BURNING A GRAPH AS A MODEL FOR THE SPREAD OF SOCIAL CONTAGION

by

Elham Roshanbin

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
April 2016

To my parents

Table of Contents

List of Figures

Abstract

The spread of social contagion is an active area in social network analysis. Assume that we want to spread a message among all the users of a network. Knowing the structure of the network, we may ask how fast can we do this and what is the best strategy? Graph burning is a new graph process which is used as a model for the spread of social contagion. Burning number is a graph parameter associated with the burning process and measures the speed of contagion in the underlying graph of a social network.

In the thesis, we provide several results on the burning number. In particular, we study the relationship of the burning number to other graph structural parameters, the burning number of some specific graphs, the computational complexity of this problem, and the probabilistic versions of this parameter. We also consider the competitive diffusion game on graphs that was our first motivation to define the graph burning process, and we discuss the existence of pure Nash-equilibrium for this game on some specific graph families.

List of Abbreviations and Symbols Used

$b(G)$	the burning number of graph G
$b_i(G)$	the i-burning number of graph G
$\gamma_k(G)$	the k-distance domination number of graph G
$V(G)$	the node set of graph G
$E(G)$	the edge set of graph G
$G[S]$	the induced subgraph by S
$d_G(u,v)$	the graph distance between the two nodes u and v in a graph G
$d_E(u,v)$	the Euclidean distance between the two points u and v in the Cartesian plane
$N_r^G[u]$	the r-closed neighbourhood of node u in graph G
$N(u)$	the open neighbourhood of node u
rad(G)	the radius of graph G
diam(G)	the diameter of graph G
$\Delta(G)$	the maximum degree of graph G
$\overline{G}$	the complement of graph G
P_n	the path of order n
C_n	the cycle of order n
Q_n	the n-hypercube graph of order 2^n
K_n	the complete graph of order n
$K_{m,n}$	the complete bipartite graph with parts of orders m and n

$SP(s,r)$ .... the spider graph with s arms, each arm of length r

$G \square H$ .... the Cartesian product of graphs G and H

$G \boxtimes H$ .... the strong product of graphs G and H

$G \cdot H$ .... the lexicographic product of graphs G and H

$G \mathbin{\dot\cup} H$ .... the disjoint union of graphs G and H

ILT .... the Iterated Local Transitivity model

$[n]$ .... the set of numbers $\{1, 2, \ldots, n\}$

O_m .... the set of numbers $\{1, 3, \ldots, 2m - 1\}$

MPF_k^t .. the set of maximal path-forests with t components and with burning number k

PS_k^t .... the set of perfect spider graphs with t arms and with radius k

$k - BMS^t$ .... the set of k-burning maximal spiders graphs with t arms

$\mathscr{G}(n,r)$. the set of random geometric graphs of order n and with parameter r defined on the unit square

$wt(u)$ .... the weight of the node u in a weighted graph

$f = o(g)$ .... asymptotic notation

$f = O(g)$ .... asymptotic notation

$f = \omega(g)$ .... asymptotic notation

$f = \Omega(g)$ .... asymptotic notation

$f = \Theta(g)$ .... asymptotic notation

$f = (1 + o(1))g$ .... asymptotic notation

Acknowledgements

I would like to express my sincere thanks to my supervisors Anthony Bonato and Jeannette Janssen for their support and guidance in writing this thesis, and for all the great academic lessons that I have learned from them. It was a great opportunity for me to work with Anthony who defined graph burning and the burning number, and he invited me to join him in working on this. It was a great honor for me to work with Jeannette with her endless enthusiasm for mathematics and her great ideas on the math problems that we solved together.

I also would like to thank my committee members Stephen Finbow, Gena Hahn, and Richard Nowakowski for reading this thesis, and their great suggestions and comments. I thank my collaborators during my PhD, Pawel Prałat and Dieter Mitsche whom I met while visiting Ryerson University.

I am grateful to all of the people in the department of mathematics and statistics at Dalhousie University for all of the nice experiences that I have had with them. My special thanks to our graduate coordinators Sara Faridi and David Iron for their support throughout my PhD. I thank Jason Brown, Richard Nowakowski, Pawel Prałat, and Edward Susko for the wonderful courses that I took with them, and for the great scientific advice that I received from them. My thanks to Dorette Pronk for the great academic advice that I received from her.

It was a blessing for me to come to Halifax to make wonderful new friends that always will stay in my heart with the great memories that we have shared. There are no words to explain the mentorship, friendship, love, and support that I experienced from Aloma Graham. I am grateful to my friends Ben Cameron, Huda Chuangpishit, Darien DeWolf, Abbas Mehrabian, Lucas Mol, Rory Wilson, Danielle Cox, and many others whom I met at Dalhousie for their support.

Finally, I would like to thank my parents, my sister, and my brothers; without their unconditional love and support I could not have been here.

Chapter 1

Introduction

Assume that we have a message or a piece of information (such as a new idea, or a rumour) that we want to distribute among all the users in a network such as Facebook or Twitter. Our goal is to accomplish this task in the fastest possible way with the minimum cost. That is, we want to send the message to only a few of the users, that we refer to as the *target users*. The hope is that, through the interactions or links between the individuals in the network, the message will reach to everyone in the network in the shortest possible time.

At the end of the process, every individual in the network is to receive the message either directly or indirectly from a source of information. At the same time, we may think that every user that is already influenced by the message, sends it to all or a subset of its neighbours. By its neighbours, we mean those individuals that have direct interaction with that user; for example, our neighbours in the Facebook are our friends who can see the shared posts directly from our timeline. We can think of many different possible ways for doing so, and there are several recent studies that model these types of processes; see [6, 13, 41].

Usually in studies of models of the spread of influence in social networks, every user or individual in the network is either *active*, when it is influenced by the information (or it has adopted the idea), or is *inactive*, when it has not received the influence from any other user. In these models we use graphs as the abstract mathematical objects to show the structure of a network. Simply put, we construct a graph in which every node is corresponding to an individual in the network, and two nodes are adjacent if and only if there is an interaction or link between the corresponding users of the network.

As a new simple deterministic approach for modelling these sort of real world phenomenon, we define a new graph process called *graph burning*. Graph burning is inspired by graph theoretic processes like Firefighting [7, 19, 27], graph cleaning [3], and graph bootstrap percolation [5]. In a graph burning process, we have a simple finite graph G (representing a network), and we want to burn its nodes in the fastest possible way. We start the burning process by burning one of the nodes in the first round. Then in each subsequent round, we burn one new unburned node if such a node is available, and at the same time the fire spreads from the set of burning nodes (from earlier stages) to their unburned neighbours. Throughout the process, each node is either *burned* or *unburned*; if a node is burned, then it remains in that state until the end of the process. The process ends when every node is burned. Our goal is to minimize the number of steps needed for burning G.

The *burning number* of a graph G, denoted by $b(G)$, is the minimum number of rounds needed for a burning process of G to end. For example, it is straightforward to see that $b(K_n) = 2$. However, even for a relatively simple graph such as the path P_n on n nodes, computing the burning number is more complex; in fact, $b(P_n) = \lceil n^{1/2} \rceil$ as stated in Theorem 19. Burning may be viewed as a simplified model for the spread of social contagion in a social network such as Facebook or Twitter. The lower the value of $b(G)$, the easier it is to spread such contagion in the graph G. Below are the known problems that are related to the burning problem, and that are our motivation for defining this new parameter.

A well-known example of a problem for modelling the propagation processes in social networks is the *r-neighbour bootstrap percolation* introduced in 1979 by Chalupa, Leath and Reich [21], and is defined as follows. Suppose that we have a graph G and A is a subset of nodes in G. We start the process (at time $t = 0$) by infecting the nodes in A, and setting $A_0 = A$. Then in each round $t \geq 1$, a new node becomes infected if and only if it has at least r neighbours in A_{t-1}, where A_{t-1} is the set of the nodes that are infected at the end of round $t - 1$. If a node is infected at round $t \geq 0$, then it remains infected until the process ends. The process ends, when we cannot infect a new node by the above rules. We say set A *percolates*, if the process terminates by infecting the whole graph G. The r-neighbour bootstrap percolation has been widely studied by both mathematicians and physicists due to its interesting applications; see [1, 14].

One of the most extensively studied graph theory parameters since its introduction in the 1950's (see [11, 48]) is the *domination number*, which naturally arises in problems involving social networks, and computer social networks; see [11, 29, 43]. Assume that we have a graph G and we want to find a set of nodes, say D, such that every node in $V(G) \setminus D$ is adjacent to a node in D. If G is representing a social network, then it means that the users corresponding to the nodes in D are able to spread an influence to any other user out of this set. If D is a small set, then it can be a good candidate for being a target set in a rumor propagation process, or a convenient set for placing agents that can serve the rest of the network in an emergency situation. Such a set of nodes in G is called a *dominating set*, and we call the size of a minimum dominating set in graph G as the *domination number* of G, denoted by $\gamma(G)$. There are many different variations of the dominating set problem, such as the *r-distance domination* and the *broadcast domination* which are defined below.

In r-distance domination, we want to find a set of nodes like D_r in a graph G such that every node in $V(G) \setminus D_r$ is within distance r from a node in D_r. A simple interpretation of this problem could be efficiently placing some utilities such as fire stations in a city, so as to be within r units of any building in the city. Another example could be building radio transmitters in the cities of a province or a country; see [34, 35, 56]. Imagine that every radio transmitter has a restricted power and people can hear it while they are within distance r from a city that has a transmitter. Since building transmitters is costly, we want to minimize the total number of transmitters needed.

