex $v_i$ one by one and obtain the distance matrix for the remaining vertices. If any entry in the distance matrix is $\infty$ then $v_i$ is a cut vertex. Similarly, by considering each pair of vertices, we obtain cut pairs of vertices.

EXAMPLE 6.16. For the $G_S^P$ in Figure 5.1, the cut vertices are $\{R_{15}, R_9\}$. Since $R_9$ and $R_{15}$ are the cut vertices, every tile paired with $R_9$ and $R_{15}$ forms a cut pair. The other cut pairs are $\{(R_1, R_2), (R_1, R_{10}), (R_2, R_3), (R_{13}, R_{14})\}$.

### 6.1.5 CHARACTERISTIC POLYNOMIAL AND EIGENVALUES

The next graph invariant is the characteristic polynomial of the adjacency matrix and from this polynomial, we derive eigenvalues.

The *characteristic polynomial* of a matrix $A_{n\times n}$ is the polynomial $p_A(x) = \det(xI - A)$, where **det** is the determinant and $I$ is the $n \times n$ identity matrix.

The *eigenvalues* of $A$ are the roots of its characteristic polynomial.

EXAMPLE 6.17. For the adjacency matrix corresponding to the $G_S^P$ in Figure 5.1, the characteristic polynomial and the eigenvalues are:

$p(x) = x^{16} - 29x^{14} - 26x^{13} + 278x^{12} + 422x^{11} - 955x^{10} - 2196x^9 + 763x^8 + 4044x^7 + 1407x^6 - 2316x^5 - 1538x^4 + 242x^3 + 315x^2 + 28x - 4.$

Eigenvalues: $-2.52$, $-2.37$, $-2.12$, $-1.47$, $-1.12$, $-1$, $-0.85$, $-0.75$, $-0.22$, $0.08$, $0.51$, $0.85$, $1.46$, $2.35$, $2.79$, $4.40$.

There are a lot of interesting results for the characteristic polynomial and the eigenvalues which can be associated with various geometrical interpretations. Some of the results are as follows.

PROPOSITION 6.18. *Suppose* $p_A(x)$ *is written as*

$$p_A(x) = x^n + c_1x^{n-1} + \cdots + c_n.$$

*Then*

1. $c_1 = 0$.
2. $-c_2$ *is the number of edges in the graph* $G$.
3. $-c_3$ *is twice the number of triangles in* $G$.

*Proof.* For a proof, see Biggs [3], Proposition 2.3. □

From above Proposition, we can say that the *characteristic polynomial of the adjacency graph* $G_S^R(n)$ is of the form $z^n - E_S^R(n)z^{n-2} - 2D_S^R(n)z^{n-3} + \cdots$, where $E_S^R(n)$ and $D_S^R(n)$ are the number of edges and number of triangles in the $G_S^R(n)$ respectively (cf. Theorem 3.13).

### 6.1.6 BIPARTITE GRAPHS

DEFINITION 6.19. A graph $G = (V, E)$ is called *bipartite* if $V$ admits a partition into two classes such that every edge has one end in each class: vertices in the same partition class must not be adjacent.

PROPOSITION 6.20. *A graph is bipartite if and only if its spectrum is symmetric about zero.*

*Proof.* For a proof, refer to Beineke and Wilson [2], Theorem 5.2. □

EXAMPLE 6.21. By using Proposition 6.21 for the obtained eigenvalues (Example 6.17), we conclude that the $G_S^P$ (Figure 5.1) is not bipartite.

### 6.1.7 CHROMATIC NUMBER

DEFINITION 6.22. A *vertex colouring* is an assignment of labels or colours to each vertex of a graph such that no edge connects two identically coloured vertices.

The *chromatic number* of a graph $G$ is the smallest number of colours needed to colour the vertices of $G$ so that no two adjacent vertices share the same colour.

EXAMPLE 6.23. The $G_S^P$ illustrated in Figure 5.1 has chromatic number 3.

The one-colourable graphs are totally disconnected, and two-colourability is a synonym for bipartiteness.

REMARK. For further details about many such graph invariants, we refer to [6], chapter 15 and Nanchen [31].

### 6.1.8 MOMENT OF INERTIA

Now we assume that the graph $G$ is weighted, i.e., each vertex $v_i$ has a weight $s_i$ which is a real number. In the case of tiling by rectangles, the weight

can for example be equal to the area of the corresponding tile; therefore it is always strictly positive.

For each vertex $v$, the *moment of inertia* of $G$ relative to this vertex was introduced by Coray in [6] and [7]. In analogy with classical mechanics, it is given by

$$I_v = \frac{1}{s}\sum_{i=1}^{n} s_i d(v, v_i)^2 ,$$

where $n$ is the number of vertices, $s = \sum s_i$ the total weight, and $d(v, v_i)$ the distance of vertex $v$ to vertex $v_i$. It is assumed that $s > 0$.

In classical mechanics, the moment of inertia is a measure of a rigid body's resistance to rotational acceleration about an axis. But of course, it is out of question to rotate a graph around a vertex. However, in architectural terms, the moment of inertia is a good indication of the relative heaviness of a building. A large room in the centre contributes very little to the moment of inertia while the same room far from the centre leads to big numbers.

$I_0$ is defined as half the weighted average of the individual moments of inertia:

$$I_0 = \frac{1}{2s}\sum_{i=1}^{n} s_i I_{v_i} = \frac{1}{2s^2}\sum_{i=1}^{n}\sum_{j=1}^{n} s_i s_j d(v_i, v_j)^2 .$$

$I_0$ can be regarded as a global covariant which expresses the quality of the construction, its compactness, relative ease of going from one room to another, etc.

For a distribution of mass in space, the *centre of gravity* is the unique point with the property that the weighted position vectors relative to this point sum to zero. In the case of a rigid body, it is well known that its motion through space can be described completely in terms the motion of the total mass concentrated at the centre of gravity and rotation of the object about its centre of gravity.

Interestingly, the *centre of gravity is also the point u where the moment of inertia is minimal.* In fact, one shows that $I_v = I_u + d(u, v)^2$ for any other point (see Coray and Pellegrino [6]). Then it is easy to see that the moment of inertia at the centre of gravity is equal to $I_0$. However, in the case of graphs, we can talk about vertices where the moment of inertia is minimal, but there is no other equivalent of a centre of gravity (see [6] or [7] for a detailed discussion).

One more invariant related to the moment of inertia is the *first-order moment*, which is defined by

$$I_v^1 = \frac{1}{s}\sum_{i=1}^{n} s_i d(v, v_i)\,.$$

Simply $d(v, v_i)^2$ is replaced by $d(v, v_i)$.

If we know the distance between any two vertices and the area of all the tiles, the moments of inertia and the first-order moments of a $G_S^P$ (with respect to each vertex) can easily be calculated.

EXAMPLE 6.24. For the $G_S^P$ in Figure 5.1, the moments of inertia with respect to all the vertices are given by (4.54, 2.70, 3.77, 7.83, 6.02, 4, 3.92, 5.86, 5.71, 11.49, 6.66, 8.16, 5.55, 4.05, 2.73, 10.70).

The first-order moments with respect to all the vertices are (1.87, 1.48, 1.79, 2.59, 2.29, 1.86, 1.88, 2.25, 1.92, 2.01, 2.24, 2.85, 2.07, 2.04, 1.41, 2.89).

Also, $I_0 = 2.61$. Clearly the tiles in the centre contributed less to the moments (e.g. $R_{15}$, $R_2$) in comparison to the tiles in the periphery (e.g. $R_{10}$, $R_{16}$).

For further details about these moments, refer to Coray and Pellegrino [6], Appendix and Coray [7].

REMARK. If we change the position or spiral used for each group, the *invariants* defined in this section behave as *covariants* for $G_S^P$ (or $T_S^P$).

## 6.2. COVARIANTS WITH RESPECT TO CHANGE OF POSITION AND SPIRAL

In Section 4.6, it is exposed that about 4 million $T_S^P$ can be constructed by changing the position of each group and the spiral to be used in each. But it is very difficult to single out one solution out of so many possibilities. Actually, it is not even feasible to select a really good one without some efficient criteria. Following is a list of some covariants which have already been defined with respect to changes in position and in the spiral for each group.

*1. Area*

Each solution may have a different total area. Therefore we may consider only those solutions having minimum or maximum area.

*2. The number of extra spaces*

For each solution, there are a certain number of extra spaces. Thus, solutions with either minimum or maximum number of extra spaces are sought out.

Similarly, we can consider solutions having:

*3. Maximum degree of connectivity*

*4. Maximum of mean of all vertex degrees*

*5. Minimum of mean of all distances*

*6. Minimum radius or minimum diameter*

*7. The moment of inertia and first-order moment*

For each solution, we calculate minimum of all moments of inertia and obtain solutions having minimum(minimum(moments of inertia)). To examine the range of minimum moments of inertia, we also compute maximum(minimum(moments of inertia)). Similarly, we obtain solutions having min(min(first-order moments)) and max(min(first-order moments)).

In the forthcoming section we proceed with one of our standard examples by obtaining the best solution on the basis of some of the covariants (e.g. *area* and *moment of inertia*).

## 6.3. Best solution for the change of a spiral

In this section, we consider covariants related to the change of a spiral, after each group has been positioned. As we saw in Section 4.6, $8^5$ different $T_S^P$ can be generated by changing the spiral for each group. Here we pick the best solution among $8^5$ solutions on the basis of *area* and *moment of inertia*.

### 6.3.1 The minimum and maximum area solutions

In this section, from $8^5$ possible $T_S^P$ only two $T_S^P$ will be taken into account, one with minimum area and one with maximum area. To obtain these two solutions, we first compute the area of all $T_S^P$ when we change spiral for every group. After obtaining the minimum and maximum values, the spiral used for each group is noted. After establishing the position and the spiral for each group, we construct the two required $T_S^P$.

EXAMPLE 6.25. Consider the same set of data we used in Example 4.7. We also fix the following position for each group:

1. *Central group:* $R_9, R_{16}$.
2. *Left group:* $R_1, R_2, R_7, R_6, R_{15}$.
3. *Upper group:* $R_3, R_4, R_8 and R_5$.
4. *Right group:* $R_{10}, R_{11}$.
5. *Lower group:* $R_{12}, R_{13}, R_{14}$.

Using the developed software (see Chapter 8), we found that for these specific groups and their fixed positions, there are 1024 solutions with minimum area, namely, 986. Since it is not feasible to display all the solutions, we select one and display it as an example. For this solution, the spirals for the central, left, upper, right and lower groups are *spiral*6, *spiral*8, *spiral*6, *spiral*6, *spiral*8 respectively. The computer-generated $T_S^P$ with minimum area is shown in Figure 6.1.

For the same groups and the same positions, there are 1024 solutions having maximum area, namely, 1501.2. Again, only one solution is chosen and put forth as an example. From this solution, the spirals for the central, left, upper, right and lower groups are *spiral*8, *spiral*8, *spiral*6, *spiral*8, *spiral*6 respectively. The computer-generated $T_S^P$ with maximum area is shown in Figure 6.2.

### 6.3.2 THE MINIMUM AND MAXIMUM OF MINIMUM(MOMENTS OF INERTIA)

For the moments of inertia of $G_S^P$ relative to each of the $n$ tiles (vertices) we determine their minimum and obtain min(min(moments of inertia)) and max(min(moments of inertia)) for all $8^5$ $G_S^P$. After having the position and spiral for each group corresponding to min(min(moments of inertia)) and max(min(moments of inertia)) $G_S^P$ respectively, we construct the required $T_S^P$.

EXAMPLE 6.26. Consider the same data and the same positions for each group as given in Example 6.25.

Using the developed software, we found that there are 64 $G_S^P$ with min(min(moments of inertia)), namely 1.79. The spirals for the central, left, upper, right and lower groups are respectively *spiral*7, *spiral*3, *spiral*3, *spiral*1, *spiral*1. And the area for corresponding $T_S^P$ is 1252.32. The computer-generated $T_S^P$ corresponding to min(min(moments of inertia)) $G_S^P$ is shown in Figure 6.3.

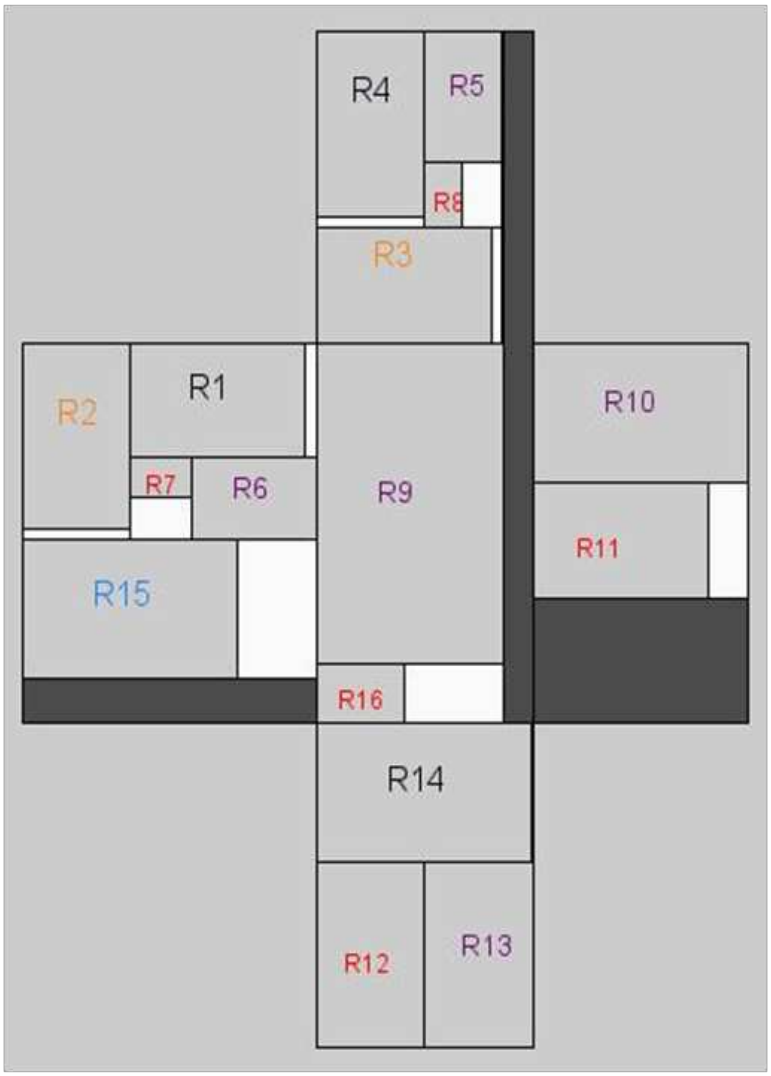


FIGURE 6.1
Minimum area $T_S^P$

Similarly, we found that there are 8 $G_S^P$ with max(min(moments of inertia)), namely 3.89. The spirals for the central, left, upper, right and lower groups are respectively *spiral*1, *spiral*6, *spiral*5, *spiral*8, *spiral*8. And the area for corresponding $T_S^P$ is 1165.68.

For the computer-generated $T_S^P$ corresponding to max(min(moments of inertia)) $G_S^P$, refer to Figure 6.4.


FIGURE 6.2
Maximum area $T_S^P$

REMARK. For the $T_S^P$ in Figure 6.1 and Figure 6.2, the minimum(moment of inertia) of corresponding $G_S^P$ is 2.77 and 2.31 respectively, as opposed to $T_S^P$ in Figure 6.3.

The four solutions mentioned above were compared on the basis of their area and moment of inertia. Since our prime focus is on the connectivity, we found that the $T_S^P$ corresponding to maximum(minimum(moment of inertia)) $G_S^P$ is least connected while the $T_S^P$ corresponding to minimum(minimum(moment of inertia)) $G_S^P$ and maximum area $T_S^P$ are most connected.

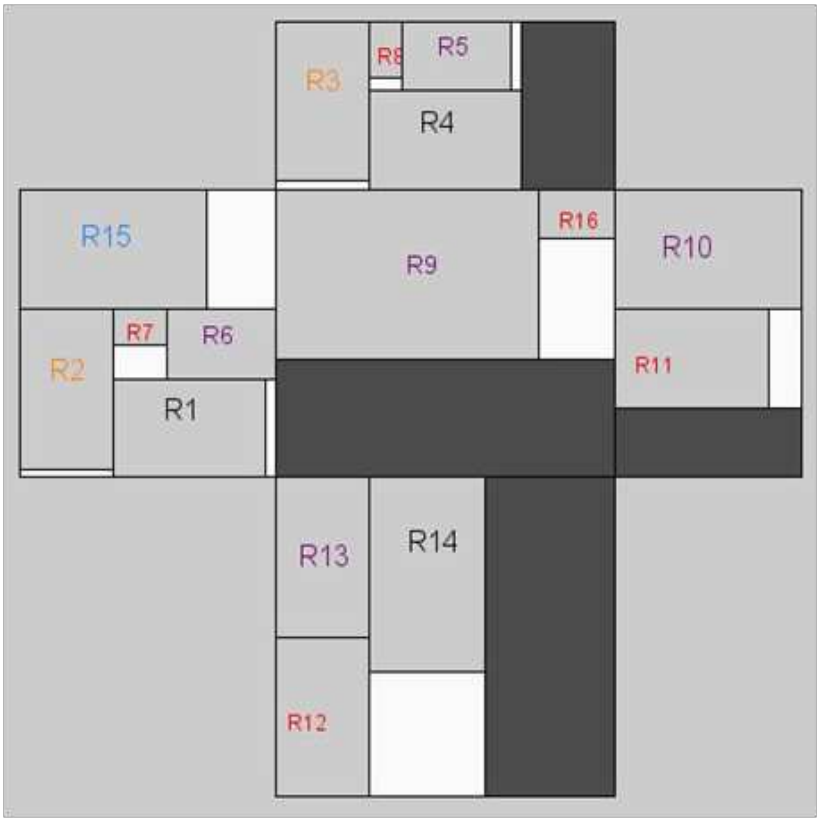


FIGURE 6.3
$T_S^P$ corresponding to min(min(moments of inertia)) $G_S^P$

## 6.4. REDUCING THE NUMBER OF SOLUTIONS

In Section 4.6 it is shown that a total number of $5! \times 8^5$ different $T_S^P$ can be generated. As discussed earlier, it is not possible to examine every $T_S^P$. That is why in Section 6.3 the best solution is obtained on the basis of a particular covariant. In this section, our concern is not about extracting the best solution but about providing a method to reduce this large number($5! \times 8^5$) to a smaller one, on the basis of some covariants (here *degree of connectivity*).

Let $D_i$ be the degree of connectivity of a $G_S^P$. There exist at most $5! \times 8^5$ values of degree of connectivity for $5! \times 8^5$ $G_S^P$.

LEMMA 6.27. *There exist at most* 13 $D_j$ *such that for* $i \neq j$, *we have* $D_j \neq D_i$ *where* $i = 1, 2, \ldots, 5! \times 8^5$ *when adjacency is considered among the members of different groups.*

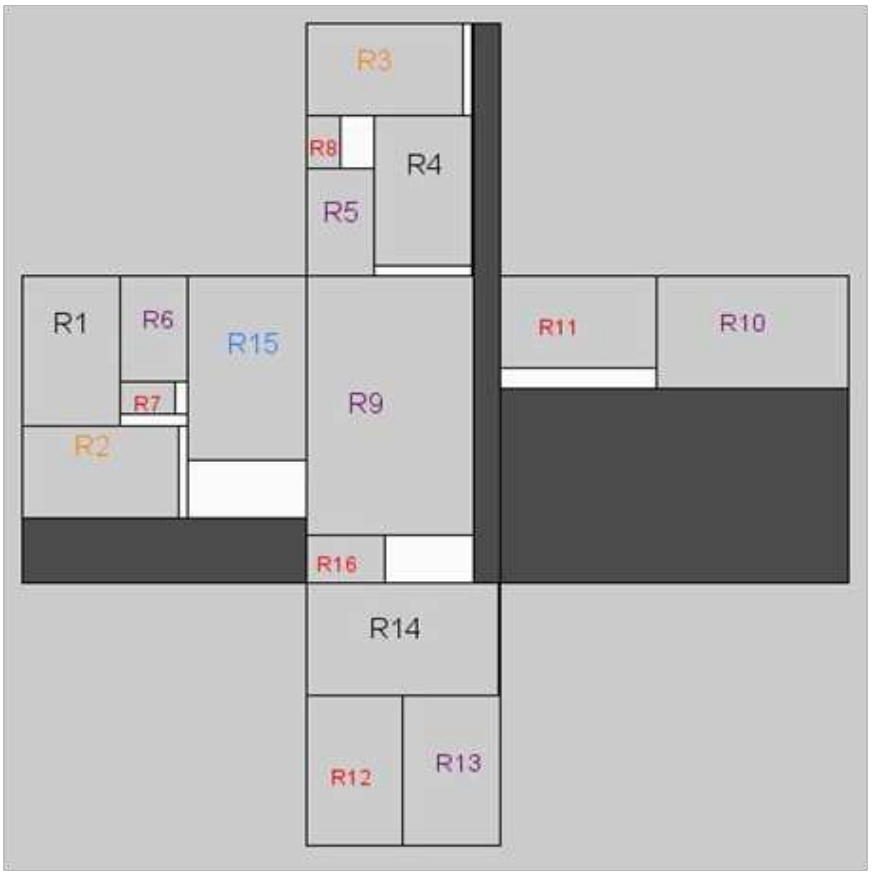


FIGURE 6.4
$T_S^P$ corresponding to max(min(moments of inertia)) $G_S^P$

*Proof.* From Corollary 5.12, a $G_S^P(n)$ can have at most $(3n-35)+16$ edges if each $T_R^S$ has at least four members. In this corollary, the number 16 is because of adjacency among the members of the different groups. It implies that in this case the number of edges can be at most 16. To construct a $T_S^P$, each group should have at least one tile. If all the groups have only one tile then the corresponding $G_S^P$ will have only 4 edges. It means that for a $G_S^P$, the minimum and maximum degree of connectivity are 4 and 16 when adjacency is considered among the members of different groups. Now to prove the result, we compute the other possible values for the degree of connectivity.

Suppose the number of tiles for the central, left, upper, right and lower $T_S^R$ is written as 5-tuple $(k_1, k_2, k_3, k_4, k_5)$. To discuss other possibilities for

the number of the edges, we provide a particular case for each possibility. Consider the following cases:

*1. If the number of tiles is* $(1,1,1,1,1)$*, then* $E_S^P = 4$

In this case, a member of the left, upper, right and lower $T_S^R$ is adjacent to a member of the central $T_S^R$. Therefore, $E_S^P = 4$.

*2. If the number of tiles is* $(1,2,1,1,1)$*, then* $E_S^P \leq 5$

Here, only 1 member of the upper, right and lower $T_S^R$ is adjacent to a member of the central $T_S^R$ but 1 or 2 members of the left $T_S^R$ can be adjacent to a member of the central $T_S^R$. Therefore, the $E_S^P$ can be 4 or 5.

*3. If the number of tiles is* $(1,2,2,1,1)$*, then* $E_S^P \leq 6$

Now, only 1 member of the right and lower $T_S^R$ is adjacent to a member of the central $T_S^R$ but 1 or 2 members of the left and upper $T_S^R$ can be adjacent to a member of the central $T_S^R$. Therefore, the $E_S^P$ can be 4 or 5 or 6.

*4. If the number of tiles is* $(1,2,2,2,1)$*, then* $E_S^P \leq 7$

In this case, only one member of the lower $T_S^R$ is adjacent to a member of the central $T_S^R$ but 1 or 2 members of the left, upper and right $T_S^R$ can be adjacent to a member of the central $T_S^R$. Therefore, the $E_S^P$ can be 4 or 5 or 6 or 7.

*5. If the number of tiles is* $(1,2,2,2,2)$*, then* $E_S^P \leq 8$

In this case, 1 or 2 members of the left, upper, right and lower $T_S^R$ can be adjacent to a member of the central $T_S^R$. Therefore, the $E_S^P$ can be at most 8.

**Note:** *In all the following cases, if the number of tiles is* $k$*, it means that the number of tiles is greater than 3.*

*6. If the number of tiles is* $(1,k,2,2,2)$*, then* $E_S^P \leq 9$

Here, at most 3 members of the left $T_S^R$ can be adjacent to a member of the central $T_S^R$. And 1 or 2 members of the upper, right and lower $T_S^R$ can be adjacent to a member of the central $T_S^R$. Therefore, the $E_S^P$ can be at most 9.

*7. If the number of tiles is* $(1,k,k,2,2)$*, then* $E_S^P \leq 10$

Here, at most 3 members of the left and upper $T_S^R$ can be adjacent to a member of the central $T_S^R$. And 1 or 2 members of the right and lower $T_S^R$ can be adjacent to a member of the central $T_S^R$. Therefore, the $E_S^P$ can be at most 10.

*8. If the number of tiles is* $(1,k,k,k,2)$*, then* $E_S^P \leq 11$

Now, at most 3 members of the left, upper and right $T_S^R$ can be adjacent to a member of the central $T_S^R$. And 1 or 2 members of the lower $T_S^R$ can be adjacent to a member of the central $T_S^R$. Therefore, the $E_S^P$ can be at most 11.

*9. If the number of tiles is* $(k,2,2,2,2)$*, then* $E_S^P \leq 12$

From the first point of Lemma 5.13, in this case $p_2$ get reduced by 4, hence the $E_S^P$ can be at most $16-4=12$.

*10. If the number of tiles is* $(k,k,2,2,2)$*, then* $E_S^P \leq 13$

Again from the first point of Lemma 5.13, in this case $p_2$ get reduced by 3, hence the $E_S^P$ can be at most $16-3=13$.

*11. If the number of tiles is* $(k,k,k,2,2)$*, then* $E_S^P \leq 14$

Again from the first point of Lemma 5.13, in this case $p_2$ get reduced by 2, hence the $E_S^P$ can be at most $16-2=14$.

*12. If the number of tiles is* $(k,k,k,k,2)$*, then* $E_S^P \leq 15$

Again from the first point of Lemma 5.13, in this case $p_2$ get reduced by 1, hence the $E_S^P$ can be at most $16-1=15$.

*13. If the number of tiles is* $(k,k,k,k,k)$*, then* $E_S^P \leq 16$

From Lemma 5.13, in this case the $E_S^P$ can be at most 16.

All the 13 cases concludes the proof. □

REMARK. In Section 6.4, the number $5! \times 8^5$ has been reduced to 13 on the basis of covariant *adjacency among the members of different groups*.

In this chapter, several graph invariants are studied and then the best solution has been obtained on the basis of some of them. Also, the number of solutions is reduced to some extent using a covariant. At this stage of our work, there is a lot of room for further development but right now we look forward for the application part of the work done. Therefore, in the next chapter we discuss the application of tiling by rectangles to the architectural field.

# CHAPTER 7

# APPLICATIONS TO ARCHITECTURE – SPACE ALLOCATION

As mentioned earlier in the introduction, tiling by rectangles can be applied for answering different problems related to various fields. As an illustration, in this chapter, we solve a problem which is common in architecture and for which no sufficient mathematical solution had been proposed.

DEFINITION 7.1. *Space allocation* is the computational arrangement of rooms (spaces) in a floor plan.

Space allocation is one of the most interesting and difficult areas of computer-aided architectural design. One of the most cherished tasks of architects is a layout of spaces in a building; they want it according to some rational principles (mostly style and minimization of distances between the spaces). For details refer to Kalay [23], Chapter 13.

In this chapter, we arrange the given spaces in a plus-shape floor plan by using the spiral-based plus-shape algorithm.

REMARK. In space allocation the *spaces* represent different elements of a building, e.g., rooms, offices, kitchens, bathrooms, WC etc. For simplicity only *rectangular* spaces are considered here.

## 7.1. RELATED WORK

Space allocation was one of the first design problems to be subjected to computational synthesis and several solutions have been proposed by many researchers. Tabor, 1970, ([27], Chapter 13) took initiative in this direction by categorising the space allocation problem into two approaches, the *additive approach* and the *permutation approach*. The additive approach starts with an empty floor plan and builds up a low cost layout with one activity at a time

while the other approach goes through every possible layout and searches for one having the least cost.

Galle, 1981 [14] published an algorithm which generates rectangular plans on modular grids, to provide all possible solutions. But we followed an approach which provides particular solutions only. In the past also, many researchers used particular approaches from various fields to obtain certain solutions. For example, in 1988, Lai et al. [25] used *graph theory* for floor plan design, in a study based on the reduction of the rectangular dualization problem to a matching problem on bipartite graphs. In 1988, Jo et al. [22] got their idea from *natural genetics*. They described a method based on constructing a evolutionary design model. On the other hand, Arvin et al. [1] in 2002 used the concept of *physics of motion*. They provided an approach where the designer creates a space plan by specifying and modifying graphic design objectives. Recently in 2010, Marson and Musse [28] introduced a room subdivision method based on *squarified treemaps*. The treemaps recursively subdivide an area into smaller areas, depending on some specific criteria.