However, this model does not include the case where each transmitter has a different power. In such as case, the broadcast domination, introduced in 2002 by Erwin [25], would be the appropriate problem for modelling the situation. In broadcast domination, we have a graph G and a function $f : V(G) \to \mathbb{Z}^+$ that describes the power of a transmitter located at each node. The set of nodes $D_f = \{v \in V(G) : f(v) \geq 1\}$ is called the set of *dominators*, and any node in $G \setminus D_f$ is called *receiver*. In fact, for any node $v \in D_f$, $f(v)$ is the power of the transmitter located at v. We say that the function f is a *broadcast domination* for G if for any node $u \in V(G)$ there is a transmitter v in D_f for which $d(u,v) \leq f(v)$. The goal here is finding a broadcast domination with minimum *cost*; that is, $\sum_{v \in V(G)} f(v)$.

The (k,r)-*centre problem* is a facility location problem or clustering problem that is similar to both r-domination and broadcast domination (see [8]), and has similarities to the burning problem as well. Assume that for a graph G, we want to find a set of nodes $S \subseteq V(G)$ of size at most k for which every node in G is within distance r from a node in S. If r is fixed, then it is the r-domination problem. If r is not fixed and k is fixed, then it is known as the *k-centre problem* [32], where the goal is to obtain the minimum r for which we have a r-dominating set of size at most k. The (k, r)-centre problem was introduced by Barilan *et al.* [8] in 1993, and since then has been widely studied due to its applications in facility location and clustering. For example, suppose that we are tasked with building k fire stations (or any sort of utilities or services) in a city, knowing that each fire station would work efficiently for those houses within distance at most r blocks away from the station. The question is: Which set of buildings in the city are the best candidates for having the minimum cost in setting fire stations under the above constraint?

The *Firefighter problem* is a deterministic discrete time process on graphs that was first introduced by Hartnel in 1995 [33] for modelling the spread of fire (or social contagion). At the beginning of this process, a fire is initiated on a node of a graph. Then at each round of the process, the firefighter will defend one unburned (unprotected) node against the fire. After the firefighter protects a node, the node is called *defended* or *protected*, and can not be burned any more throughout the process. At the end of each round, the fire spreads from the burning nodes to their unburned, unprotected neighbours. Once a node is burned, it stays in this status until the end of the process. The process terminates when the fire can not be spread to any unburned node. An objective of the Firefighter problem is to determine the maximum number of nodes that can be saved from burning by firefighter. Another approach would be finding the fastest way for stopping the fire; see the survey [27]. There have been many studies on the Firefighter problem, considering the problem from different perspectives as it has beautiful applications in viral marketing, spread of virus in computers, and spread of disease (see [26, 27]).

In all problems we mentioned above, except for the Firefighter, we consider a single idea or influence that must dominate the entire body or a large portion of a network. For instance, in the domination problem, the goal is just to influence the nodes out of the dominating set by the same idea. However, many real examples of social contagion are more complex. These may involve more than one agent, all of whom are in competition with each other; see [13, 24, 41]. There are many studies considering these kind of social interactions, where we have a set of parties or agents outside of the network with the objective to influence a large number of the users in the network by their own idea using some representatives inside the network; see for example, the *Voronoi games* [24, 44]. One of these models that motivates the burning problem (as we will see in Chapter 5) is the *competitive diffusion* model that was introduced by Alon *et al.* in 2009 [2]. They defined a game on graphs modelling competition among different agents inside of a network.

In this game, we have a set of players (corresponding to the agents) each with its exclusive colour (product being advertised) that tries to colour (influence) as many as possible number of nodes (individuals) in a graph (network) with its own colour. Initially, each player chooses one node and colour the node with its own colour (as her strategy). If two or more players choose the same node, then the node becomes gray, and those players have to leave the game. Hence, we may assume that this is illegal. Then in each subsequent round, any uncoloured node that has coloured neighbours with only one colour gets the same colours from them. If an uncoloured node has two or more neighbours with different colours, then it becomes gray, and is deleted from the graph. The game continues till we can not colour any more node by the above rules. The gain or payoff of each player is the number of the nodes that the player achieves at the end of the game with its own colour.

Finally, a well-studied non-deterministic rumour spreading model for the social networks is the *push and pull protocol* which was introduced by Demers *et al.* in [22]. Suppose that one node in a network is aware of a piece of information or a message, and wants to spread it to all nodes quickly. The push and pull protocol is a randomized graph process for this goal and proceeds in discrete time-steps or rounds as follows. In each round, every informed node contacts a random neighbour and influences it by the message; that is, it *pushes* the message. At the same time, every uninfluenced node contacts a random neighbour and receives the message if the neighbour knows it; that is, it *pulls* the message; also see [40].

1.1 Preliminaries

In this section we briefly explain some mathematical concepts and notation that we use in the thesis. If A is a finite set, then the *cardinality* or *size* of A is the number of the elements in A, and is denoted by $|A|$. For example, the cardinality of set $A = \{a, b, 3\}$ is 3, and we can show this simply by $|A| = 3$. We usually denote the set of numbers $\{1, 2, \dots, n\}$ by $[n]$. The *Cartesian* product of two sets such as A and B, denoted by $A \times B$, is defined as below.

$$A \times B = \{(x, y) : x \in A, y \in B\}.$$

We sometimes use the following asymptotic notation that is used frequently in mathematics to describe the asymptotic behaviour of a complicated function in terms of a simplified function. Suppose that f and g are two functions defined on the set of positive integers with the same domain.

We say that f is *asymptotic* to g, and we write $f \sim g$, if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1.$$

We say that function f is *big oh* of g, and we denote it by $f = O(g)$, if there exists a positive constant c and a positive integer n_0, such that for every $n \geq n_0$, we have that

$$f(n) \leq cg(n).$$

We say that f is *omega* of g, and we write $f = \Omega(g)$, if there exists a positive constant a and a positive integer n_0, such that for every $n \geq n_0$, we have that

$$f(n) \geq ag(n),$$

or equivalently, $g = O(f)$.

We say that f is *theta* of g, and denote it by $f = \Theta(g)$, if we can find positive constants a and c, and a positive integer n_0, such that for every $n \geq n_0$, we have that

$$ag(n) \leq f(n) \leq cg(n),$$

or equivalently, $f = O(g)$ and $f = \Omega(g)$.

If for all large enough values of n, $f(n)$ is negligible compared to $g(n)$, then we say that f is *little oh* of g, and we write $f = o(g)$. More precisely, if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0,$$

or even we sometimes write $f \ll g$. In particular, $f = o(1)$ implies that $\lim_{n\to\infty} f(n) = 0$. Hence, by definition of the little oh notation, we can say that $f \sim g$ if and only if $f = (1 + o(1))g$.

If $f(n)$ grows faster than $g(n)$, then we say that f is *little omega* of g, and we write $f = \omega(g)$. More precisely, if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty,$$

or we sometimes write $f \gg g$.

1.2 Graph Theory

Graphs model the relations between elements in a set. In this section, we provide some preliminaries that we need in the thesis from graph theory. We refer the readers for more information on graph theory to [58].

A graph $G = (V(G), E(G))$ consists of a non-empty set of *nodes* or *vertices* denoted by $V(G)$ or simply V, and a set of *edges* denoted by $E(G)$ or simply E. Each edge $e \in E(G)$ consists of a pair of nodes in $V(G)$ that are called the the *end points* of e and they can be equal. If the end points of an edge e are equal, then the edge e is called a *loop* in G. For convenience, we sometimes denote an edge e with end points u and v by uv. Also, we may have different edges such as e_1 and e_2, with exactly the same end points. In this case, we call e_1 and e_2 as *parallel* or *multiple* edges in G. We may use $G = (V, E)$ for representing graph G whenever it does not cause a confusion. Throughout the thesis we show generally a graph $G = (V(G), E(G))$ simply just by G assuming that the reader knows that graph G consists of a vertex set $V(G)$ and an edge set $E(G)$; unless we need to define G by a specific set of nodes and edges.

Every edge $e = uv$ in a graph G represents a connection between the nodes u and

v in G. In this case, we say that u and v are *joined*, *adjacent*, or *neighbours*. We can represent any finite graph $G = (V, E)$ with a drawing or diagram in the plane, in which every node is represented by a distinct point or tiny circle in the plane. We then connect two different points by an arbitrary line in the plane, if and only if the corresponding nodes are adjacent in G. For example, Figure 1.1 demonstrates the diagram of graph $G = (V(G), E(G))$, with $V(G) = \{a, b, c\}$ and $E(G) = \{ab, ac\}$ in the plane.

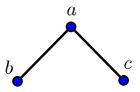


Figure 1.1: A drawing of the graph G.

A graph G that does not contain any loop or multiple edges is called a *simple* graph. We only work with simple graphs throughout this thesis. The cardinality of the node set $V(G)$ is called the *order* of graph G, and we call the number of the edges in G the *size* of G. All the graphs discussed in this thesis are *finite*; that is, graphs with finite orders and sizes.