### 7.2. A PLUS-SHAPE FLOOR PLAN INSIDE A RESIDENTIAL PLOT

DEFINITION 7.2. A *floor plan* is the most fundamental architectural diagram, an aerial view showing the arrangement of spaces in a building in the same way as a map, but focussing on the arrangement at a particular level (floor) of a building. In our context, the floor plan is simply a view from above showing the arrangement of spaces and extra spaces in a building.

DEFINITION 7.3. A *circulation* refers to the way people move through and interact within a building. Structures such as corridors, elevators, stairs are often referred to as the circulation elements.

A *terrace* is an outdoor living space.

In the spiral-based algorithm, the extra spaces are generated automatically. We consider that an extra space adjacent to the outside world is a terrace. Otherwise it is a circulation.

DEFINITION 7.4. A *residential plot* is a piece of land on which a building is erected. Other than the building itself, the plot can have some external elements like a garden, a parking etc.

Suppose given a rectangular residential plot. Here the aim is to obtain a plus-shape floor plan from the given rooms and allocation matrix, and then to introduce this floor plan into the plot. Afterwards, the remaining space is subdivided into eight rectangular parts so that they can be used for other purposes, like a parking, a garden, a swimming pool etc.

The plus-shape floor plan can be situated at any position with respect to the plot but for reasons of symmetry, we place the floor plan in the centre of the plot. Also, from an architectural point of view a house in the centre always has a 360-degree view.

In Section 4.4.2, it is shown that a plus-shape frame is made up of two big rectangles, $R_A$ and $R_B$. We have $L_B = L^2 + L^4 + \max(L^1, L^3, L^5)$ and $H_A = H^3 + H^5 + \max(H^1, H^2, H^4)$. Let $L^p$ and $H^p$ be the width and height of the plot.

Assuming that $L^p \geq L_B$ and $H^p \geq H_A$, we draw the given plot around the plus-shape floor plan and divide the remaining space into eight parts.

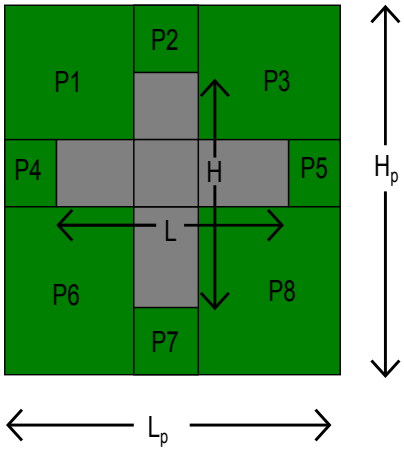


FIGURE 7.1
A plus-shape frame in a plot

For a better understanding refer to Figure 7.1, where a plus-shape frame within a plot and the eight different parts of the remaining space are shown. Clearly to draw a plot and its parts, the width, height and position of the plot and its parts are required which are obtained by using the calculations given in Section A.5.

## 7.3. AN EXAMPLE INTRODUCING A PLUS-SHAPE FLOOR PLAN IN A PLOT

EXAMPLE 7.5. In this example, we construct a plus-shape floor plan and its graph and place the floor plan in the centre of a residential plot.

We proceed by considering the same data as in Example 4.7. Just to have an architectural environment we introduce the following changes:

R1, R2, R3, R4 are rooms.

R5 and R6 are bathrooms, denoted by BA1 and BA2.

R7 and R8 are WC, denoted by WC1 and WC2.

R9 is a living room, denoted by LR.

R10 and R11 are a dining room and a kitchen, denoted by DR and KIT.

R12, R13 are studies and R14 is a library; they are denoted by OF1, OF2 and LIB.

R15 is a playroom, denoted by PR.

R16 is an entrance-hall, denoted by ENT.

The position, spiral and members of each group are:

*1. Central group: R3, R4, WC2 and BA1 with spiral*2.

*2. Left group: OF1, OF2 and LIB with spiral*5.

*3. Upper group: DR and KIT with spiral*8.

*4. Right group: R1, R2, WC1, BA2, PR with spiral*4.

*5. Lower group: LR and ENT with spiral*6.

Consider a square plot with area $2550m^2$. Here the width and height of the plot would be $50.5m$. For the obtained groups, $L_B = 42.3m$ and $H_A = 46.4m$. Clearly $L_B < L^p$ and $H_A < H^p$. Hence, we can place the obtained plus-shape floor plan in the given plot. For the computer generated plus-shape floor plan and plot, see Figure 7.3. The graph of this floor plan is shown in Figure 7.2.

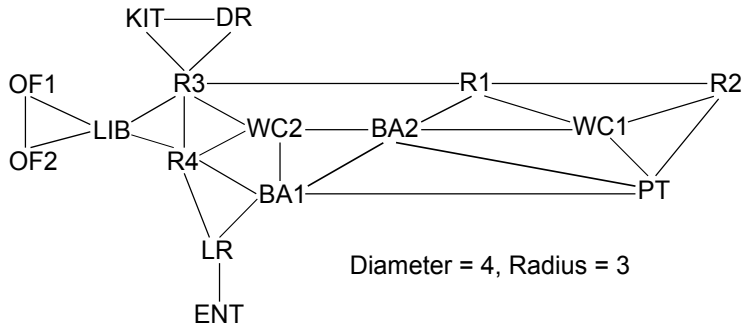


FIGURE 7.2

Graph of a spiral-based plus-shape floor plan in a plot

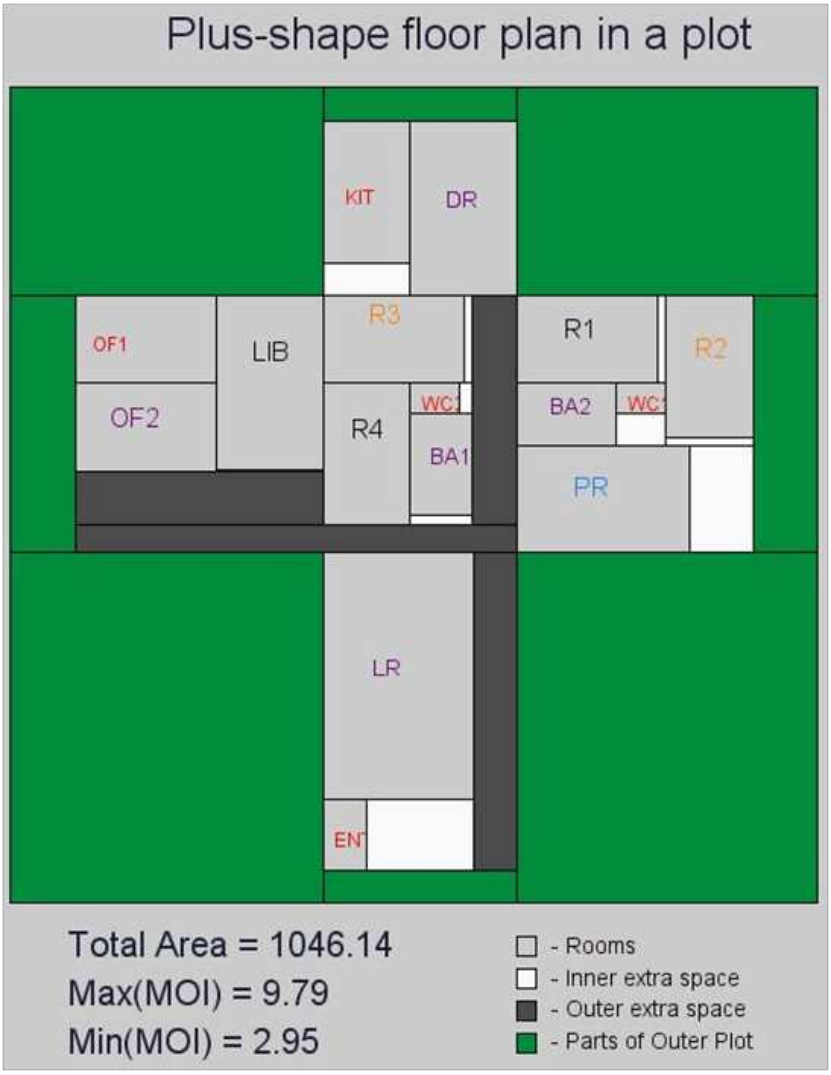


FIGURE 7.3
A spiral-based plus-shape floor plan in a plot

REMARK. The rectangle floor plan obtained from the spiral-based algorithm starts from the smallest space, i.e., the smallest space is always at the centre of the floor plan. Generally in a house also, the toilets are among the smallest spaces and one wants them to be in the centre and connected to a maximum number of spaces (inside a group). This shows that spiral-based rectangle tiling provides a solution for space allocation. Also, the position of extra spaces provides a hint for the position of some doors.

From here on, there is great scope for development. At this point, we aim to visualize the work done so far in this work. To this purpose, a piece of computer software has been developed which takes input from the user and outputs a $T_S^P$, its graph and all the associated covariants. In the next chapter, we discuss the steps involved in the development of this software.

# CHAPTER 8

# SOFTWARE FOR TILING BY RECTANGLES

In the preceding chapters we analysed $T_S^R$, $T_S^P$, their connectivity and some covariants associated with them. Along with this work, we worked on a piece of software which provides a $T_S^P$, *its graph and associated covariants* as output for a given set of data.

## 8.1. PROCESSING® LANGUAGE AND INTRODUCTION

DEFINITION 8.1. *Processing*® is an open source programming language and environment for people who want to create images, animations, and interactions (cf. `http://www.processing.org`).

It was developed by *Casey Reas* and *Benjamin Fry*, both formerly of the Aesthetics and Computation Group at the MIT Media Lab. Software written using Processing® is in the form of so-called *sketches*. These sketches are written in a specific *text editor*, which can have lots of tabs to manage different files.

After trying various other systems, we have written our code for the software in Processing®. This code is subdivided into several components to make it more comprehensive and each component is written in a separate file (tab). The software has two external files and 16 tabs in total; one external file for input and another one for output. In this chapter, we explain the functioning of each tab.

## 8.2. INPUT FOR THE SOFTWARE

The input for the code is extracted from an external file *input.txt*. While writing the code, input.txt is kept external so that it can be more user-friendly. This file has also been transformed into a *GUI (Graphic user interface)*. The file has the following eight different elements:

*1. An allocation matrix ($A_T(n)$)*

REMARK. The number of tiles($n$) and a list of all the tiles are specified within the code in a tab *gettinginput* instead of Input.txt file.

*2. The area of each tile*

*3. The area of the plot*

*4. The ratio of width over height for each tile*

*5. Change of a tile*

The formation of groups is done by using an algorithm but it can sometimes happen that one is not pleased with the formed groups. Therefore, we kept an option which enables one to move a tile from one group to another.

To move a tile from one group to another, three numbers are required. The first one is the group number from which its member is moved, the second one is another group number to which a new tile is added and the third number is the member number, i.e., the tile number as given in the list of tiles. For example if numbers $2, 4, 14$ are mentioned, this means $15^{th}$ tile is moved from $3^{rd}$ group to $5^{th}$ group. We write $-1$ as the tile number if we don't change the position of any tile. Also, at the present stage of development of our software at most two members can be moved.

*6. The position of groups*

As discussed in Chapter 4, there are only five positions for groups therefore the position of the groups is given in terms of five numbers. For example, the following sequence of five numbers $2, 0, 1, 3, 4$ indicates that $3^{rd}$, $1^{st}$, $4^{th}$ and $5^{th}$ groups are the central, left, upper, right and lower groups respectively.

REMARK. In Processing® an array always starts from zero, therefore in programming all numbering begins with zero.

*7. Assigning a spiral for each group*

As mentioned in Chapter 2, a group is constructed with any of the eight spirals therefore any five numbers between 0 and 7 stand for the spirals in each corresponding group. For example, the sequence $2, 1, 4, 5, 1$ indicates that the central, left, upper, right and lower groups are constructed with *spiral*3, *spiral*2, *spiral*5, *spiral*6 and *spiral*2 respectively. For details about the eight spirals, see Section 2.1.4.

## 8.3. Software Construction

In this section all tabs of the code are explained one by one in the order in which they occur.

### 8.3.1 Getting input

In the tab *gettinginput*, we import the input from the external file input.txt by calling a function *gettinginput()*.

Also using the areas of all the tiles and the ratio between their width and height, the width and height of each tile are computed.

### 8.3.2 Initial adjacency pairs

In the tab *initialadjacency* by calling function *initialadjacency()* all the initial adjacency pairs are obtained. For details, refer to Sections 4.2 and A.3.

### 8.3.3 Groups

Once we have the initial adjacency pairs, by using the tab *groups*, the required groups and their members are obtained. For details, refer to Sections 4.3 and A.4.

### 8.3.4 Change of a tile

The function of the tab *changingtile* is to move a tile from one group to another and simultaneously it revises the position of the tile in the corresponding array.

### 8.3.5 ARRANGING THE MEMBERS OF EACH GROUP IN ASCENDING ORDER

The tab *arrangingsizes* considers each group and its members one by one and then arranges them in the increasing order according to their areas (see Section 2.2.2.1).

### 8.3.6 OBTAINING A SPIRAL-BASED PLUS-SHAPE TILING

In the tab *plusshapedframe*, by calling function *plusshapedframe()* the required $T_S^P$ is constructed and displayed on the screen. This function has two parts, the first one does the necessary calculations while the second one deals with the construction of a $T_S^P$ inside the plot.

First part: **Interchanging the width and height of tiles and calculating the area of $T_S^P$**

This part has many steps, some of which may call some other functions. The details of these newly-defined functions are provided later.

*1. Set* $i = 0$
*2. Consider the* $(i+1)^{th}$ *group*

REMARK. For all the upcoming steps, the $1^{st}$, $2^{nd}$, $3^{rd}$, $4^{th}$ and $5^{th}$ groups represent the central, left, upper, right and lower groups respectively.

*3. Interchanging width and height of the* $1^{st}$ *member of each group*

This step is performed to reduce the area of $T_S^P$. It works well in most cases but sometimes it might work adversely.

For $i$ = 0, 1 or 3, namely, for $1^{st}$, $2^{nd}$ or $4^{th}$ group if $l_1 < h_1$, we swap $l_1$ and $h_1$. This is to reduce $H_B$.

For $i = 2$ or 4, namely, for $3^{rd}$ or $5^{th}$ groups, if $l_1 > h_1$, we swap $l_1$ and $h_1$. This is to reduce $L_A$. For $L_A$ and $H_B$, see Section 4.4.2.

*4. Interchanging width and height of the members of* $(i+1)^{th}$ *group*

If *spiral*1, *spiral*2, *spiral*5 or *spiral*6 is used for the $(i+1)^{th}$ group then we call function *shape*1(). If *spiral*3, *spiral*4, *spiral*7 or *spiral*8 is used for the $(i+1)^{th}$ group then we call function *shape*2().

The functions shape1() and shape2() swap the width and height of all the members of each group, *if required* (for condition if required refer to

Definition 2.20). For details about swapping of the width and height, refer to Section 2.2.2.

*5. Calculating the width and height of the* $(i+1)^{th}$ *group*

All the functions defined in this step compute the width and height of corresponding groups. If *spiral*1, *spiral*2, *spiral*5 or *spiral*6 is used for the $(i+1)^{th}$ group and if

5.1. $i = 0$, we call function L_and_H1_1() otherwise for remaining spirals we call function L_and_H2_1().

5.2. $i = 1$, we call function L_and_H1_2() otherwise for remaining spirals we call function L_and_H2_2().

5.3. $i = 2$, we call function L_and_H1_3() otherwise for remaining spirals we call function L_and_H2_3().

5.4. $i = 3$, we call function L_and_H1_4() otherwise for remaining spirals we call function L_and_H2_4().

5.5. $i = 4$, we call function L_and_H1_5() otherwise for remaining spirals we call function L_and_H2_5().

*6. If* $i = 4$*, we go to the next step otherwise we increase* $i$ *by one and go to step 2.*

*7. Computing the area of* $T_S^P$ *(cf. Section 4.4.2).* □

Going through the details of all the functions used in the first part is lengthy and tedious; therefore to understand the concept of all these functions, we shall elaborate on the steps of only two functions, namely, shape1() and L_and_H1_1(). These two functions are given in Sections A.6 and A.7.

Second part: **Drawing** $T_S^P$ **and plot**

1. Let $i = 0$.
2. Consider the $(i+1)^{th}$ group.
3. Calculating the width and height of the inner extra spaces

If $R_j$ is drawn to the left or right of $T_S^R(j-1)$ and $l_j$ is greater than the width of $T_S^R(j-1)$, we draw an inner extra space to the right side of $T_S^R(j-1)$. This extra space is a virtual part of $T_S^R(j-1)$ and it virtually increases the width of $T_S^R(j-1)$. To obtain the starting point of $(i+1)^{th}$ group (e.g. second group), the width of the inner extra space is subtracted from $x$.

If *spiral*1, *spiral*2, *spiral*5 or *spiral*6 is used for the $(i+1)^{th}$ group, we call function *shape*1_1().

If *spiral*3, *spiral*4, *spiral*7 or *spiral*8 is used for the $(i+1)^{th}$, we call function *shape*2_1().

Here the functions shape1_1() and shape2_1() calculate the width and height of some of the inner extra spaces of the $(i+1)^{th}$ group.

4. Obtaining starting point of each group and drawing the outer extra spaces

In this step first we compute the starting point of the $(i+1)^{th}$ group and then we draw an outer extra space if required.
For $i = 0$, 1, 2, 3, 4, we call functions extra1_1($p_1, p_2$), extra1_2($p_1, p_2, p_3, p_4$), extra1_3($p_1, p_2, p_3$), extra1_4($p_1, p_2$) or extra1_5($p_1, p_2$) respectively. Here $p_1, p_2, p_3$ and $p_4$ are variables which are passed to the corresponding function and the value of $p_1, p_2, p_3$ and $p_4$ may be different for each spiral.

5. Drawing the $(i+1)^{th}$ group

Corresponding to the *spiral*1, *spiral*2, *spiral*3, *spiral*4, *spiral*5, *spiral*6, *spiral*7 and *spiral*8, we call functions shape1_2(), shape2_2(), shape3_2(), shape4_2(), shape5_2(), shape6_2(), shape7_2() and shape8_2() respectively.

Each of these functions draws the corresponding group following the corresponding spiral at the starting point obtained in step 4.

6. If $i = 4$ we move to the next step otherwise we go back to the second step.

7. Drawing the plot and its parts (cf. Section A.5). □

Again explaining all the functions is an extensive and verbose process so for clarification of the functions, shape1_1(), extra1_2($p_1, p_2, p_3, p_4$) and shape1_2() are discussed. These functions are given in Sections A.8, A.9 and A.10 respectively.

REMARK. For the upcoming computations, it is not feasible to draw tiles again and again, therefore we allocate the tiles for the required computations. Allocating does not mean drawing, it means drawing virtually. Allocating the tiles instead of drawing them reduces the complexity of the code and the code consumes less time for displaying the final output.

### 8.3.7 OBTAINING THE FINAL ADJACENCY PAIRS

The tab *adjacencypairs* computes the adjacency pairs among the components of each group and among members of different groups on the basis of the definitions given in Sections 3.1 and 5.1 respectively.

Function: **adjacencypairs()**

*1. Set* $i = 0$ *and* $j = 0$ *where* $i$ *and* $j$ *are variables.*

*2. Obtaining adjacency pairs of the first group*

To obtain adjacency pairs of two different groups, after allocating each member we calculate the width and height of corresponding $T_S^R$.

If $j = 0$,

If *spiral*1, *spiral*2, *spiral*5 or *spiral*6 is used, the first and second tile is drawn one above the other. By calling function *adjacencycal1()* we compute the width and height of $T_S^R$ after allocating each member.

If *spiral*3, *spiral*4, *spiral*7 or *spiral*8 is used, the first and second tile is drawn side by side. By calling function *adjacencycal2()* we compute the width and height of $T_S^R$ after allocating each member.

If $i = 0$, to calculate adjacency pairs of the first group we call function *adjacency1()*.

*3. Obtaining adjacent pairs of* $(j+1)^{th}$ *group*

If $i = j + 1$,

For $(j+1)^{th}$ group, after allocating each member we compute the width and height of corresponding $T_S^R$ by calling any of the required functions *adjacencycal1()* or *adjacencycal2()*.

We compute adjacency pairs of $(j+1)^{th}$ group by calling function *adjacency1()*.

*4. Obtaining adjacency pairs among the first and second group*

If $i = 0$,

4.1 If $j = 0$, we obtain those members (and their heights) of the first group which can be adjacent to some members of the second group by calling function adjacencyHeight$(p_3, p_4)$.

The values of $p_3$ and $p_4$ are different for each spiral. Since the second group is drawn to the left of the first group, adjacency among the members of these two groups is obtained by comparing their heights.

4.2 Obtaining the members (and their heights) of the second group which can be adjacent to the members of first group and then computing the adjacency pairs among these two groups

If $j = 1$, we first check which spiral is used and then call function adjacencyHeight$(p_3, p_4)$ (the values of $p_3$ and $p_4$ are different for each spiral).

Afterwards we call function findingadjacencies$(p_1, p_2)$. This function computes the adjacency pairs among the members of two different groups. Here it computes the adjacency pairs among the members of the first and second group. The value of $p_1$ and $p_2$ can be same or different for different spirals.

*5. Obtaining adjacent members of the first and third group*

In this case we check if $i = 1$, $j = 0$ for the first group and $j = 2$ for the third group. To obtain steps 5.1 and 5.2, we replace function adjacencyHeight$(p_3, p_4)$ by function adjacencyLength$(p_3, p_4)$ in the steps 4.1 and 4.2.

Since the third group is drawn above the first group, we compare the widths of the members of these two groups and that is why we replaced adjacencyHeight$(p_3, p_4)$ by adjacencyLength$(p_3, p_4)$.

*6. Obtaining adjacent members of the first and fourth group*

In this case we check if $i = 2$, $j = 0$ for the first group and $j = 3$ for the fourth group. Steps 6.1 and 6.2 are same as the steps 4.1 and 4.2.

*7. Obtaining adjacent members of the first and fifth group*

In this case we check if $i = 3$, $j = 0$ for the first group and $j = 4$ for the fifth group. Steps 7.1 and 7.2 are same as the steps 5.1 and 5.2.

*8. If $j < 4$, we increase $j$ by one and go to step 2. If $j = 4$, we increase $i$ by one. If $i < 4$, we consider $j = 0$ and go to step 2 otherwise stop the process.* □

Explaining all the functions is an extensive and verbose process so for clarification of the functions adjacencycal1(), adjacency1(), adjacencyHeight$(p_3, p_4)$ and findingadjacency$(p_1, p_2)$ are discussed. All these functions are given in Sections A.11, A.12, A.13 and A.16 respectively.

### 8.3.8 OBTAINING COVARIANTS ASSOCIATED WITH THE GRAPHS

The next five tabs are *adjacencymatrix, distance, cutvertex, eccentricity and momentofinertia.* These functions furnish the following results in accordance with the definitions and methods of Section 6.1.

1. *Function adjacencymatrix()*

1.1 This function computes the *adjacency matrix* from obtained adjacency pairs.

1.2 From the adjacency pairs it calculates *degree of connectivity* of the graph.

1.3 From the adjacency matrix, it obtains the *degree* of each vertex of $G_S^P$ and then the mean, standard deviation, maximum and minimum of all degrees.

2. *Function distance()*

This function calculates the *distance* between any two vertices of $G_S^P$ and then the mean, standard deviation, maximum and minimum of all distances.

3. *Function cutvertex()*

This function computes all the *cut vertices* and *cut pairs* of $G_S^P$.

4. *Function eccentricity()*

This function first provides the *eccentricity* of each vertex of $G_S^P$ and then calculates the *diameter, radius* and *centre* of $G_S^P$. At the end, it computes the mean and standard deviation of all eccentricities.

5. *Function momentofinertia()*

We compute moments of $G_S^P$ relative to each vertex by the following two ways:

1. by considering the weight of each tile equal to its area,

2. by considering the weight of each tile as one unit.

As noticed by Coray [6] the moments of inertia generally provide a more accurate measure of the centre of $G_S^P$ than eccentricity, even when the graph is equipped only with the trivial weighting.

After having the first-order moments and the moments of inertia of $G_S^P$ relative to each vertex, we compute the mean, standard deviation, maximum and minimum of all moments.

### 8.3.9 OBTAINING EIGENVALUES

From the tab *eigenvalue*, we obtain the eigenvalues of the adjacency matrix and its characteristic polynomial, by using inbuilt library *Jama*. Afterwards the maximum and minimum of all eigenvalues are computed. For details about the eigenvalues, refer to Section 6.1.5.

### 8.3.10 DRAWING THE GRAPH

Using the tab *drawinggraph*, we draw the $G_S^P$ on the same screen on which $T_S^P$ is displayed.

### 8.3.11 PRINT

Using the tab *print*, all the results are displayed in an external file *output.txt*. A list of these results is given in Section 8.4. In this tab the following calculations are made:

1. Using the inbuilt library *jgrapht*, we obtain the *chromatic number* of $G_S^P$.

2. By means of the paths for calculating distances, we compute a *shortest path* between each pair of vertices of $G_S^P$.

3. With the eigenvalues and Proposition 6.21, we compute whether $G_S^P$ is *bipartite* or not.

### 8.3.12 CALLING ALL FUNCTIONS

In the tab *spiralbasedallocation* the function *setup()* calls all the main functions defined in Sections 8.3.1 to 8.3.11.

## 8.4. OUTPUT FOR THE SOFTWARE

When we run the obtained software, a $T_S^P$ and its graph are displayed on the screen. In addition, some important covariants, like the area of $T_S^P$, spirals used for the central, left, upper, right and lower $T_S^R$, the minimum and maximum moments of inertia, the radius and diameter of $G_S^P$ are also displayed. At the same time, a new file output.txt is obtained, which contains the following results:

*1. The width and height of each tile*
*2. All the five groups and their members*

*3. The number of inner and outer extra spaces*

*4. The area of* $T_S^P$

*5. The total area of all the extra spaces*

*6. The adjacency matrix*

*7. The number of edges*

*8. The degrees of all tiles, their mean, standard deviation, dispersion, maximum and minimum.*

*9. The eigenvalues of adjacency matrix and the corresponding polynomial. Also, the minimum and maximum of all eigenvalues.*

*10. Whether the* $G_S^P$ *is bipartite or not*

*11. The distance matrix and the mean, standard deviation, dispersion, maximum and minimum of all distances*

*12. A shortest path between each pair of tiles*

*13. All the cut vertices and cut pairs*

*14. The eccentricities of all tiles, their mean, standard deviation, and dispersion. The radius, diameter and centre of* $G_S^P$

*15. The moments and their mean, standard deviation, dispersion, maximum and minimum.*

*16. The chromatic number*

## 8.5. SOURCE FOR THE CODE

As mentioned before our code for this software is written in the programming language Processing® therefore the first requirement is to download the most recent version of Processing®. This can be done from the following address:

`http://www.processing.org/download/`

The software *TOR* has six different forms, all of them can be downloaded from my web page:

`http://krishnendrashekhawat.weebly.com/downloadable.html`

Now we elaborate on these six different forms.

**1. General Solution**

For a given position and spiral for each group, a $T_S^P$ is displayed along with its *graph* by running this code. All the associated *covariants* are calculated and displayed in the file *output.txt*. This code is available under the name **spiralbasedtiling.zip**.

**2. Minimum Area Solution**

As discussed previously when the position of the groups is fixed and the spiral for each group is changed, $32,768$ different solutions can be obtained. For these solutions, the following code gives the minimum area solution. There can be more than one solution having minimum area, therefore in the file output.txt, the number of minimum area solutions is mentioned.

For details about minimum area solution refer to Section 6.3.1, where an example is also given.

By default this code gives the last solution having minimum area. But by replacing this specific line

```
// if(No_Soln == 1) { g0 = 8; g1 = 8; g2 = 8; g3 = 8; g4 = 8;
break; }
```

by the line

```
if(No_Soln == 1) { g0 = 8; g1 = 8; g2 = 8; g3 = 8; g4 = 8; break;
}
```

in the tab *plusshapedframe*, the first solution having a minimum area is obtained. Similarly, other solutions can be explored by replacing `No_Soln == 1` by some other value $j$ such that $1 < j <$(number of minimum area solutions).

This code is available under the name **minareatiling.zip**.

REMARK. While obtaining a minimum area solution, all the $32,768$ solutions are not displayed, only the area for all solutions is calculated. Then minimum area solution is selected and displayed. The same procedure is used for the next 3 forms of the code.