Assume that e is an edge in a simple graph G. The *contraction of edge e* is an operation on G in which we delete edge e from G, and simultaneously we merge the end points of e. We also delete any possible parallel edges or loops created by this operation. We denote the resulting graph by $G \cdot e$.

A graph $H = (V(H), E(H))$ is a *subgraph* of a graph G, and is denoted by $H \subseteq G$, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A subgraph H of the graph G is an *induced* subgraph of G, if for every edge $e \in E(G)$ for which the end points of e are in $V(H)$, we have that $e \in E(H)$. If $S \subseteq V(G)$, then the *subgraph induced by S*, denoted by $G[S]$, is the subgraph of G with $V(G[S]) = S$, and $E(G[S])$ is the set of all edges e in G whose end points are in S. A subgraph H is a *spanning* subgraph of G if $V(H) = V(G)$. If $S \subseteq V(G)$, then $G - S$ is the subgraph induced by $V(G) \setminus S$. When $S = \{x\}$, then we simply write $G - x$, instead of $G - \{x\}$.

A sequence of nodes and edges such as $v_1, e_1, v_2, e_2, \ldots, v_{k-1}, e_{k-1}, v_k$, in which $e_i = v_i v_{i+1}$, for $1 \leq i \leq k - 1$, and v_i's are all distinct nodes of G, is called a *path* in graph G. The nodes v_1 and v_k are called the *end points* of the path. We can simply show the path only by the nodes that appear in it; that is, by $v_1, v_2, \ldots, v_k$, since every edge is determined by its end points uniquely (in a simple graph). We usually denote a path on n nodes by P_n. The *length* of a path P is the number of edges in P. A *cycle* in a graph G is a path in which the end points are equal, and all other nodes are distinct. We denote a cycle on n nodes by C_n. Note that in a cycle the number of nodes and edges are equal. The *length* of a cycle is the number of edges in the cycle. A cycle whose length is an even integer is called an *even cycle*, and a cycle with odd number of edges is called an *odd cycle*. A cycle of length three is sometimes called a *triangle*.

The *graph distance* or *geodesic distance* between two nodes u and v in a graph G is the length of a shortest path in G with end points u and v, and is denoted by $d_G(u,v)$ or simply by $d(u,v)$ whenever this does not cause confusion. It is a known fact in graph theory that, the distance defined above is a metric on $V(G) \times V(G)$. That is, for each pair of nodes such as u and v in $V(G)$, we have that,

1. $d(u,v) \geq 0$, and $d(u,v) = 0$ if and only if $u = v$.

2. $d(u,v) = d(v,u)$.

3. $d(u,v) \leq d(u,w) + d(w,v)$, for any node $w \in V(G)$.

For a nonnegative integer r, the *r-th open neighbourhood* of a node u in graph G, denoted by $N_r^G(u)$, or simply by $N_r(u)$, is defined as the set $\{v \in V(G) : 0 < d(u,v) \leq r\}$. If $r = 0$, then $N_r(u) = \varnothing$, and if $r = 1$, then we denote $N_1(u)$ by $N(u)$, which is the set of the neighbours of u in G. The *r-th closed neighbourhood* of a node u, denote by $N_r^G[u]$ or simply $N_r[u]$, is the set $\{v \in V(G) : d(u,v) \leq r\}$. If $r = 0$, then $N_0[u] = \{u\}$, and if $r = 1$, then we denote $N_1[u]$ by $N[u]$. We note that $N_r[u] = N_r(u) \cup \{u\}$.

A graph G is called *connected*, if for every pair of distinct nodes u and v in G there is a path with end points u and v. A graph G that is not connected, is called a *disconnected* graph. A *component* of a graph G is a maximal subgraph of G that is connected. If G is connected, then G by itself is the only component of G. If G is disconnected, then it must consist of at least two distinct components. A connected graph without any cycles is called a *tree*. A tree is typically denoted by T. It is known that trees are connected graphs with n nodes and $n-1$ edges. In fact, every minimal connected graph is a tree. This implies that every connected graph contains at least one spanning tree. It is also known that a graph T is a tree if and only if every pair of the nodes in T are connected by a unique path. Every subgraph of a graph G that is a tree is called sometimes a *subtree* of G. A disconnected graph in which every component is a tree is called a *forest*. In other words, a forest is a disconnected graph without any cycle, or an *acyclic* disconnected graph. If each component of a forest F is a path, then we call F a *path-forest*.

A graph G is *planar* if there is a drawing of G in the plane such that no two edges of G cross each other except at the end-points. Assume that C is a cycle in a graph G of length at least 4. A *chord* in C is an edge that connects two non-successive nodes in C. We call the graph G a *chordal graph* if the induced subgraph on the node set of any cycle of length at least 4 in G has at least one chord. In other words, any cycle in a chordal graph is triangulated.

The degree of the node u in a graph G is the number of neighbours of u in G, and is denoted by $\deg_G(u)$, or $d_G(u)$. We usually use the notation $d(u)$, when we talk about a specific graph and it does not cause a confusion. An *isolated node* is a node of degree zero, that is a node that does have any neighbour. A *universal node* is a node that is adjacent to every other node in graph G. The *maximum degree* of graph G is the largest degree that a node in G has, and is denoted by $\Delta(G)$. The *minimum degree* of graph G is the smallest degree among the set of degrees of the nodes in G, and we denote it by $\delta(G)$. If a graph G contains an isolated node, then $\delta(G) = 0$, and if G contains a universal node and is of order n, then $\Delta(G) = n - 1$. A node u with $\deg(u) = 1$ is called a *pendant* node or a *leaf*.

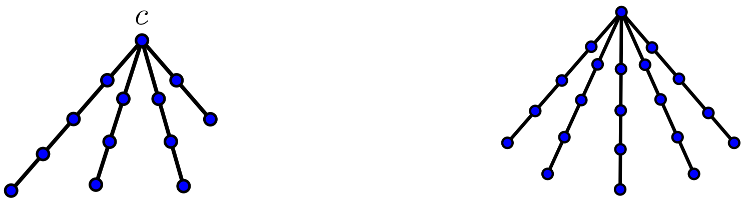
The *eccentricity* of a node u in graph G is defined as $\max\{d(v, u) : v \in V(G)\}$. The *radius* of G is the minimum eccentricity over the set of all nodes in G, and is denoted by $\mathrm{rad}(G)$. The *centre* of G consists of the nodes in G with minimum eccentricity. That is, a node u is in the centre of G if and only if u attains $\mathrm{rad}(G)$ as its eccentricity. Each node u that is in the centre of G is called a *central node* of G. The *diameter* of G, denoted by $\mathrm{diam}(G)$, is the maximum eccentricity over the node set of G. In other words, $\mathrm{diam}(G) = \max\{d(u, v) : u, v \in v(G)\}$.

A *complete graph* of order n is a graph in which every pair of nodes are connected by an edge, and is denoted by K_n. The *complement* of a graph G, denoted by $\overline{G}$ or sometimes by G^c, is a graph with $V(\overline{G}) = V(G)$ in which two nodes u and v are adjacent if and only if u and v are not adjacent in G. In other words, the edge set of

$\overline{G}$ is the complement of the edge set of G with respect to the edge set of the complete graph on the node set of G. For example, we can easily see that $\overline{K_n}$ is a graph with n nodes and no edges.

A set of nodes S in a graph G is called an *independent set* of G, if no two nodes in S are adjacent. In other words, $G[S]$ does not have any edge. A graph G is called a *bipartite graph* if the node set of G can be partitioned into two subsets X and Y such that X and Y are each an independent set in G. Then (X,Y) is called a *bipartition* for G, and we sometimes denote G by $G[X,Y]$. Note that G does not have necessarily a unique bipartition, for example, when G is bipartite and it also contains an isolated node. It is known that a graph is bipartite if and only if it does not contain an odd cycle. A *complete bipartite* graph is a bipartite graph for which there is a bipartition (X,Y) such that every node in X is adjacent to each node in Y. We denote a complete bipartite graph $G[X,Y]$, in which $|X| = n$ and $|Y| = m$ by $K_{n,m}$. A *star* is a complete bipartite graph in which one of the parts consists of a single node. In other words, a graph G is a star if for some positive integer n, $G = K_{1,n}$. That is, G is a star if it has a universal node u, and $G - u$ forms an independent set. In a star $K_{1,n}$ with $n \geq 2$, the central node is the only node of degree more than 1 and every other node is a leaf.

We call a tree that has only one node c of degree at least three a *spider graph*, and the node c is called the *spider head*. In a spider graph every leaf is connected to the spider head by a path which is called an *arm*. Figure 1.2a shows an example of a spider graph.



(a) A spider graph with spider head c.

(b) A spider graph $SP(5,4)$.

Figure 1.2: Spider graphs.

If all the arms of a spider graph with maximum degree s are of the same length r, then we denote it by $SP(s,r)$. The spider graph shown in Figure 1.2b is an example of $SP(5,4)$.

A *dominating set* in a graph G is a set of nodes D for which every node in $G - D$ is adjacent to at least one node in D. The cardinality of a minimum dominating set in G is called the *domination number* of G and is denoted by $\gamma(G)$.

An *isomorphism* between a graph G and a graph H is a bijection $f : V(G) \longrightarrow V(H)$ such that for any two nodes $u, v \in V(G)$, u and v are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. In such a case, we say that the graph G is *isomorphic* to the graph H.