To display all the solutions, one way is to display and erase each solution one by one; other way is to display all the solutions in multiple windows. If we calculate the area of all the solutions and display only one solution having minimum area then this reduces the complexity of the code and it consumes less time for displaying the final output.

**3. Maximum Area Solution**

This code provides a maximum area solution, its graph and associated covariants (for details, see Section 6.3.1). The different solutions having maximum area are obtained by making the same changes as above. This code is available under the name **maxareatiling.zip**.

**4. Min(min(moments of inertia)) Solution**

This code provides min(min(moments of inertia)) solution, its graph and the associated covariants (for details, refer to Section 6.3.2). This code is available under the name **minmoitiling.zip**.

**5. Max(min(moments of inertia)) Solution**

This code bears the name **maxmoitiling.zip**.

**6. General GUI Solution**

This code is exactly the same as the code for the general solution. The only difference is that after running this code a user interface window named *GUI* will be created, where the position of groups and spiral used for each group can be chosen. While using this window, the instructions on the window should be read and followed strictly. This code is available under the name **tilinggui.zip**.

**7. Libraries**

For all these codes, three libraries are required as follows:

**7.1 jgrapht**

This library is used to calculate the chromatic number.

**7.2 Jama**

This library is used to compute the eigenvalues.

**7.3 controlP5**

This library is used to obtain a graphic user interface.

All these libraries are available through **libraries.zip**, which can be downloaded from the site

`http://www.processing.org`

All these codes and libraries should be downloaded in a new folder only, for example Processing at *My Documents*. For syntax and other details about Processing® refer to Terzidis [40].

Our software will be updated with any future advancement in the theory developed here.

# CONCLUSION AND FUTURE WORK

An arrangement of objects is an elementary problem in daily life and also in the context of a larger framework. Therefore, there is a necessity to handle this problem and that's why in the present work a particular aspect of this problem, namely tiling by rectangles has been defined and handled mathematically. The process of solving this problem was commenced by arranging several tiles of different sizes in a rectangular frame, i.e., by constructing a $T_S^R$. Since the obtained $T_S^R$ was recognised as one of the best solutions from the point of view of connectivity, we used $T_S^R$ as a base for obtaining other shape tilings. To demonstrate the relevance of this approach, a $T_S^P$ was constructed from five $T_S^R$ (cf. Chapter 4).

After constructing $T_S^R$ and $T_S^P$, as an advancement in this direction, we will try to pursue the following five steps:

*1. We will obtain tiling for given tiles (rectangular) and for a given shape which can easily be divided into rectangles, for example, a plus-shape was partitioned into five rectangles (see Figure 4.1).*

*2. In the next step, we will obtain tiling for a given shape and area which can easily be partitioned into rectangles.*

*3. After that, we will obtain tiling for a given shape which is made up of straight lines but it can not be partitioned into rectangles. For example, a triangle can't be partitioned into rectangles.*

*4. Then we will obtain tiling for a given shape which is not made up of straight lines; for example, a circle.*

*5. At last, we will try to obtain tiling by polygons.*

For the arrangement of tiles in a tiling, we have taken into account adjacency among the tiles and tried to obtain tilings having best degree of connectivity. In this direction, we have obtained $T_S^R$ which is one of the best rectangle tilings from the point of view of connectivity (cf. Section 3.3.2).

We know that in the spiral-based algorithm an extra space is generated automatically and we have defined some methods to reduce the size of extra spaces such that connectivity remains preserved (cf. Section 2.2). Because of

this, when we apply the concept of tiling by rectangles to a new field for example to architecture (to obtain a floor plan), we do not need to worry about position of extra spaces. In future, we would develop some more methods to reduce the size of extra spaces such that connectivity remains preserved.

For generalization of $T_S^R$ to obtain other shape tilings, we have considered the allocation matrix (see Section 4.1.1) which behaves as a set of constraints and worked very well for solving the problem and for refining solutions. In future, we will try to develop more constraints, for example, constraints can be on the number of extra spaces in the tiling or on the number of cut vertices in the corresponding graph of the tiling.

Using allocation matrices and combining different $T_S^R$, we have obtained different shape tilings and for a particular shape (e.g. plus-shape), we have proposed several solutions (cf. Section 4.6). But as said before it is hard to come across every solution thus to refine solutions, we have developed many covariants associated with the tiling and we have studied a lot of graph covariants. On the basis of some covariants, we have proposed some techniques to reduce the number of solutions (cf. Section 6.4) and obtained one of the best solutions for a particular covariant (cf. Section 6.3).

Each covariant has its own function and can be used independently and there are lots of covariants therefore using some concepts of statistics, we will derive some techniques to obtain a solution that would be best on the basis of all covariants.

As an example of an application, the concept of tiling by rectangles has been applied to architecture for obtaining floor plans. In future we will try to construct multi-floor plans by adding stairs etc.

To envision the work done a piece of software was also developed. The best part of this software is that it requires no prior knowledge of mathematics, so that it can easily be used by everyone. In future we will put continuous effort to update and improve the software so as to make it even more user-friendly.

We have raised many questions throughout the text and we hope to get some interesting suggestions or solutions in future.

## REFERENCES


[1] ARVIN, S. A., and D. H. HOUSE. Modeling architectural design objectives in physically based space planning. *Automation in Construction*, 11, 2, 213 – 225, 2002.

[2] BEINEKE, L. W. and R. J. WILSON. *Topics in Algebraic Graph Theory.* Cambridge University Press, 2004.

[3] BIGGS, N. *Algebraic Graph Theory (Second Edition).* Cambridge University Press, 1993.

[4] BOUWKAMP C. J. and A. J. W. DUIJVESTIJN. *Catalogue of Simple Perfect Squared Squares of orders 21 through 25.* EUT Report 92-WSK-03 Eindhoven, 1992.

[5] BROOKS, R. L., C. A. B. SMITH, A. H. STONE and W. T. TUTTE. The dissection of rectangles into squares. *Duke Math. J.7*, 312 – 340, 1940.

[6] CORAY, D., P. PELLEGRINO et AL. *Arquitectura e Informática.* Gustavo Gili, Barcelona, 1999.

[7] CORAY, D. Les invariants associés aux graphes en architecture. *Preprint, Section de Mathématiques, Université de Genève,* 11 – 16, *2009.*

[8] CRAAL. La conceptualisation de l'espace en architecture (rapport au FNSRS). *Université de Genève, 1995.*

[9] DEHN, M. Über die Zerlegung von Rechtecken in Rechtecke. *Math. Ann. 57*, 314 – 332, 1903, (in German).

[10] DIESTEL, R. *Graph Theory (Third Edition).* Springer-Verlag Berlin Heidelberg, 2006.

[11] DUDENEY, H. E. *The Canterbury Puzzles.* Thomas Nelson and Sons, New York, 2002.

[12] DUNLAP, R. A. *The Golden Ratio and Fibonacci Numbers.* World Scientific, Singapore, 1997.

[13] FORMALISATION ET SENS DU PROJET ARCHITECTURAL. *FNSRS, Université de Genève, 2008.*

[14] GALLE, P. An algorithm for exhaustive generation of building floor plans. *Commununications of the ACM,* 24, 12, 813 – 825, 1981.

[15] GRAPH THEORY & ITS APPLICATIONS. *Instructional Workshop, Institute of Mathematics & Applications, Bhubaneswar, India, 2009.*

[16] GROSS, J. L. and J. YELLEN. *Graph Theory and Its Applications (Second Edition).* Chapman & Hall/CRC, Boca Raton, 2006.

[17] GRÜNBAUM, B. and G. C. SHEPHARD. *Tilings and Patterns (An Introduction).* W. H. Freeman and Company, New York, 1989.

[18] HARDY, G. H. and E. M. WRIGHT. *An Introduction To The Theory Of Numbers.* Oxford at the Clarendon Press, 1938.

[19] HARRIS J., J. L. HIRST and M. MOSSINGHOFF. *Combinatorics and Graph Theory (Second Edition).* Springer, 2008.

[20] HUNTLEY, H. E. *The Divine Proportion (A Study in Mathematical Beauty).* Dover Publications, Inc., New York, 1970.

[21] INTERNATIONAL CONFERENCE ON RECENT TRENDS IN GRAPH THEORY AND COMBINATORICS. *Department of Mathematics, Cochin University of Science & Technology, India, 2010.*

[22] JO, J. H. and J. S. GERO. Space layout planning using an evolutionary approach. *Artificial Intelligence in Engineering*, 12, 149 – 162, 1998.

[23] KALAY, Y. E. *Architecture's New Media (Principles, Theories, and Methods Of Computer-Aided Design).* The MIT Press, 2004.

[24] KORF, R. Optimal rectangle packing: new results. *American Association for Artificial Intelligence*, 1 – 5, 2004.

[25] LAI, T.-T., AND S. M. LEINWAND. Algorithms for floorplan design via rectangular dualization. *IEEE Transactions on Computer-Aided Design*, 7, 12, 1278 – 1289, 1988.

[26] LOI, M(ED). *Mathématiques et Art.* Hermann, Paris, 1995.

[27] MARCH, L. and P. STEADMAN. *The Geometry of Environment (An introduction to spatial organization in design).* Methuen & Co Ltd, London, 1971.

[28] MARSON, F. and S. R. MUSSE. Automatic generation of floor plans based on squarified treemaps algorithm. *International Journal of Computer Games Technology,* 3 – 9, *2010*.

[29] MITCHELL, W. J. *The Logic of Architecture (Design, Computation, and Cognition).* The MIT Press, 1994.

[30] MOROŃ, Z. *O Rozkladach Prostokatow Na Kwadraty (On the Dissection of a Rectangle into Squares).* Prezeglad Mat. Fiz. 3, 152 – 153, 1925.

[31] NANCHEN, E. *Distances entre classes d'isomorphisme de graphes.* Thèse no. 3012, Section de Mathématiques, Université de Genève, 1998.

[32] OPTIMAL DISCRETE STRUCTURES AND ALGORITHMS. *University of Rostock, Germany, 2010.*

[33] PAPENFUS, M. Tiling rectangles with squares (for Dummies). The University of Arizona, 2010.

[34] PEMMARAJU, S. and SKIENA, S. *Computational Discrete Mathematics (Combinatorics and Graph Theory with Mathematica).* Cambridge University Press, 2003.

[35] PRAKASH, V. *Chandigarh's Le Corbusier (The Struggle for Modernity in Postcolonial India).* The University of Washington Press, 2002.

[36] RODRIGUE, J. P., C. COMTOIS and B. SLACK. *The Geography of Transport Systems*. Routledge, New York, 2006.

[37] SPIVAK, M. D. *The Joy of Tex (Second Edition).* American Mathematical Society, 1990.

[38] STINY, G. *Shape (Talking about Seeing and Doing).* The MIT Press, 2006.

[39] TERZIDIS, K. *Algorithmic Architecture.* Architectural Press Elsevier, New York, 2006.

[40] TERZIDIS, K. *Algorithms For Visual Design (Using the Processing Language).* Wiley Publishing, Inc, Indianapolis, 2009.

[41] TUTTE, W. T. *Graph Theory As I Have Known It.* Oxford University Press, 1998.

[42] WALSER, H. *The Golden Section.* The Mathematical Association of America, 2001.

[43] WEYL, H. *Symmetry.* Princeton University Press, 1952.

# LIST OF FIGURES

# APPENDIX

## A.1 DEGREE SEQUENCE

From the degree of all vertices, there follows one more graph invariant: the degree sequence.

DEFINITION A.1. The *degree sequence* of a graph is the sequence formed by arranging the vertex degrees in non-increasing order.

DEFINITION A.2. A sequence $\langle d_1, d_2, \ldots, d_n \rangle$ is said to be *graphic* if there is a permutation of it which is the degree sequence of some simple graph. Such a simple graph is said to *realize* the sequence.

Given the degree of all the vertices, we can construct a graph using the following two results.

THEOREM A.3. *Let $\langle d_1, d_2, \ldots, d_n \rangle$ be a graphic sequence, with $d_1 \geq d_2 \geq \cdots \geq d_n$. Then there is a simple graph with vertex-set $\{v_1, v_2, \ldots, v_n\}$ satisfying* $\deg(v_i) = d_i$ *for $i = 1, 2, \ldots, n$, such that $v_1$ is adjacent to vertices $v_2, \ldots, v_{d_1+1}$.*

*Proof.* For a proof, refer to Jonathan and Yellen [16], Theorem 1.1.6. □

COROLLARY A.4 [Havel(1995) and Hakimi(1961)]. *A sequence $\langle d_1, d_2, \ldots, d_n \rangle$ of nonnegative integers such that $d_1 \geq d_2 \geq \cdots \geq d_n$ is graphic if and only if the sequence $\langle d_2 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n \rangle$ is graphic.*

*Proof.* For a proof, refer to Jonathan and Yellen [16], Corollary 1.1.7. □

To illustrate these results, we consider an example where the graph is constructed for the given degrees of all vertices. For details about the algorithm used in Example A.5, refer to Jonathan and Yellen [16], Algorithm 1.1.1 and Example 1.1.12.

EXAMPLE A.5. Let's consider a graph having five vertices such that
$\deg(v_1) = 2,\ \deg(v_2) = 3,\ \deg(v_3) = 2,\ \deg(v_4) = 4,\ \deg(v_5) = 3.$

By arranging the above degrees in non-increasing sequence, the following sequence is derived: $\langle 4, 3, 3, 2, 2\rangle$. Now using Corollary A.4, for this degree sequence we construct a graph.

1. Initially the degree sequence is $\langle 4, 3, 3, 2, 2\rangle$. Here $d_2 = 3 \implies d_2 - 1 = 2, d_1 = 4 \implies d_{d_1+1} = d_5 = 2$. Using Corollary A.4, we obtain the following sequence:

2. $\langle d_2 - 1, d_3 - 1, d_4 - 1, d_5 - 1\rangle = \langle 2, 2, 1, 1\rangle$

Using Corollary A.4 once more, we obtain the following sequence:

3. $\langle 1, 0, 1\rangle$

On permuting this sequence, we get:

$\langle 1, 1, 0\rangle$

Using Corollary A.4 once more, we obtain the following sequence:

4. $\langle 0, 0\rangle$ which is graphic.

Now to construct the required graph,we follow the above four steps in reverse order.

4. $\langle 0, 0\rangle$

This step implies that there are two isolated vertices as in Figure A.1(A).