There are various ways for creating a new graph by combining a set of two or more graphs. Suppose that $G = (V(G), E(G))$ and $H = (V(H), E(H))$ are two distinct graphs with $V(G) \cap V(H) = \varnothing$. The *disjoint union* of G and H, denoted by $G \uplus H$, is a graph with node set $V(G) \cup V(H)$, and the edge set $E(G) \cup E(H)$. The *Cartesian product* of graphs G and H, denoted by $G \square H$, is a graph with node set $V(G) \times V(H)$, in which two nodes (u_1, v_1) and (u_2, v_2) are adjacent if and only if, either $u_1 = u_2$ and $v_1 v_2 \in E(H)$, or $u_1 u_2 \in E(G)$ and $v_1 = v_2$.

The *strong product* of graphs G and H, denoted by $G \boxtimes H$, is a graph with node set $V(G) \times V(H)$, in which two nodes (u_1, v_1) and (u_2, v_2) are adjacent if and only if $v_1 v_2 \in E(H)$, or $u_1 u_2 \in E(G)$. By definition, we can see that $G \square H \subseteq G \boxtimes H$. The *lexicographic product* of graphs G and H, denoted by $G \circ H$, is a graph with node set $V(G) \times V(H)$, in which two nodes (u_1, v_1) and (u_2, v_2) are adjacent if and only if either $u_1 u_2 \in E(G)$, or $u_1 = u_2$ and $v_1 v_2 \in E(H)$. Intuitively, if $V(G) = \{u_1, u_2, \ldots, u_n\}$, then $G \circ H$ is isomorphic to the graph that is constructed by replacing each node u_i in G by a copy of H, called H_i, and then adding all the edges uv, where $u \in V(H_i)$, $v \in V(H_j)$, and $u_i u_j$ is an edge in G. For any two nodes (u_i, v_j) and (u_l, v_s) in $G \circ H$, we have the following possibilities: If $u_i \neq u_l$, then

$$d_{G \circ H}\left((u_i, v_j), (u_l, v_s)\right) = d_G(u_i, u_l);$$

If $u_i = u_l$ and $v_j \neq v_s$, then

$$d_{G \circ H}\left((u_i, v_j), (u_l, v_s)\right) = \min\{2, d_H(v_j, v_s)\}.$$

For more on graph products see [37].

1.3 Complexity Theory

In this section, we give a brief review of some concepts from computational complexity theory. *Computational complexity theory* is a branch of the theory of computing that considers computational problems and their classifications according to their difficulty. A *computational problem* is a problem that can be solved by a computer, or equivalently, by an algorithm in finite number of steps. All problems we refer to in this chapter are computational problems. Here, indeed the word "problem" applies to an infinite collection of instances. For example, finding a minimum dominating set for graphs is a problem, and an *instance* of it would be finding a minimum dominating set for a specific graph (among infinitely many graphs that we could imagine). By an *algorithm*, we mean a detailed step by step procedure designed for solving a problem. Each algorithm, designed for solving a problem, starts from an initial state called its *input*, and after performing a finite number of steps it terminates providing us with an *output*. This output represents a solution to the problem.

There are different types of computational problems such as optimization problems and decision problems. In an *optimization* problem, we try to minimize or maximize a parameter over a set of feasible solutions for a problem. For example, finding the minimum order of a dominating set in a graph is an optimization problem. A *decision problem* is a problem whose answer (or output) is either Yes or No. Many mathematical problems, including graph theory problems can be written as a decision problem. For example, finding a minimum dominating set for a graph is a computational optimization problem whose decision version would be: For some positive integer k, is there a dominating set in a graph of order at most k or not? It is a computational problem, as we can write a computer code that takes a graph G as its input (for instance by encoding its adjacency matrix or its adjacency list), and then consider all of the k-subsets of nodes in G to check if they form a dominating set for G or not.

Algorithms can be time-consuming and when run on computers they also require space considerations. Hence, for solving a problem we wish to design *efficient* algorithms; that is, algorithms that consume a reasonable amount of time and space. Here, we only consider the time-complexity of the problems. Suppose that we are given a specific problem and an algorithm for solving that problem. Fix an encoding scheme for all the instances of this problem. The *length of input* or *input length* is a positive integer-valued function on the set of all possible instances of the problem that measures the size of the input for every instance of this problem, and is defined in Section 4.2.1 of the book [30]. Usually in graph theory problems, the length of the input is the number of nodes in a given graph.

Assume that we have a specific problem and an algorithm designed for solving that problem. Also, suppose that we fix the input length n. The *time-complexity* or *complexity* or *running time* of the algorithm is the number of steps for solving the worst case input of the problem which is described as a function of n. In this definition, we ignore the actual time for performing each step of an algorithm, as we take it as the unit of time. For example, given an algorithm for solving the dominating set problem, and a fixed positive integer n, a worst case input is a graph of order n that requires the maximum number of steps for the algorithm to find a minimum dominating set for the graph.

The time-complexity or the complexity of a problem is the complexity of the best algorithm over the set of all possible algorithms designed for solving the problem. Accordingly, decision problems are divided into some known classes such as **P**, **NP**, and **NP**-complete. We provide a brief overview of these concepts that are commonly used in the literature, and refer the reader to [30] for more formal and detailed definitions.

We say that a decision problem is *solvable in polynomial time* if its complexity is bounded above by a polynomial in terms of the length of input. The set of all decision problems that are solvable in polynomial time is called **P**. Unfortunately, as we see later on, there is no polynomial time algorithm known for the dominating set problem (see [30, 34]). There are however many graph theory problems that are known to be in **P**. For example, determining whether a graph is Eulerian or not is a decision problem that is solvable in polynomial time. Namely, we can write a computer code that takes a graph as input, and checks the degrees of its nodes. If the degrees are all even, then the answer is Yes; otherwise, the answer is No. Similarly, an algorithm is called a *polynomial time* algorithm, if the number of steps it takes to get the output is bounded by a polynomial in terms of the length of the input. Hence, equivalently a problem is in **P**, if there is a polynomial time algorithm that solves all the instances of the problem.

As we prefer to design efficient algorithms, determining whether or not a problem is solvable in polynomial time is an important question. For some problems it is not known whether a polynomial time algorithm exists or not, and it is highly possible that there is no such algorithm. However, we can usually check the truth of a candidate solution for an instance of these problems in polynomial time. We say a decision problem is in class **NP**, if for any instance of the problem for which the answer is Yes and any *certificate* (or a candidate solution) for the answer, the truth of the certificate can be checked in polynomial time (in terms of the length of the input). For example, the dominating set problem is in **NP**. Given any graph G and a subset D of order at most k, we can check in polynomial time if every node in $V(G) \setminus D$ has a neighbour in D or not.

We have that $\mathbf{P} \subseteq \mathbf{NP}$, as given any certificate for an instance of a problem in **P**, we can just ignore the certificate, and solve the problem in polynomial time. Hence, intuitively the problems in **NP** sound harder that the problems in **P**, and it is a well-known conjecture in computational complexity theory that $\mathbf{P} \neq \mathbf{NP}$; that is, $\mathbf{P} \subsetneq \mathbf{NP}$, see [30]. However, there are some problems in **NP** that look to be more difficult than all other problems in **NP**, and they are **NP**-complete problems. Before going through the definition, we first define the concept of polynomial reduction.

A *polynomial reduction*, or simply a *reduction*, from a decision problem A to a decision problem B is a polynomial time algorithm that transforms every instance I of the problem A to an instance J of the problem B, such that I is a Yes instance of A if and only if J is a Yes instance of B. If such a reduction from A to B does exist, then we say that the problem A is *polynomially reducible* (or for short, *reducible*) to the problem B, and we denote it by $A \leq B$. In such a case, if there is a polynomial time algorithm for B, then it can be transformed to a polynomial time algorithm for solving problem A. Thus, if $A \leq B$, and problem B is in **P**, then so is A. By contraposition, we conclude that, if $A \leq B$ and A is not in **P**, then B is not in **P** either. Intuitively, it means that problem B is at least as hard as problem A. Note that the relation " $\leq$ " defined on the set of decision problems, is reflexive and transitive.

A problem $A \in \mathbf{NP}$ is called **NP**-*complete* if for each problem B in **NP** we have that $B \leq A$. That is, all the problems in **NP** are reducible to A. The set of all

NP-complete problems is denoted by **NPC**. By definition, $\mathbf{NPC} \subseteq \mathbf{NP}$. Intuitively, a problem in **NPC** is as hard as any other problem in **NP**. Thus, a significant advantage of defining reductions is to show the **NP**-completeness of a new problem, by reducing from a problem that is already known to be **NP**-complete. More precisely, suppose that we know that a problem A is **NP**-complete, and for a new problem B, we show that $B \in \mathbf{NP}$, and also $A \leq B$. Since the relation " $\leq$ " is transitive, and for all problems such as $C \in \mathbf{NP}$, $C \leq A$, then we conclude that $C \leq B$, for each $C \in \mathbf{NP}$. Therefore, it follows that B is also **NP**-complete.

Stephen Arthur Cook in 1971 and Leonid Levin in 1973 [30] independently showed the existence of **NP**-complete problems by proving the **NP**-completeness of the Boolean Satisfiability Problem or the SAT problem. Since then there have been many problems that have been proved to be **NP**-complete, such as the dominating set problem. The 3-*SAT problem* (a special case of the SAT problem) is one of the most famous problems that is known to be **NP**-complete, and has been widely used to show the **NP**-completeness of other problems, using the reductions. We need to provide some required concepts before stating the 3-SAT problem.