3. $\langle 1, 1, 0\rangle$

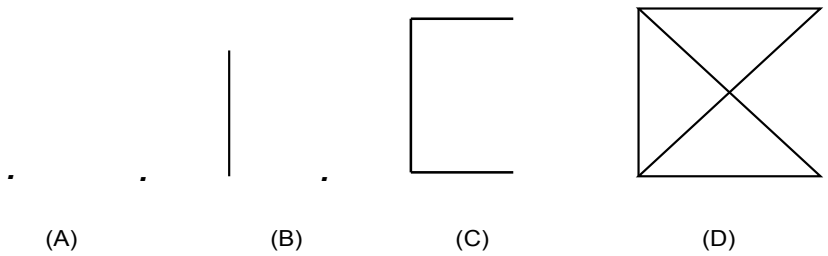


FIGURE A.1
Graphs obtained from the given degree sequence

This step implies that there are three vertices, two of which have degree 1 while one vertex has degree 0. The graph for this sequence has an edge between the two vertices from the previous step and one isolated vertex; see Figure A.1(B).

2. $\langle 2, 2, 1, 1 \rangle$

The graph of this sequence is constructed from the graph of the previous step by adding an edge between $v_1$ and $v_3$, and $v_2$ and $v_4$, as shown in Figure A.1(C).

1. $\langle 4, 3, 3, 2, 2 \rangle$

The graph of this sequence is obtained from Figure A.1(C) by adding an edge between $v_5$ and the remaining vertices, as shown in Figure A.1(D).

The graph in Figure A.1(D) realizes the given sequence.

## A.2 FLOYD'S ALGORITHM

This algorithm (cf. Pemmaraju and Skiena [34], Chapter 8) is used to obtain the distance and a shortest path between any two vertices.

The algorithm works by updating two matrices, $D_k$ and $Q_k$, $n$ times for an $n$-vertex graph. The matrix $D_k$, in any iteration $k$, gives the value of the shortest distance among all pairs of vertices $(i, j)$ as obtained till the $k^{th}$ iteration. The matrix $Q_k$ has $q_{ij}^k$ as its elements. The value of $q_{ij}^k$ gives the immediate predecessor vertex from vertex $i$ to vertex $j$ on the shortest path as determined by the $k^{th}$ iteration. The starting matrix $D_0$, with entries $d_{ij}^0$, is defined as follows:

$d_{ij}^0 = 1$ if $i \neq j$ and vertex $i$ is adjacent to vertex $j$
$d_{ij}^0 = \infty$ if $i \neq j$ and vertex $i$ is not adjacent to vertex $j$
$d_{ij}^0 = 0$ if $i = j$

The entries $q_{ij}^0$ of the predecessor matrix $Q_0$ are defined as follows: $q_{ij}^0 = i$, for $i \neq j$, i.e., for every pair of distinct vertices $(i, j)$, the immediate predecessor of vertex $j$ on a shortest path leading from vertex $i$ to vertex $j$ is (temporarily) assumed to be vertex $i$. After defining $D_0$ and $Q_0$ the following steps are used repeatedly to determine $D_n$ and $Q_n$.

Step 1: Set $k = 1$

Step 2: The entries $d_{ij}^k$ of the shortest path matrix $D_k$ are defined by:

$$d_{ij}^k = \min(d_{ij}^{k-1}, {d_{ik}}^{k-1} + {d_{kj}}^{k-1})$$

.

Step 3: The entries $q_{ij}^k$ of the predecessor matrix $Q_k$ are defined as follows:
If $d_{ij}^k \neq d_{ij}^{k-1}$ then $q_{ij}^k = {q_{kj}}^{k-1}$ else $q_{ij}^k = q_{ij}^{k-1}$.

Step 4: If $k = n$, the algorithm is terminated. If $k < n$, increase $k$ by 1, and return to step 2.

Now we take a look at the algorithm in a little more detail. In step 2, each time one goes through the algorithm, it is checked whether a shorter path exists between vertex $i$ and vertex $j$. In step 3, if it is established that $d_{ij}^k \neq d_{ij}^{k-1}$, i.e., the length of the shortest path $d_{ij}^k$ between vertices $i$ and $j$ is less than the length of the shortest path $d_{ij}^{k-1}$, it is required to change the immediate predecessor vertex to vertex $j$. Since the length of the new shortest path is:

$$d_{ij}^k = {d_{ik}}^{k-1} + {d_{kj}}^{k-1}$$

it is clear that here node $k$ is the new immediate predecessor vertex to $k$, and therefore:

$$q_{ij}^k = {q_{kj}}^{k-1}$$

After passing through the algorithm $n$ times, the entries $d_{ij}^n$ of the final matrix $D_n$ will constitute a shortest path going from vertex $i$ to vertex $j$.

Matrix $Q$ gives the immediate predecessor vertex to vertex $j$ on the shortest path. To have all vertices of the shortest path between vertex $i$ and $j$, starting from vertex $j$ obtain the immediate predecessor one by one till vertex $i$.

The obtained shortest path is of course not unique in general.

## A.3 INITIAL ADJACENCY PAIR ALGORITHM

This algorithm calculates the initial adjacency pairs from a given allocation matrix (cf. Section 4.2 where initial adjacency pairs are defined and derived from the given allocation matrix).

1. Let $A_T = [a_{ij}]_{n\times n}$, $M = \max\{a_{ij}\}$ where $i = 1, ..., n$; $j = 1, ..., n$ and $n$ be the number of tiles. Initially $M = 10$, $j = 1$.

2. Consider the $j^{th}$ row.

3. If it corresponds to any tile which is covered in any of the obtained initial adjacency pairs we skip this row otherwise we obtain all the pairs of tiles corresponding to number $M$ in the allocation matrix and consider them as adjacency pairs.

4. If all the tiles are covered in the obtained adjacency pairs, terminate the algorithm; otherwise go to the next step.

5. Increase $j$ by one.
6. If $j < n+1$, go to step 2.
7. If $j = n+1$, reduce $i$ by one, consider $j = 1$ and go to step 2.

## A.4 Algorithm for the grouping of tiles

This algorithm obtains groups from the initial adjacency pairs. Here in particular, the algorithm is given for obtaining five groups which will be used to obtain a $T_S^P$. If the number of groups is greater or less than five, the initial adjacency pairs will require updating. Therefore this algorithm does not only obtain five groups, but also revises the initial adjacency pairs. Here are the steps of the algorithm:

1. Let the number of groups be $i$ and initially $i = 1$.

2. Obtaining the $1^{st}$ member of the $i^{th}$ group
   a. Consider each tile one by one from the given list of tiles.
   b. Select the tile which does not exist in any of the groups obtained so far.
   c. Now regard this tile as the $1^{st}$ member of the $i^{th}$ group.
   **Note:** *To start the process of forming groups, we consider the* $1^{st}$ *tile as the* $1^{st}$ *member of the* $1^{st}$ *group.*

3. Forming the $i^{th}$ group
   a. Among the adjacency pairs, we find those tiles which are adjacent to the $1^{st}$ member of the group.
   b. Then we include these tiles as members of the group.
   c. If newly included members are adjacent to other tiles from the initial adjacency pairs, we add those tiles to the group.
   d. We repeat Step 3.c until the remaining tiles from the initial adjacency pairs are adjacent to any other member of the group.
   e. When all members along with their adjacent tiles are included in the group we stop the process.

4. Review all the remaining tiles. If the number of tiles in the given list is equal to the number of all tiles included in the groups (i.e., if all the tiles are included in the groups)

a. Then proceed to step 5

b. Otherwise increase $i$ by one and go to step 2 to form another group.

5. To obtain a plus-shape tiling five groups are required, therefore

a. If $i = 5$, we stop.

b. If $i < 5$, we go to step 6 to increase the number of groups.

c. If $i > 5$, we go to step 7 to reduce the number of groups.

6. When the number of groups is less than 5 ( $i < 5$)

a. We search for the group having the maximum number of tiles as members.

b. If there is more than one group we consider the one which comes first.

c. Let this group be named $G$.

d. We look in the allocation matrix for a pair of elements of $G$ with minimum weight. If there is more than one pair we consider the one which comes first.

e. Let the tiles from this pair be $(R_i, R_j)$.

f. Now we update the initial adjacency pairs by deleting all those pairs which have any member in common with $G$.

g. Split $G$ into two parts, so that $i$ gets increased by one. Splitting has the effect of forming two new groups:

$(i)$. $G_1$ which contains $R_i$

$(ii)$. $G_2$ which contains $R_j$.

h. To find the members of $G_1$ and $G_2$, we look at the weight of each member of $G$ corresponding to $R_i$ and $R_j$.

i. If the obtained weight of any member corresponding to $R_i$ is greater than the weight corresponding to $R_j$ then this tile forms an adjacency pair with $R_i$ and we consider it as a member of $G_1$ otherwise it forms an adjacency pair with $R_j$ and we consider it as a member of $G_2$.

j. Repeat step 6.i until $G_1 \cup G_2 = G$.

k. Now $G$ is replaced by $G_1$ and $G_2$. Also, $i$ got increased by one.

l. Go to step 5.

7. When the number of groups is greater than 5 $(i > 5)$

a. From among the $i$ groups, we choose two having a minimum number of members.

b. Let these groups be named $G_1$ and $G_2$.

c. Combine $G_1$ and $G_2$ to form a new group.

d. We look in the allocation matrix for a pair of elements of $G_1$ and $G_2$ with maximum weight. If there is more than one pair we consider the one which comes first.

e. Consider this pair to be an adjacency pair.

f. Now $G_1$, $G_2$ together form a new group. Also, $i$ got reduced by one.

g. Go to step 5.

**Note:** *The steps 6.h, 7.d, 7.e, 7.f are meant to update the initial adjacency pairs. They have nothing to do with the formation of groups.*

## A.5 THE WIDTH, HEIGHT AND POSITION OF THE PLOT AND ITS PARTS

1. *Drawing Plot*: Let $(x, y)$ be the upper left corner of the central $T^R$. Draw a plot with width $L^p$ and height $H^p$ at position $(x-L^2-(L^p-L_B)/2, y-H^3-(H^p-H_A)/2)$. Because of choosing this position, the floor plan will be placed in the centre of the plot, i.e., the size of P1=P3, P2=P7, P4=P5 and P6=P8 (cf. Figure 7.1).

2. *Drawing P1*: Position: $(x-L^2-(L^p-L_B)/2, y-H^3-(H^p-H_A)/2)$, width: $L^2+(L^p-L_B)/2$, height: $H^3+(H^p-H_A)/2$.

3. *Drawing P2*: Position: $(x, y-H^3-(H^p-H_A)/2)$, width: $\max(L^1, L^3, L^5)$, height: $(H^p-H_A)/2$.

4. *Drawing P3*: Position: $(x+\max(L^1, L^3, L^5), y-H^3-(H^p-H_A)/2)$, width: $L^4+(L^p-L_B)/2$, height: $H^3+(H^p-H_A)/2$.

5. *Drawing P4*: Position: $(x+L^4+\max(L^1, L^3, L^5), y)$, width: $(L^p-L_B)/2$, height: $\max(H^1, H^2, H^4)$.

6. *Drawing P5*: Position: $(x-L^2-(L^p-L_B)/2, y)$, width: $(L^p-L_B)/2$, height: $\max(H^1, H^2, H^4)$.

7. *Drawing P6*: Position: $(x-L^2-(L^p-L_B)/2, y+\max(H^1, H^2, H^4))$, width: $L^2+(L^p-L_B)/2$, height: $H^5+(H^p-H_A)/2$.

8. *Drawing P7*: Position: $(x, y+H^5+\max(H^1, H^2, H^4))$, width: $\max(L^1, L^3, L^5)$, height: $(H^p-H_A)/2$.

9. *Drawing P8*: Position: $(x+\max(L^1, L^3, L^5), y+H^3+H^5+\max(H^1, H^2, H^4))$, width: $L^2+(L^p-L_B)/2)$, height: $H^5+(H^p-H_A)/2)$.

**Note:** For the upcoming sections, let $L_i$ and $H_i$ are the width and height of $T_S^R(i)$, $l_i$ and $h_i$ are width and height of its member $R_i$ where $i = 1, \ldots, n$.

## A.6 FUNCTION SHAPE1()

This function swaps the width and height of members of a group when *spiral*1, *spiral*2, *spiral*5 or *spiral*6 is used for the corresponding group.

1. When $R_2$ is going to be allocated above $R_1$

   1.1 Calculating the area of extra spaces
   If $\ell_1 > \ell_2$, then $A_O = (\ell_1 - \ell_2) \times h_2$ otherwise $A_O = (\ell_2 - \ell_1) \times h_1$
   If $\ell_1 > h_2$, then $A_I = (\ell_1 - h_2) \times \ell_2$ otherwise $A_I = (h_2 - \ell_1) \times h_1$.

   1.2 Interchanging the width and height (if required)
   If $A_O > A_I$, then swap $\ell_2$ and $h_2$, i.e.,
   temp $= \ell_2$, $\ell_2 = h_2$, $h_2 =$ temp.

   1.3 Calculating $L_2$ and $H_2$
   Initially $L_1 = l_1$, $H_1 = h_1$. Now $L_2 = \max(\ell_2, \ell_1)$, $H_2 = H_1 + h_2$.

2. When $R_i$ is going to be allocated to the left or right of $R_{i-1}$

   We are calculating the heights of $T_S^R$ only because either $H_i \geq h_i$ or $H_i < h_i$ but $L_i$ is simply $L_{i-1} + l_i$. Also, for further calculations, when $R_i$ is allocated to the left or right of $R_{i-1}$, only $H_i$ will be used.

   2.1 Calculating $H_i$
   $H_i = H_{i-2} + h_{i-1}$.

   2.2 Calculating the area of extra spaces
   If $H_i > h_i$, $A_O = (H_i - h_i) \times \ell_i$ otherwise $A_O = (h_i - H_i) \times L_{i-1}$
   If $H_i > \ell_i$, $A_I = (H_i - \ell_i) \times h_i$ otherwise $A_I = (\ell_i - H_i) \times L_{i-1}$.