A *Boolean variable* is a variable that can take value either True or False. Assume that $U = \{u_1, u_2, \ldots, u_k\}$ is a finite set of Boolean variables. A 3-element set $C = \{x_1, x_2, x_3\}$ in which each x_i is an element of U such as u_i or its negation (denoted by $\overline{u}_i$), and we do not have both u_i and $\overline{u}_i$ in the same set C, is called a *clause* over U. By a *truth assignment* for the variables in U, we mean a function $f : U \to \{\mathrm{True}, \mathrm{False}\}$, that assigns to each variable in U a *truth value*; that is, a value chosen from the set $\{\mathrm{True}, \mathrm{False}\}$. Here is the statement of the 3-SAT problem.

Problem: 3-SAT
Instance: A finite set $U = \{u_1, u_2, \ldots, u_n\}$ of Boolean variables and a collection $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ of 3-element clauses over U.
Question: Is there any truth assignment for the variables in U such that every clause in $\mathcal{C}$ contains at least one True assigned variable?

The decision version of the dominating set problem that we state below, was first proven by Johnson [34] to be **NP**-complete, by a reduction from the 3-SAT problem. Below we bring the formal statement of this problem and its **NP**-completeness proof, as an example that demonstrates the above concepts and definitions. Although, today we can find other proofs with different reductions for this theorem; see [30].

Problem: Dominating Set Problem

Instance: A graph $G = (V(G), E(G))$ of order n and a positive integer $k \leq n$.

Question: Is there any dominating set such as D for G with $|D| \leq k$?

Theorem 1 ([34]). *The dominating set problem is **NP**-complete.*

Proof. To show the **NP**-completeness of the dominating set problem, we need to do two things. First, we have to show that it is in **NP**. This was discussed earlier as an example of a problem in **NP**. Second, we must show that there is a reduction from a problem in **NPC** to the dominating set problem. We will show that there is a polynomial time reduction from the 3-SAT problem to the dominating set problem as follows.

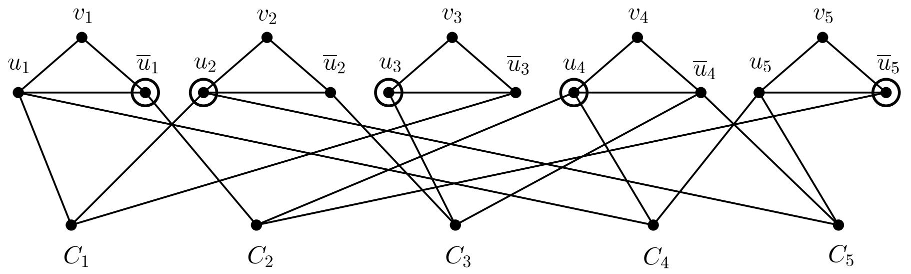


Figure 1.3: The graph $G(\mathcal{C})$.

Suppose that we are given an instance of the 3-SAT problem; that is, a finite set $U = \{u_1, u_2, \dots, u_n\}$ of Boolean variables and a collection $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$ of 3-element clauses over U. We then construct a graph called $G(\mathcal{C})$ as follows. Corresponding to each variable $u_i \in U$, we make a 3-cycle with nodes $u_i, \overline{u}_i$, and v_i. For each clause $C_i = \{u_{i_1}, u_{i_2}, u_{i_3}\} \in \mathcal{C}$, we add a new node, called C_i, and then we join the node C_i to the nodes u_{i_1}, u_{i_2}, and u_{i_3}. Therefore, we have a graph $G(\mathcal{C})$, consisting of $3n + m$ nodes, that is a polynomial of the length of the input. For example, assume that $U = \{u_1, u_2, u_3, u_4, u_5\}$, and $\mathcal{C} = \{C_1, C_2, C_3, C_4, C_5\}$, where $C_1 = \{u_1, u_2, \overline{u}_3\}$, $C_2 = \{\overline{u}_1, u_4, \overline{u}_5\}$, $C_3 = \{\overline{u}_2, u_3, \overline{u}_4\}$, $C_4 = \{u_1, u_4, u_5\}$, and $C_5 = \{u_2, \overline{u}_4, u_5\}$. Then the graph $G(\mathcal{C})$ that we defined above is shown in Figure 1.3.

We want to show that U and $\mathcal{C}$ is a Yes instance of the 3-SAT problem if and only if $G(\mathcal{C})$ has a dominating sat of order at most n (that is, $G(\mathcal{C})$ is a Yes instance of the dominating set problem for $k = n$).

First, suppose that there is a truth assignment for the variables in U such that every $C_i \in \mathcal{C}$ contains at least one True-assigned variable. We then define a set $S \subseteq V(G(\mathcal{C}))$ as follows. For $1 \leq i \leq n$, if u_i has value True, then we add the node u_i to S; otherwise, we add $\overline{u}_i$ to S. Thus, S is set of order n in $G(\mathcal{C})$, such that for each triangle of $G(\mathcal{C})$ such as $u_i, \overline{u}_i, v_i$, either u_i or $\overline{u}_i$ is in S. Therefore, set S dominates all the u_i's, $\overline{u}_i$'s, and v_i's in $G(\mathcal{C})$. It remains to show that S dominates the nodes C_i's as well. However, by assumption each clause C_i contains at least one true-assigned value variable, that is by definition corresponding to a node in S. Thus, S is a dominating set of order n for $G(\mathcal{C})$. For example, in the specific instance that we mentioned above, $f : U \rightarrow \{\text{True}, \text{False}\}$, with $f(u_1) = \text{False}$, $f(u_2) = \text{True}$, $f(u_3) = \text{True}$, $f(u_4) = \text{True}$, and $f(u_5) = \text{False}$, is a satisfying truth assignment. Also, $S = \{\overline{u}_1, u_2, u_3, u_4, \overline{u}_5\}$ is a dominating set for the graph $G(\mathcal{C})$. We show the nodes in S with a circles around them in Figure 1.3.

Now, suppose that $G(\mathcal{C})$ has a dominating set S of order at most n. Since every node v_i, for $1 \leq i \leq n$ must be dominated by at least one node in S, then it implies that $|S| \geq n$. Hence, we have that $|S| = n$, and there must be at least one node from each triangle in S. Therefore, we do not have any of the nodes C_i's in S, and each C_i must have at least one neighbour in S. We define a truth-assignment $f : U \rightarrow \{\text{True}, \text{False}\}$ as follows. For $1 \leq i \leq n$, if $u_i \in S$, then we define $f(u_i) = \text{True}$; otherwise, we define $f(u_i) = \text{False}$. Since every node C_i must have a neighbour in S, and we assigned a value of True to any variable corresponding to a node in S, it then implies that each clause C_i contains at least one True-assigned variable. Thus, U and $\mathcal{C}$ is a Yes instance of the 3-SAT problem.

The argument above shows a polynomial reduction from the 3-SAT problem to the dominating set problem, and thus, the proof follows. □

For some problems, for example the Subset-Sum problem (as stated below), the input contains a list of numerical parameters. In such a case, we can have a positive integer-valued function on the set of all instances of the problem that measures the maximum numeric parameter in the given input list for every instance of the problem, and is defined in Section 4.2.1 of the book [30]. We call this function the *maximum* of input. A *number problem* is a problem in which the input contains a list of numerical parameters such that the maximum of input is not bounded polynomially above by the length of input. For example, the following problem which is called the *Subset-Sum* problem is a number problem.

Problem: Subset-Sum
Instance: A finite multi-set $X = \{a_1, a_2, \ldots, a_n\}$ of positive integers, and a positive integer t.
Question: Is there any subset $S \subseteq X$ such that the numbers in S add up to t?

The Subset-Sum problem is an example of a number problem, since the numbers in a given set X can be exponentially large in the length of input. The number problems that are **NP**-complete are divided into two classes: the weakly **NP**-complete problems and the strongly **NP**-complete problems. A number problem that is **NP**-complete, even when restricted to the numbers that are bounded by a polynomial in terms of the length of the input, is called a *strongly* **NP**-complete problem, or equivalently, **NP**-complete problem in the strong sense; Otherwise, it is called *weakly* **NP**-complete. For example, it is known that the Subset-Sum problem is weakly **NP**-complete, while the 3-Partition problem that is stated below is strongly **NP**-complete.

Problem: 3-Partition
Instance: A finite multi-set $X = \{a_1, a_2, \ldots, a_{3n}\}$ of positive integers, and a positive integer B where $\sum_{i=1}^{3n} a_i = nB$, and $B/4 < a_i < B/2$, for $1 \leq i \leq 3n$.
Question: Is there any partition of X into n triples such that in each triple the elements add up to B?

Usually a graph theory problem that is **NP**-complete is automatically strongly **NP**-complete, since there is only one parameter in the input which is often bounded by the number of the nodes in the given graph that is the length of the input; see [30].