   2.3 Interchanging the width and height (if required)
   If $A_O > A_I$, then swap $\ell_i$ and $h_i$.

   2.4 Updating $H_i$
   If $h_i > H_i$, then $H_i = h_i$.

3. When $R_i$ is going to be allocated above or below $R_{i-1}$

   3.1 Calculating $L_i$
   $L_i = L_{i-2} + \ell_{i-1}$.

   3.2 Calculating the area of extra spaces
   If $L_i > \ell_i$, $A_O = (L_i - \ell_i) \times h_i$ otherwise $A_O = (\ell_i - L_i) \times H_{i-1}$
   If $L_i > h_i$, $A_I = (L_i - h_i) \times \ell_i$ otherwise $A_I = (h_i - L_i) \times H_{i-1}$

3.3 Interchanging the width and height (if required)
If $A_O > A_I$, then swap $\ell_i$ and $h_i$.

3.4 Updating $L_i$
If $\ell_i > L_i$, then $L_i = \ell_i$.

4. Keep repeating steps 2 and 3 until all members of the corresponding group are allocated.

## A.7 FUNCTION L_AND_H1_1()

This function calculates the width and height of the first group when *spiral*1, *spiral*2, *spiral*5 or *spiral*6 is used for the corresponding group.

1. If the number of tiles in the $1^{st}$ group is greater than one

1.1 If the number of tiles in this group is an even number then
$L^1 = L_{n-1}$ and $H^1 = H_{n-1} + h_n$

In this case if *spiral*1, *spiral*2, *spiral*5 or *spiral*6 is used, then $R_n$ will be allocated above or below $T_S^R(n-1)$. Therefore after allocating $R_n$, the width of the group will be $L_{n-1}$ but for the height, $H_{n-1}$ will be augmented by $h_n$.

1.2 If the number of tiles in this group is an odd number then
$L^1 = L_{n-1} + l_n$ and $H^1 = H_{n-1}$

In this case if *spiral*1, *spiral*2, *spiral*5 or *spiral*6 is used, then $R_n$ will be allocated to the left or right side of $T_S^R(n-1)$. Therefore after allocating $R_n$, the height of the group will be $H_{n-1}$ but for the width, $L_{n-1}$ will be augmented by $l_n$.

2. If the number of tiles in the $1^{st}$ group is equal to one then $L^1$ and $H^1$ are $\ell_1$ and $h_1$ respectively.

### A.8 FUNCTION SHAPE 1_1()

This function computes the width or height of the inner extra space corresponding to each member of every group.

1. Calculating $L_2$ and $H_2$
   Initially $L_1 = l_1$, $H_1 = h_1$. Now $L_2 = \max(\ell_2, \ell_1)$, $H_2 = H_1 + h_2$.

2. Calculating the width of inner extra space after allocating $R_2$
   Width(inner extra space) $= |\ell_2 - \ell_1|$
   This value will be used while drawing the corresponding extra space.

3. When $R_i$ is going to be allocated to the left or right of $R_{i-1}$
   3.1 $H_i = H_{i-2} + h_{i-1}$.
   3.2 If $h_i > H_i$ then height(inner extra space)$= h_i - H_i$ otherwise we consider it as 0 (the explanation for considering the height of inner extra space only when $h_i > H_i$ is given in upcoming function).

4. When $R_i$ is going to be allocated above or below $R_{i-1}$
   4.1 $L_i = L_{i-2} + \ell_{i-1}$.
   4.2 If $\ell_i > L_i$, then width(inner extra space)$= \ell_i - L_i$ otherwise we consider it as 0.

5. Keep repeating steps 3 and 4 until all members of the corresponding group are covered.

### A.9 FUNCTION EXTRA 1_2($p_1, p_2, p_3, p_4$)

This function calculates the starting point of the second group and draw an outer extra space below the first or the second group.

1. Obtaining the starting point of the second group

Let $(x, y)$ is the upper left corner of the first group and initially the starting point of the second group.

When a member $R_i$ of the second group is drawn to the right side of $T_S^R(i-1)$, it overlaps with some members of the first group (e.g. if *spiral*2 is used for the second group, its second member is drawn at position $(x + l_1, y)$;

clearly this member overlaps with some members of the first group). Therefore we deduct the widths of all $R_i$, drawn to the right side of corresponding $T_S^R(i-1)$, from $x$ to obtain the starting point of the second group.

In function extra1_2$(p_1, p_2, p_3, p_4)$, $p_2$ represents the first value of $i$ for which $R_i$ is drawn to the right side of $T_S^R(i-1)$ and after $R_{p_2}$ every fourth member (if exist) is drawn to the right side of a $T_S^R$. Using $p_2$, we compute the widths of all $R_i$ drawn to the right side of corresponding $T_S^R(i-1)$ and subtract them from $x$.

Also $R_1$ of the second group overlaps with some members of the first group. Therefore corresponding $l_1$ is subtracted from $x$. For some spirals, $R_2$ is drawn to the left or to the right of $R_1$ (e.g. *spiral*2, cf. Figure 2.9). In this case, $p_1 = 0$ and $l_1$ is subtracted from $x$. For some spirals, $R_2$ is drawn above or below $R_1$ (e.g. *spiral*1, cf. Figure 2.9). In this case $p_1 = 1$ and $L_2$ is deducted from $x$.

For the second group, we require $(x, y)$ should be its upper right corner. When a member $R_i$ of the second group is drawn above $T_S^R(i-1)$, $(x, y)$ would not remain upper right corner of the second group. To obtain this position, the heights of all those $R_i$ which are drawn above corresponding $T_S^R(i-1)$, are added to $y$.

Here $p_3$ represents the first value of $i$ for which $R_i$ is drawn above $T_S^R(i-1)$ and after $R_{p_3}$ every fourth member (if exist) is drawn above some $T_S^R$.

If a member $R_j$ of the second group is drawn to the left or right of $T_S^R(j-1)$ and $l_j$ is greater than the width of $T_S^R(j-1)$, we draw an inner extra space to the right side of $T_S^R(j-1)$. This extra space is a virtual part of $T_S^R(j-1)$ and it virtually increases the width of $T_S^R(j-1)$. To obtain the starting point of the second group, the width of the inner extra space is subtracted from $x$. The width of inner extra spaces has already been obtained in the previous function.

Here $p_1$ also represents the first value of $i$ for which $R_i$ is drawn above or below $T_S^R(i-1)$ where $l_i$ is greater than the width of $T_S^R(i-1)$. For this case we have considered the upper and lower sides only, therefore every second member after $R_{p1}$ is drawn either above or below some $T_S^R$. Therefore using $p_1$ the width of all the inner extras are obtained and subtracted from $x$.

After all these calculations, obtained value of $x$ and $y$ gives the starting point of the second group.

REMARK. We have not considered the members $R_i$ whose height is greater than the height of $T_S^R(i-1)$ because inner extra space is always drawn below

a $T_S^R$. And to obtain the starting point, the heights of only those rectangles which are drawn above some $T_S^R$ are subtracted from $y$.

In case of the third group, we have not considered the cases when the width of members $R_i$ is greater than the width of $T_S^R(i-1)$ because in these cases, inner extra space is always drawn to the right of $T_S^R(i-1)$. And to get the starting point, the widths of only those $R_i$ which are drawn to the left of $T_S^R(i-1)$ are added to $x$.

Similarly for the first, fourth and fifth groups, any of the cases when the width or the height of $R_i$ is greater than the width (or the height) of $T_S^R(i-1)$, have not considered.

2. Drawing an outer extra space

If $H^1 > H^2$, we draw an outer extra space below the second group such that its lower left vertex is the upper left vertex of the extra space. The width and height of this extra space is $L^2$ and $H^1 - H^2$ respectively having position $(x - L^2, y + H^2)$.

If $H^1 < H^2$, we draw an outer extra space below the first group such that its lower left vertex is the upper left vertex of the extra space. The width and height of this extra space is $L^1$ and $H^2 - H^1$ respectively having position $(x, y + H^1)$.

REMARK. A particular colour is used for all the outer extra spaces to distinguish them from others spaces. Also after drawing an outer extra space, the number of outer extra spaces is increased by one so that in the end the total number of outer extra spaces is achieved.

## A.10 FUNCTION SHAPE1_2()

This function is used to draw all the members of any group when *spiral*1 is used for the group. Suppose $i^{th}$ group is going to be drawn and its starting point is $(x, y)$.

*1. Drawing* $R_1$

$R_1$ is drawn with the width and height $\ell_1$ and $h_1$ respectively at position $(x, y)$.

If the number of members of $i^{th}$ group is greater than one, then move to the next step otherwise stop here.

REMARK. After drawing each member, its name is printed at its centre and a particular colour is used for all the members of each group to distinguish them from the extra spaces.

*2. Drawing $R_2$ and the inner extra space*

$R_2$ is drawn with the width and height $\ell_2$ and $h_2$ respectively at position $(x, y - h_2)$.

If $\ell_2 < \ell_1$, then an inner extra space is drawn to the right side of $R_2$ with the width and height obtained in the function shape1_1() at position $(x + \ell_2, y - h_2)$.

If $\ell_2 > \ell_1$, then an inner extra space is drawn to the right side $R_1$, with the width and height obtained in the function shape1_1() at position $(x + \ell_1, y)$.

REMARK. A particular colour is used in all the inner extra spaces to distinguish them from other spaces. Also after drawing an inner extra space the number of inner extra spaces is increased by 1 so that in the last the total number of inner extra spaces will be achieved.

*3. Calculating $L_2$ and $H_2$*

*Initially* $L_1 = l_1$, $H_1 = h_1$. *Now* $L_2 = \max(\ell_2, \ell_1)$, $H_2 = H_1 + h_2$.

*4. Obtaining a position for $R_3$*

Since upper left vertex of $R_3$ should be upper right vertex of $T_S^R(2)$, we subtract $h_2$ from $y$, namely $y = y - h_2$, to obtain position of $R_3$.

*5. Drawing $R_i$ to the right side of $T_S^R(i-1)$*

Add $L_i$ to $x$ to obtain position of $R_i$. We draw $R_i$ with width $\ell_i$ and height $h_i$ at position $(x, y)$.

If $h_i < H_i$, we draw an inner extra space at position $(x, y + h_i)$ with width $\ell_i$ and height $H_i - h_i$. If $h_i > H_i$, we draw an inner extra space at position $(x - L_i, y + H_i)$ with width $L_i$ and height $h_i - H_i$.

In this case we update $H_i$ by $H_i = h_i$.

*6. Drawing $R_i$ below $T_S^R(i-1)$*

Subtract $L_i$ from $x$ and add $H_i$ to $y$ to obtain position of $R_i$. We draw $R_i$ with width $\ell_i$ and height $h_i$ at position $(x, y)$.

If $\ell_i < L_i$, we draw an inner extra space at position $(x + \ell_i, y)$ with width $L_i - \ell_i$ and height $h_i$.

If $\ell_i > L_i$, we draw an inner extra space at position $(x + L_i, y - H_i)$ with width $\ell_i - L_i$ and height $H_i$.

In this case we update $L_i$ by $L_i = \ell_i$.

*7. Drawing $R_i$ to the left side of $T_S^R(i-1)$*

Subtract $\ell_i$ from $x$ and subtract $H_i$ from $y$ to obtain position of $R_i$. We draw $R_i$ with width $\ell_i$ and height $h_i$ at position $(x, y)$.

If $h_i < H_i$, we draw an inner extra space at position $(x, y + h_i)$ with width $\ell_i$ and height $H_i - h_i$. If $h_i > H_i$, we draw an inner extra space at position $(x + \ell_i, y + H_i)$ with width $L_i$ and height $h_i - H_i$.

In this case we update $H_i$ by $H_i = h_i$.

*8. Drawing $R_i$ above $T_S^R(i-1)$*

Subtract $h_i$ from $y$ to obtain position of $R_i$. We draw $R_i$ with width $\ell_i$ and height $h_i$ at position $(x, y)$.

If $\ell_i < L_i$, we draw an inner extra space at position $(x + \ell_i, y)$ with width $L_i - \ell_i$ and height $h_i$. If $\ell_i > L_i$, we draw an inner extra space at position $(x + L_i, y + h_i)$ with width $\ell_i - L_i$ and height $H_i$.

In this case we update $L_i$ by $L_i = \ell_i$.

*9. Keep repeating the steps 5, 6, 7 and 8 until all the members are drawn.*

## A.11 FUNCTION ADJACENCYCAL1()

This function calculates the width (or the height) of $T_S^R$ after allocating each tile for each group when *spiral*1, *spiral*2, *spiral*5 or *spiral*6 is used for the corresponding group.

1. Initially $L_1 = l_1$, $H_1 = h_1$. Now $L_2 = \max(\ell_2, \ell_1)$, $H_2 = H_1 + h_2$.

2. When $R_i$ is going to be allocated to the left or right of $R_{i-1}$
   $H_i = H_{i-2} + h_{i-1}$. If $h_i > H_i$ we have $H_i = h_i$.

3. When $R_i$ is going to be allocated above or below $R_{i-1}$ member
   $L_i = L_{i-2} + \ell_{i-1}$. If $\ell_i > L_i$ we have $L_i = \ell_i$.

4. Keep repeating steps 2 and 3 until all the members of corresponding group are covered.

## A.12 FUNCTION ADJACENCY1()

This function calculates the adjacency pairs among each group.

*1. Obtaining adjacency of the first member with other members of the same group*

1.1. If $n = 2$ then $R_1$ will be adjacent to $R_2$.
1.2. If $n = 3$ then $R_1$ will be adjacent to $R_2$ and $R_3$.
1.3. If $n > 3$ then $R_1$ will be adjacent to $R_2$, $R_3$, $R_4$ and $R_5$.

*2. Obtaining adjacency with every next member*

Starting from $R_2$, each $R_i$ will be adjacent to every $R_{i+1}$ till $i < n$.

*3. Obtaining adjacency with every third next member*

Starting from $R_2$, each $R_i$ will be adjacent to every $R_{i+3}$ till $i < (n-2)$.

*4. Obtaining adjacency with every fourth next member*

Starting from $R_2$, each $R_i$ will be adjacent to every $R_{i+4}$ till $i < (n-3)$.

## A.13 FUNCTION ADJACENCYHEIGHT($p_3, p_4$)

This function calculates the members and their heights of the first and second group (resp. fourth group) which can be adjacent to each other.

There are four cases for the number of members of a group, namely $n = 1$, $n = 2$, $n = 3$ or $n > 3$. For $n > 3$, there are four sub-cases namely $n \equiv 1 \pmod 4$, $n \equiv 2 \pmod 4$, $n \equiv 3 \pmod 4$ and $n \equiv 0 \pmod 4$. We represent all these cases using $p_3$.

We know that at most three members of the first group can be adjacent to at most three members of the second or the fourth group (cf. Lemma 5.8). We represent these members by $r_1$, $r_2$ and $r_3$ and their heights by $s_1$, $s_2$ and $s_3$.

Here $p_4 = 1$ stands for the members (and their heights) of the first group which can be adjacent to the members of the second group (resp. fourth group) when *spiral*1, *spiral*2, *spiral*3 or *spiral*4 (resp. *spiral*5, *spiral*6, *spiral*7 or *spiral*8) is used for the first group.

$p_4 = 2$ represents the members (and their heights) of the first group which can be adjacent to the members of the second group (resp. fourth group)

when *spiral*5, *spiral*6, *spiral*7 or *spiral*8 (resp. *spiral*1, *spiral*2, *spiral*3 or *spiral*4) is used for the first group.

$p_4 = 3$ corresponds to the members (and their heights) of the second group (resp. fourth group) which can be adjacent to the members of the first group when *spiral*5, *spiral*6, *spiral*7 or *spiral*8 (resp. *spiral*1, *spiral*2, *spiral*3 or *spiral*4) is used for the second group (resp. fourth group).

$p_4 = 4$ symbolizes the members (and their heights) of the second group (resp. fourth group) which can be adjacent to the members of the first group when *spiral*1, *spiral*2, *spiral*3 or *spiral*4 (resp. *spiral*5, *spiral*6, *spiral*7 or *spiral*8) is used for the second group (resp. fourth group).

Let $e_1$, $e_2$ and $e_3$ are the members of the first group and $g_1$, $g_2$ and $g_3$ represent their heights respectively. If three members of a group (which can be adjacent to other members of any other group) are drawn one above the other such that $e_3$ at top, $e_2$ in middle and $e_1$ at bottom. Let $f_1$, $f_2$ and $f_3$ represents members of the second or the fourth group and $h_1$, $h_2$ and $h_3$ represents their height respectively. Similarly if $f_1$, $f_2$ and $f_3$ are drawn one above the other then we have $f_3$ at top, $f_2$ in middle and $f_1$ at bottom. For example, in Figure 4.5 for the left $T_S^R$, $R_5$, $R_8$, $R_3$ are $f_3$, $f_2$, $f_1$ respectively.

1. Set $p_1 = p_3 - 1$.
2. If $p_1 = 1$ (here $n = 1$)

In this case, only one member of the first group can be adjacent to one member of the second or the fourth group.

Here $r_1$ is $R_1$ and $s_1 = h_1$. If $p_4 = 1$ or $2$, we have $e_1 = r_1$, $g_1 = s_1$. If $p_4 = 3$ or $4$, we have $f_1 = r_1$, $h_1 = s_1$.

3. If $p_1 = 2$

3.1 If $n = 1$, this step is same as Step 2.

3.2 If $n = 2$ we have $r_2$ is $R_1$ with $s_2 = h_1$, $r_1$ is $R_2$ with $s_2 = h_1$.

If $p_4 = 1$ we have $e_1 = r_1$, $e_2 = r_2$, $g_1 = s_1$, $g_2 = s_2$.

If $p_4 = 2$ we have $e_1 = r_2$, $e_2 = r_1$, $g_1 = s_2$, $g_2 = s_1$.

If $p_4 = 3$ we have $f_1 = r_1$, $f_2 = r_2$, $h_1 = s_1$, $h_2 = s_2$.

If $p_4 = 4$ we have $f_1 = r_2$, $f_2 = r_1$, $h_1 = s_2$, $h_2 = s_1$.

4. If $p_1 = 3$

4.1 If $n = 1$, this step is same as Step 2.

4.2 If $n = 2$ we have $r_1$ as the first member, $s_1 = H_1$. The cases for $p_4 = i$, $i = 1, \ldots, 4$ are same as in Step 2.

4.3 If $n = 3$ we have $r_2$ as the first member, $s_2 = H_1$, $r_1$ as the third member, $s_2 = h_3$. The cases for $p_4 = i$, $i = 1, \ldots, 4$ are same as in Step 3.2.

5. If $n > p_1$

5.1 If $n \equiv p_3 \pmod 4$ (($n$ congruent to $p_3$ modulo 4)) then $r_1$ is $R_n$, $s_1 = H_n$. The cases for $p_4 = i$, $i = 1, \ldots, 4$ are same as in Step 2.

5.2 If $n \equiv p_3 + 1 \pmod 4$ then $r_2$ is $R_n$, $s_2 = h_n$, $r_1$ is $R_{n-1}$, $s_1 = H_{n-1}$. The cases for $p_4 = i$, $i = 1, \ldots, 4$ are same as in Step 3.2.

5.3 If $n \equiv p_3 + 2 \pmod 4$ then $r_2$ is $R_{n-1}$, $s_2 = h_{n-1}$, $r_1$ is $R_{n-2}$, $s_1 = H_{n-2}$. The cases for $p_4 = i$, $i = 1, \ldots, 4$ are same as given in Step 3.2.

5.4 If $n \equiv p_3 + 3 \pmod 4$ then $r_3$ is $R_{n-2}$, $s_3 = h_{n-2}$, $r_2$ is $R_{n-3}$, $s_2 = H_{n-3}$ and $r_1$ is $R_n$, $s_1 = h_n$.

If $p_4 = 1$ we have $e_1 = r_1, e_2 = r_2, e_3 = r_3, g_1 = s_1, g_2 = s_2, g_3 = s_3$.

If $p_4 = 2$ we have $e_1 = r_3, e_2 = r_2, e_3 = r_1, g_1 = s_3, g_2 = s_2, g_3 = s_1$.

If $p_4 = 3$ we have $f_1 = r_1, f_2 = r_2, f_3 = r_3, h_1 = s_1, h_2 = s_2, h_3 = s_3$.

If $p_4 = 4$ we have $f_1 = r_3, f_2 = r_2, f_3 = r_1, h_1 = s_3, h_2 = s_2, h_3 = s_1$.

6. If $p_4 = 1$ or $p_4 = 2$ we have $e_4 = n$. If $p_4 = 3$ or $p_4 = 4$ we have $f_4 = n$. The values of $e_4$ and $f_4$ will be used in the upcoming functions.

EXAMPLE A.6. Refer to Figure 4.5 where the members of the first group can be adjacent to the members of the second group, we have $p_4 = 2$ and $p_3 = 4$. From Step 5.2, we have $r_1 = R_2$, $r_2 = R_{15}$, hence $e_1 = R_{15}$, $e_2 = R_2$.

Before moving to the function findingadjacencies($p_1, p_2$), we talk about function adjacency($p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9$) which is going to be used in the function findingadjacencies($p_1, p_2$).

At most three members of the first group can be adjacent to at most three members of another group, therefore there are nine possibilities to be considered for computing adjacency among the members of different groups. All these nine possibilities are given in the function adjacency($p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9$).

### A.14 FUNCTION ADJACENCY($p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9$)

If $p_1 = 1$, $p_2 = 1$, $p_3 = 1$, we consider $e_1$ is adjacent to $f_1$, $f_2$, $f_3$ respectively.

If $p_4 = 1$, $p_5 = 1$, $p_6 = 1$, we consider $e_2$ is adjacent to $f_1$, $f_2$, $f_3$ respectively.

If $p_7 = 1$, $p_8 = 1$, $p_9 = 1$, we consider $e_3$ is adjacent to $f_1$, $f_2$, $f_3$ respectively.

As said before 1, 2 or 3 members of the first group can be adjacent to 1, 2 or 3 members of any other group. These possibilities are expressed by following functions:

*adjacentrects*11(), *adjacentrects*21(), *adjacentrects*31(), *adjacentrects*12(), *adjacentrects*22(), *adjacentrects*32(), *adjacentrects*13(), *adjacentrects*23(), *adjacentrects*33().

For example function adjacentrects31() represents that only one member of the first group can be adjacent to at most three members of any other group. For these functions, adjacency pairs are obtained by comparing the width or height of the members of corresponding groups.

Here it is not possible to go through all these nine functions, therefore we consider only one of them. For an illustration, we discuss the steps of function adjacentrects22().

### A.15 FUNCTION ADJACENTRECTS22()

1. If $(h_2 + h_1) \leq g_2$ then $f_1$, $f_2$ is adjacent to $e_2$ and we call function adjacency(0, 0, 0, 1, 1, 0, 0, 0, 0) to obtain adjacency pairs $(f_1, e_2)$ and $(f_2, e_2)$.

2. If $h_2 < g_2$ & $h_1 > (g_2 - h_2)$ then $f_2$ is adjacent to $e_2$ and $f_1$ is adjacent to $e_1$, $e_2$. Here we call function adjacency(1, 0, 0, 1, 1, 0, 0, 0, 0) to obtain corresponding adjacency pairs.

3. If $h_2 > g_2$ & $h_2 < (g_2 + g_1)$ then $f_2$ is adjacent to $e_1$, $e_2$ and $f_1$ is adjacent to $e_1$. Therefore, we call function adjacency(1, 1, 0, 0, 1, 0, 0, 0, 0).

4. If $h_2 \geq (g_2 + g_1)$ then $f_2$ is adjacent to $e_1$, $e_2$ and we call function adjacency(0, 1, 0, 0, 1, 0, 0, 0, 0).

5. If $h_2 = g_2$ then $f_2$ is adjacent to $e_2$ and $f_1$ is adjacent to $e_1$. Here we call function adjacency(1, 0, 0, 0, 1, 0, 0, 0, 0).

Now we discuss function findingadjacency($p_1, p_2$).

From functions adjacencyLength($p_3, p_4$) or adjacencyHeight($p_3, p_4$), there are four cases for the values of $f_4$ and $e_4$, namely $f_4 > i$ and $e_4 > j$ where $i = 0, \ldots, 3$, $j = 0, \ldots, 3$. For obtaining adjacency pairs, it is required to consider $e_4$ and $f_4$ together. Therefore, in total there are sixteen possibilities.

In function findingadjacency($p_1, p_2$), $p_1 = 2$ and $p_2 = 1$ represents the case $f_4 > 2$, $e_4 > 1$. These sixteen possibilities are divided into following four parts.

In the first part we consider $f_4 \leq p_1$ and $e_4 \leq p_2$ where the number of sub-cases is $p_1 \times p_2$. For example for $f_4 \leq 2$ and $e_4 \leq 1$, we consider the sub-cases $f_4 = 1$ & $e_4 = 1$, $f_4 = 2$ & $e_4 = 1$.

In the second part we consider $f_4 > p_1$ and $e_4 \leq p_2$ where the number of sub-cases is $4 \times p_2$. For example for $f_4 > 2$ and $e_4 \leq 1$, we consider the sub-cases $f_4 \equiv 1 \pmod 4$ & $e_4 = 1$, $f_4 \equiv 2 \pmod 4$ & $e_4 = 1$, $f_4 \equiv 3 \pmod 4$ & $e_4 = 1$, $f_4 \equiv 0 \pmod 4$ & $e_4 = 1$. In general, for this part we call function *adjacentrects*1($p_3, p_1$). For example for $f_4 > 2$ and $e_4 \leq 1$, we have $p_3 = 1$, $p_1 = 2$.

In the third part we consider $f_4 \leq p_1$ and $e_4 > p_2$ where the number of sub-cases is $p_1 \times 4$. In general, for this part we call function *adjacentrects*2($p_3, p_1$). For example for $f_4 \leq 2$ and $e_4 > 1$, we have $p_3 = 2$, $p_1 = 1$.

In the fourth part, we consider $f_4 > p_1$ and $e_4 > p_2$ where the number of sub-cases is $4 \times 4 = 16$.

It is not possible to go through all the sixteen cases, therefore for demonstration we discuss only one case.

Suppose the first and second group is drawn using *spiral*1. This is the case $f_2 > 2$ and $e_4 > 0$ which implies that $p_1 = 2$ and $p_2 = 0$.

## A.16 FUNCTION FINDINGADJACENCY(2, 0)

For $p_1 = 2$ & $p_2 = 0$, we first consider the case $f_4 \leq 2$ & $e_4 > 0$ and then consider the case $f_4 > 2$ & $e_4 > 0$.

1. $f_4 \leq 2$ & $e_4 > 0$

If $p_1 = 2$ & $p_2 = 0$ we call function adjacentrects2(2, 0). For this particular example, the function adjacentrects2(2, 0) has the following steps:

1.1 If $f_4 = 1$ & $e_4 \equiv 1 \pmod 4$ we call function adjacentrects11(). As an example, function adjacentrects22() has already been discussed (cf. Section A.15).

1.2 If $f_4 = 1$ & ($e_4 \equiv 2 \pmod 4$ or $e_4 \equiv 3 \pmod 4$) we call function adjacentrects21().

1.3 If $f_4 = 1$ & $e_4 \equiv 0 \pmod 4$ we call function adjacentrects31().

1.4 If $f_4 = 2$ & $e_4 \equiv 1 \pmod 4$ we call function adjacentrects12().

1.5 If $f_4 = 2$ & ($e_4 \equiv 2 \pmod 4$ or $e_4 \equiv 3 \pmod 4$) we call function adjacentrects22().

1.6 If $f_4 = 2$ & $e_4 \equiv 0 \pmod 4$ we call function adjacentrects32().

2. $f_4 > 2$ & $e_4 > 0$

2.1 If ($f_4 \equiv 0 \pmod 4$ or $f_4 \equiv 1 \pmod 4$) & $e_4 \equiv 1 \pmod 4$ we call function adjacentrects12().

2.2 If ($f_4 \equiv 0 \pmod 4$ or $f_4 \equiv 1 \pmod 4$) & ($e_4 \equiv 2 \pmod 4$ or $e_4 \equiv 3 \pmod 4$), we call function adjacentrects22().

2.3 If ($f_4 \equiv 0 \pmod 4$ or $f_4 \equiv 1 \pmod 4$) & $e_4 \equiv 0 \pmod 4$ we call function adjacentrects32().

2.4 If $f_4 \equiv 2 \pmod 4$ & $e_4 \equiv 1 \pmod 4$ we call function adjacentrects13().

2.5 If $f_4 \equiv 2 \pmod 4$ & ($e_4 \equiv 2 \pmod 4$ or $e_4 \equiv 3 \pmod 4$) we call function adjacentrects23().

2.6 If $f_4 \equiv 2 \pmod 4$ & $e_4 \equiv 0 \pmod 4$ we call function adjacentrects33().

2.7 If $f_4 \equiv 3 \pmod 4$ & $e_4 \equiv 1 \pmod 4$ we call function adjacentrects11().

2.8 If $f_4 \equiv 3 \pmod 4$ & ($e_4 \equiv 2 \pmod 4$ or $e_4 \equiv 3 \pmod 4$) we call function adjacentrects21().

2.9 If $f_4 \equiv 3 \pmod 4$ & $e_4 \equiv 0 \pmod 4$ we call function adjacentrects31().