1.4 Chapter Overview

Many of the results in this thesis appeared in [16, 17, 18, 46, 51]. The thesis is organized as follows. In Chapter 2, we first define graph burning and the burning number of a graph, and we investigate the burning number as the minimum of i-burning numbers. We then consider the fundamental facts about the burning number such as bounds on the burning number of disconnected graphs, and the equivalence of graph burning and covering problems. We also give a characterization for the burning number of a graph in terms of its subtrees and its spanning subtrees, where the graph is connected. We provide a characterization for the graphs with burning number 2, and we find the burning number of some specific graphs such as paths, cycles, and complete bipartite graphs. Moreover, we consider the monotonicity of the burning number on the specific isometric subgraphs of a graph, such as isometric subtrees. We then provide some tight bounds on the burning number in terms of radius and diameter. We show a strong connection between the burning number and the distance domination, that we use it for finding bounds on the burning number, as well as proving Nordhaus-Gaddum type-results on the burning number of a graph and its complement. We present a bound on the burning number of a graph in terms of its order, and we suggest a conjecture on the best possible upper bound for the burning number of a graph in terms of its order. We determine the burning number in the Iterated Local Transitive model for social networks. We finish Chapter 2 by considering the burning number of graph products. We find asymptotic results on the burning number of Cartesian grids, toroidal grids, strong grids, and hypercube graphs. We state a conjecture on the burning number of hypercube graphs. We present bounds on the burning number of the Cartesian, strong, and lexicographic product of graphs.

In Chapter 3, we prove that the Graph Burning problem is **NP**-complete even for acyclic graphs with maximum degree three, and we find the burning number of perfect binary trees. Then we state a conjecture for finding a lower bound on the burning number of a tree, and we prove the truth of this conjecture in some specific cases, for instance, for spider graphs $SP(s,r)$ with $s \geq r$. As a result, we find the exact value of the burning number of spider graphs $SP(s,r)$ with $s \geq r$. The Graph Burning problem is shown to be **NP**-complete for spider graphs. We also prove that Graph Burning is **NP**-complete for path-forests. We finish Chapter 3 by presenting two algorithms that find the burning number of path-forest and spider graphs with bounded parameters.

In Chapter 4, we first consider the burning number of the random geometric graphs. Using Poisson random processes, we then introduce a random variation of graph burning, called Poisson random burning. We consider this random process for the graphs in general, and for paths in particular.

In Chapter 5, we consider the competitive diffusion game on graphs that was our first motivation for defining graph burning. We first consider the two-player case of this game on some specific graphs such as paths, cycles, trees, unicyclic graphs, and Cartesian grids. Then we present some results on the k-player case of this game, where $k \geq 3$.

Finally, we provide a conclusion on the thesis, and we finish the thesis by suggesting some further directions for the future work on graph burning and the burning number of graphs, and the competitive diffusion game on graphs.

Chapter 2

Graph Burning

In this chapter, we explore the core properties of the burning number, ranging from characterizations, bounds, to computing the burning number on certain kinds of graphs. We first state the formal definition of the burning process and the burning number of a graph.

Graph burning is a discrete-time graph process that is defined on the node set of a simple finite graph G as follows. There are discrete time-steps or rounds. Throughout the process, each node is either *burned* or *unburned*. Initially at time $t = 0$ all the nodes are unburned. Then in each round $t \geq 1$, we choose one unburned node to burn, if such a node is available. Once a node is burned in round t, in round $t + 1$, each of its unburned neighbours becomes burned. If a node is burned, then it remains in that state until the end of the process. The process ends when all nodes are burned. The *burning number* of a graph G, written by $b(G)$, is the minimum number of rounds needed for the burning process to end.

For example, $b(K_n) = 2$, for $n \geq 2$. To see this, first note that for each graph with at least two nodes, we need at least two burning steps, since we can not burn two distinct nodes at the first step. Thus, for $n \geq 2$, we have that $b(K_n) \geq 2$. On the other hand, since each node in K_n is a universal node, in any burning process for K_n all nodes will receive the fire in the second step. Therefore, two steps for burning K_n is efficient; that is, $b(K_n) = 2$, when $n \geq 2$. Note that our argument here implies that K_1 is the only graph that requires only one step to be burned, and for a graph G with at least two nodes, we have that $b(G) \geq 2$.

Suppose that in the process of burning a graph G, we burn the whole graph G in k steps. For each i, $1 \leq i \leq k$, we denote the node that we choose to burn directly in the i-th step by x_i. We call such a node a *source of fire*. The sequence $(x_1, x_2, \ldots, x_k)$ is called a *burning sequence* for G. With this notation, the burning number of G is the length of a shortest burning sequence for G; such a burning sequence is referred to as *optimum*. For example, for the path P_4 with node representation v_1, v_2, v_3, v_4 the sequence (v_2, v_4) is an optimum burning sequence; see Figure 2.1 (the open circles represent the fire sources).

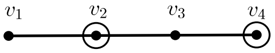


Figure 2.1: Burning the path P_4.

As we mention in Theorem 23 and we can see in Lemma 53 in Chapter 3, the burning number of a spider graph $SP(r, r)$ equals $r + 1$; that is, radius plus one. Moreover, an optimum burning sequence for $SP(r, r)$ could be started by choosing the spider head, that in this case is the centre of the graph, as the first fire source. In such a case, choosing the rest of the fire sources is not a concern, as it does not make the speed of burning faster. In general, for any graph G with $b(G) = \text{rad}(G) + 1$, by burning a central node of G every node in G will be burned after $\text{rad}(G) + 1$ steps, and therefore, we do not need to care for the best possible choices for the rest of the fire sources. That is, for $i \geq 2$, we can choose the i-th fire source to be any unburned node in stage $i-1$, and still the graph G will be burned optimally in $\text{rad}(G)+1$ steps.

Also, as we will see in Conjecture 41 for the hypercube graphs, we think that the burning number of the hypercube graph Q_n equals $k = \lceil \frac{n}{2} \rceil + 1$, and is achieved by a burning sequence $(x_1, x_2, \ldots, x_k)$ in which $d(x_1, x_2) = n$. In other words, the selection of x_i, for $i \geq 2$ is not a concern.

In general, assume that for a fixed positive integer i, we define a new graph process for G similar to the graph burning process, but we only change one of the rules: We only burn i fire sources in the first i steps of the process, and we let the fire to be spread by the same rules to the rest of the unburned nodes. We call this new process an *i-graph burning*, or simply an *i-burning* process. We define $b_i(G)$ to be the minimum number of steps that we need to burn the whole graph G in an i-burning process for G. If in an i-burning process for G we denote the node that we choose to burn in the j-th step by x_j for $1 \leq j \leq i$, then $(x_1, x_2, \ldots, x_i)$ is called an *i-burning sequence* for G. An i-burning sequence for G is referred to be *optimum* if it corresponds to an optimum i-burning process for G.

With this notation, we can easily see that $b_1(G) = \text{rad}(G) + 1$, since by burning a central node in G everything in G will be burned after $\mathrm{rad}(G)$ more steps, and by definition, $\mathrm{rad}(G)$ is the minimum eccentricity over the node set of G. Moreover, we can see that $b_i(G) \geq b_{i+1}(G)$ for $i \geq 1$. Namely, by definition, for any optimum $(i+1)$-burning sequence for G such as $(x_1, x_2, \ldots, x_{i+1})$, the sequence $(x_1, x_2, \ldots, x_i)$ forms an i-burning sequence for G. Consequently, we have the following lemma.

Lemma 2. *For any graph G we have that*

$$b(G) = \inf_{i \geq 1} b_i(G).$$

As we prove in Chapter 3, finding the burning number of a graph G is a difficult problem. Hence, finding the minimum number i for which $b(G) = b_i(G)$ is difficult too. Thus, in this thesis, we only consider the burning number of a graph G rather than considering $b_i(G)$'s.

2.1 Properties of the Burning Number

In this section, we present some basic facts and bounds for the burning number, as well as some useful techniques for finding these bounds. We characterize the graphs with burning number 2, present an alternative way for defining the burning process in terms of subtrees of a graph, give a general characterization for the burning number in terms of the burning number of trees, find the burning number of paths and cycles, and we show the relation between the distance domination numbers of a graph and its burning number. The relationship with distance domination leads to some interesting bounds on the burning number.

Suppose that $(x_1, x_2, \ldots, x_k)$ is a burning sequence for a given graph G. For $1 \leq i \leq k$, the fire started at x_i will burn only all the nodes within distance $k-i$ from x_i by the end of the k-th step. On the other hand, every node $v \in V(G)$ must be either a source of fire, or burned from at least one of the sources of fire by the end of the k-th step. In other words, every node of G must be an element of $N_{k-i}[x_i]$, for some $1 \leq i \leq k$. Moreover, for each pair i and j, with $1 \leq i < j \leq k$, we must have $d(x_i, x_j) \geq j-i$. Since otherwise, if $d(x_i, x_j) = l < j-i$, then x_j will be burned at stage $l+i$ $(< j)$, which is a contradiction. Therefore, we have the following useful observation.

Lemma 3. *A sequence $(x_1, x_2, \ldots, x_k)$ forms a burning sequence for a graph G if and only if for each pair i and j with $1 \leq i < j \leq k$, we have that $d(x_i, x_j) \geq j - i$, and the following set equation holds:*

$$N_{k-1}[x_1] \cup N_{k-2}[x_2] \cup \ldots \cup N_0[x_k] = V(G). \tag{2.1}$$

A *covering* of a graph G is a set of subsets of the nodes of G whose union is $V(G)$. The above lemma, shows that the burning problem is basically a covering problem using a set of closed neighbourhoods with a restriction on their radius. Hence, it seems that by finding a covering for a graph G using a limited number of connected subgraphs with restricted radius, we may find a bound on the burning number of G, as the following theorem shows.

Theorem 4. *If $\{C_1, C_2, \ldots, C_t\}$ is a collection of connected subgraphs in a graph G, each of radius at most k, which cover all the nodes of G, then $b(G) \leq t + k$.*

Proof. We define a burning sequence $(x_1, x_2, \ldots, x_{t'+k'})$, where $t' \leq t$ and $k' \leq k$, for G as follows. Let x_1 be a centre of the subgraph C_1. Then for $i \geq 2$, we let x_i be a central node in C_j, with $j \geq i$, if none of the central nodes of C_j are burned before the i-th step, where j is the smallest index that satisfies this condition. We continue to choose x_i's by the above rule until at some step $t' \leq t$, by burning $x_1, x_2, \ldots, x_{t'}$, each C_i, $1 \leq i \leq t$, contains a burned centre.

Now, for $j \geq 1$, we choose $x_{t'+j}$ to be a node in G that is not burned before the $(t' + j)$-th step. Since the radius of each C_i is at most k, after $k' \leq k$ steps every node in G must be burned. Thus, $b(G) \leq t' + k' \leq t + k$. □

Below we present another bound for the burning number of a graph using coverings. The proof is analogous to the one of Theorem 4.

Theorem 5. *If $\{C_1, C_2, \ldots, C_t\}$ is a covering for the nodes of a graph G, where each C_i is a connected subgraph of radius at most $k - i$, and $t \leq k$, then $b(G) \leq k$.*

Proof. Let x_1 be a centre of the subgraph C_1. Then for $i \geq 2$, we let x_i be a central node in C_j, with $j \geq i$, if none of the central nodes of C_j are burned before the i-th step, where j is the smallest index that satisfies this condition. We continue to choose x_i's by the above rule until at some step $k' \leq k$, by burning $x_1, x_2, \ldots, x_{k'}$, each C_i,

$1 \le i \le k$, contains a burned centre. Since the radius of each C_i is at most $k - i$, after k steps every node in G must be burned. Thus, $b(G) \le k$. □

The following corollary is sometimes helpful.

Corollary 6. *If $(x_1, x_2, \ldots, x_k)$ is a sequence of nodes in a graph G, such that $N_{k-1}[x_1] \cup N_{k-2}[x_2] \cup \ldots \cup N_0[x_k] = V(G)$, then $b(G) \le k$.*

Proof. Set $C_i = N_{k-i}[x_i]$, for $1 \le i \le k$, and apply Theorem 5. □

The following theorem gives us a slightly different covering technique than Theorem 4, and we use it for finding upper bounds on the burning number of graphs. The idea of the proof is similar to the proof of Theorem 4 though.

Theorem 7. *If $C_1, C_2, \ldots, C_{t+1}$ is a collection of connected subgraphs in a graph G with radius at most $k, k+1, \ldots, k+t$, respectively, which cover all the nodes of G, then $b(G) \le k + t + 1$.*

Proof. We define a burning sequence $(x_1, x_2, \ldots, x_{t'+k'})$, with $t' \le t + 1$ and $k' \le k$, for G as follows. Let x_1 be a centre of the subgraph C_{t+1}. Then for $i \ge 2$, we let x_i be a central node in C_j, with $1 \le j \le t + 1$, if none of the central nodes of C_j are burned before the i-th step, where j is the largest index that satisfies this condition. We continue to choose x_i's by the above rule until at some step $t' \le t + 1$, by burning $x_1, x_2, \ldots, x_{t'}$, each C_i, $1 \le i \le t + 1$, contains a burned central node.

Now, for $j \ge 1$, we choose $x_{t'+j}$ to be a node in G that is not burned before the $(t' + j)$-th step. Since the radius of each C_i is at most $k + i - 1$, after $k' \le k$ steps every node in G must be burned. Thus, $b(G) \le t' + k' \le t + 1 + k = k + t + 1$. □

The following theorem characterizes the graphs with burning number 2.

Theorem 8. *A graph G of order n satisfies $b(G) = 2$ if and only if G is of order at least 2, and has maximum degree $n - 1$ or $n - 2$.*

Proof. Suppose that G is a graph of order n with $b(G) = 2$, and (x_1, x_2) is an optimum burning sequence for G. Clearly, $n = |V(G)| \ge |\{x_1, x_2\}| = 2$, and by Equation (2.1) we know that $V(G) = N[x_1] \cup \{x_2\}$. It follows that every node in $V(G) \smallsetminus \{x_1, x_2\}$ is adjacent to x_1, since these nodes can only receive the fire from x_1, and have to be burned in the second step. There are two possibilities for x_2: either x_2 is adjacent to x_1 or not. If $x_2 \in N[x_1]$, then it implies that $\Delta(G) = d(x_1) = n-1$. If $x_2 \notin N[x_1]$, then we must have $\Delta(G) \geq d(x_1) = n-2$. In both cases, we conclude that $\Delta(G) \in \{n-1, n-2\}$.

Conversely, suppose that G is a graph of order $n \geq 2$ with maximum degree $n-1$ or $n-2$. Since G has at least two nodes, then we have that $b(G) \geq 2$. If $\Delta(G) = n-1 = d(u)$, for some node $u \in V(G)$, and $v \neq u$ is another node in $V(G)$ (note that $|V(G)| \geq 2$), then (u, v) must be a burning sequence for G, as u is a universal node. If $\Delta(G) = n-2 = d(u)$, for some node $u \in V(G)$, then there is a unique node $v \neq u$ that is not adjacent to u. Therefore, $V(G) = N[u] \cup \{v\}$. Hence, by Lemma 3, the sequence (u, v) is a burning sequence for G. In either case we conclude that $b(G) \leq 2$. Thus, we conclude that $b(G) = 2$. □

In the thesis, we mostly consider the burning problem for *connected graphs*. Note that as it is the case for many graph parameters, the burning number of a disconnected graph G with components $G_1, G_2, \ldots, G_t$, where $t \geq 2$, does not necessarily satisfy the equality

$$b(G) = b(G_1) + b(G_2) + \cdots + b(G_t).$$

For example, let G be the disjoint union of t paths of order 2.

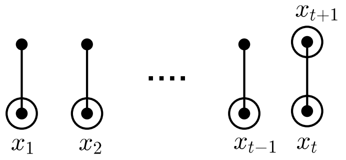


Figure 2.2: A burning sequence of length $t+1$ for G.

In order to burn all the components in G we have to choose at least one fire source from each of them. Thus, $b(G) \geq t$. On the other hand, let $b(G) = j$ for some $j \geq t$. Also, let G_i be the last component from which we choose a fire source, where $1 \leq i \leq t$. Since P_2 has two nodes, then we need exactly two steps for burning each component. Hence, the rest of the components are all burned by the end of the j-th step, and the last node in G_i will be burned by one more step. Therefore, G can not be burned in

t steps. Thus, we conclude that

$$b(G) = t + 1 < \sum_{i=1}^{t} b(G_i) = 2t,$$

as we can see an optimum burning sequence for G with $t+1$ burning steps in Figure 2.2. Generally, we have the following theorem for the burning number of disconnected graphs.

Theorem 9. *Let $G_1, G_2, \ldots, G_t$, with $t \geq 2$ be the components of a disconnected graph G. Then we have that*

$$t \leq b(G) \leq \sum_{i=1}^{t} b(G_i).$$

Proof. The lower bound is true, as for burning all the components, we need to choose at least one fire source from each of them. The upper bound holds, since by burning the nodes in a burning sequence for G_i, in order, for $1 \leq i \leq t$, we can burn all the nodes in G in exactly $\sum_{i=1}^{t} b(G_i)$ steps. □

Note that both the lower bound and the upper bound in the above theorem is achieved by a disconnected graph G in which each component is an isolated node. The upper bound in the above theorem also is obtained by a graph $G = G_1 \dot{\cup} G_2$, where G_1 is an isolated node, and G_2 is a path of squared order n^2. The proof of this is in Chapter 3, where we talk about the burning number of path-forests. Let G be a disconnected graph with components $G_1, G_2, \ldots, G_t$, with $t \geq 2$, in which G_1 is a graph of radius $t-1$, and for $i \geq 2$, G_i is a single node. Then the lower bound in Theorem 9 is achieved by G, as by burning a central node of G_1 at the first step, and burning the rest of the components in the next $t-1$ steps, the entire G will be burned. This follows since every node in G_1 is within distance $t-1$ from a central node of G_1.

Here is another lower bound for the burning number of the disconnected graphs.

Theorem 10. *If G is a disconnected graph with components $G_1, G_2, \ldots, G_t$, then*

$$b(G) \geq \max\{b(G_i) : 1 \leq i \leq t\}.$$

Proof. Assume that $b(G) = k$, and $(x_1, x_2, \ldots, x_k)$ forms a burning sequence for G.

By Lemma 3, we know that

$$V(G_i) \subset V(G) = N_{k-1}[x_i] \cup \ldots \cup N_0[x_k],$$

for $1 \leq i \leq t$. Note that by proof of Theorem 9, at least one of the x_j's must be from G_i. Suppose that $\{x_1, x_2, \ldots, x_k\} \cap V(G_i) = \{x_{i_1}, x_{i_2}, \ldots, x_{i_l}\}$, for some $1 \leq l \leq k$, where i_j is the j-th smallest index among the elements in $\{x_1, x_2, \ldots, x_k\} \cap V(G_i)$. We can see that for $1 \leq j \leq l$, $i_j \geq j$. Therefore, $\{N_{k-i_j}[x_{i_j}]\}_{j=1}^{l}$ forms a covering of G_i such that $k - i_j \leq k - j$, for $1 \leq j \leq l$. Thus, by Theorem 5, we conclude that $b(G_i) \leq k$. Since this is true for any $1 \leq i \leq t$, we have that $b(G) \geq \max\{b(G_i) : 1 \leq i \leq t\}$. □

Note that in Theorem 8, we do not need to assume that G is connected. Hence, Theorem 8 together with Theorem 9 helps us to characterize the disconnected graphs with burning number 2.

Corollary 11. *Let G be a disconnected graph. Then $b(G) = 2$ if and only if G has exactly two components, say G_1 and G_2, such that G_1 has a universal node and G_2 is a single node.*

Proof. Suppose that G is a disconnected graph of order n with $b(G) = 2$. By the lower bound in Theorem 9, we conclude that G must have exactly two components such as G_1 and G_2. Since every graph with a universal node is connected, then Theorem 8 implies that $\Delta(G) = n - 2 = d(u)$, for some node $u \in V(G)$. Thus, there must be a node $v \neq u$ that is not adjacent to u. It implies that v is an isolated node, otherwise, it must be connected to a neighbour of u which is a contradiction. Hence, without loss of generality, we conclude that $V(G_1) = N[u]$ and $G_2 = \{v\}$.

Conversely, assume that $G = G_1 \uplus G_2$, such that G_1 has a universal node such as u, and $G_2 = \{v\}$. Then Theorem 8 and Theorem 9, imply that (u, v) is an optimum burning sequence for G. Thus, the proof follows. □

In the rest of the thesis we assume that all graphs are connected, unless we specify that we talk about a disconnected graph. As another simple example, and a corollary of Theorem 8, we can find the burning number of complete bipartite graphs as follows.

Theorem 12. *Let $G = K_{m,n}$ be a complete bipartite graph of order $m + n$, where $m, n \geq 1$. If $\min\{m, n\} \geq 3$, then $b(G) = 3$; otherwise, $b(G) = 2$.*

Proof. If $m, n \geq 3$, then we know that the maximum degree of such a complete bipartite graph is not in $\{m+n-1, m+n-2\}$. Thus, by Theorem 8, we have that $b(K_{m,n}) > 2$, where $m, n \geq 3$. On the other hand, let (x_1, x_2, x_3) be a sequence of nodes in G such that all x_i's are in the same part of G. Since $N_2[x_1] = G$, then by Equation (2.1), we conclude that (x_1, x_2, x_3) is a burning sequence for G. Therefore, $b(G) = 3$.

If one of m and n equals one, then it implies that G is a star, and the central node of the star is a node of degree $\Delta(G) - 1$. Therefore, by Theorem 8, we have that $b(G) = 2$. If $min\{m, n\} = 2$, then $\Delta(G) = m + n - 2$. Thus, by Theorem 8, we conclude that $b(G) = 2$. □

We note that in Chapter 3, we show that the burning problem is **NP**-complete for bipartite graphs in general. The following theorem provides an alternative characterization of the burning number. The *depth* of a node in a rooted tree T is the number of edges in a shortest path from the node to the tree's root; in other words, the distance between the node and the root of T. The set of the nodes in T that have depth i is called the *i-th level* of the rooted tree. The *height* of T is the greatest depth in T. A *rooted tree partition* of G is a collection of rooted trees which are subgraphs of G, with the property that the node sets of the trees partition $V(G)$.

Theorem 13. *Burning a graph G in k steps is equivalent to finding a rooted tree partition into k trees $T_1, T_2, \ldots, T_k$, with heights at most $(k-1), (k-2), \ldots, 0$, respectively such that for every $1 \leq i, j \leq k$ the distance between the roots of T_i and T_j is at least $|i - j|$.*

Proof. Assume that $(x_1, x_2, \ldots, x_k)$ is a burning sequence for G. For all $1 \leq i \leq k$, after x_i is burned, in each round $t > i$ those unburned nodes of G in the $(t-i)$-neighborhood of x_i will burn. Hence, any node v is burned by receiving fire via a shortest path of burned nodes from some fire source x_i (this path can be of length zero in the case that $v = x_i$). Hence, we may define a surjective function $f : V(G) \to \{x_1, x_2, \ldots, x_k\}$, with $f(v) = x_i$ if v receives fire from x_i, where i is chosen with the smallest index if needed. Now $\{f^{-1}(x_1), f^{-1}(x_2), \ldots, f^{-1}(x_k)\}$ forms a partition of $V(G)$ such that $G[f^{-1}(x_i)]$ (that is, the subgraph induced by $f^{-1}(x_i)$) forms a connected subgraph of G. Since every node v in $f^{-1}(x_i)$ receives the fire spread from x_i through a shortest path between x_i and v, by deleting extra edges in $G[f^{-1}(x_i)]$ we can make a rooted subtree of G, called T_i with root x_i. Since every node is burned after k steps, the distance between each node on T_i and x_i is at most $k - i$. Therefore, the height of T_i is at most $k - i$.

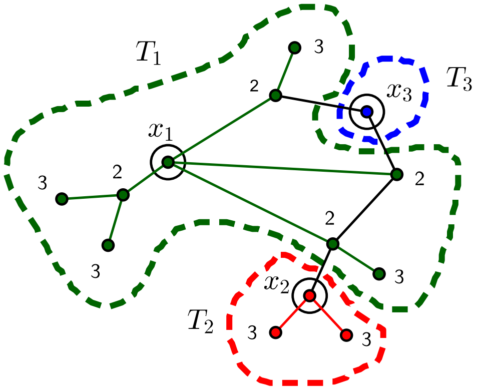


Figure 2.3: A rooted tree partition.

Conversely, suppose that we have a decomposition of the nodes of G into k rooted subtrees $T_1, T_2, \ldots, T_k$, such that for each $1 \leq i \leq k$, T_i is of height at most $k - i$. Assume that $x_1, x_2, \ldots, x_k$ are the roots of $T_1, T_2, \ldots, T_k$, respectively, and for each pair i and j, with $1 \leq i < j \leq k$, $d(x_i, x_j) \geq j - i$. Thus, by Lemma 3, $(x_1, x_2, \ldots, x_k)$ must be a burning sequence for G. □

Figure 2.3 illustrates Theorem 13. The burning sequence is (x_1, x_2, x_3). We have shown the decomposition of G into subgraphs T_1, T_2, and T_3 based on this burning sequence by drawing dashed curves around the corresponding subgraphs. Each node has been indexed by a number corresponding to the step that it is burned.

The following corollary is useful for determining the burning number of a graph, as it reduces the problem of burning a graph to burning its spanning trees. Note that for a spanning subgraph H of G, we have that $b(G) \leq b(H)$. This follows since by Equation (2.1), every optimum burning sequence for H induces a node covering for $V(G) = V(H)$, and therefore, by Corollary 6, induces a burning sequence of at most the same length for G.

Corollary 14. *For a graph G we have that*

$$b(G) = \min\{b(T) : T \text{ is a spanning tree of } G\}.$$

Proof. By Theorem 13, we assume that $T_1, T_2, \dots, T_k$ is a rooted tree partition of G, where $k = b(G)$, derived from an optimum burning sequence for G. If we take T to be a spanning subtree of G obtained by adding edges between the T_i's which do not induce a cycle in G, then $b(T) \le k = b(G) \le b(T)$, where the second inequality holds since T is a spanning subgraph of G. □

A subgraph H of a graph G is called an *isometric subgraph* if for every pair of nodes u and v in H, we have that $d_H(u,v) = d_G(u,v)$. For example, a subtree of a tree is an isometric subgraph. As another example, if G is a connected graph and P is a shortest path connecting two nodes of G, then P is an isometric subgraph of G. Let W_5 be the wheel graph formed by adding a universal node to a 5-cycle. Then the 5-cycle C_5 is an isometric subgraph of W_5, while $b(C_5) = 3 > 2 = b(W_5)$. Thus, we conclude that the burning number is not monotonic even on the isometric subgraphs of a graph. However, the following theorem shows that the burning number is monotonic on the isometric subgraphs in certain cases.

Theorem 15. *Suppose that H is an isometric subgraph of a graph G such that, for any node $x \in V(G) \setminus V(H)$, and any positive integer r, there exists a node $f_r(x) \in V(H)$ for which $N_r[x] \cap V(H) \subseteq N_r^H[f_r(x)]$. Then we have that $b(H) \le b(G)$.*

Proof. It suffices to show that for any optimum burning sequence such as $(x_1, \dots, x_k)$ for G we can assign a burning sequence of length at most k to H. Without loss of generality, we may assume that $|V(H)| > k$ (otherwise, H can be burned in at most $|V(H)| \le k$ steps).

We define the function $f : \{x_1, x_2, \dots, x_k\} \to V(H)$ as follows. For $1 \le i \le k$, if $x_i \in V(H)$, then we define $f(x_i) = x_i$; otherwise, by assumption, there is a node $f_{k-i}(x_i) \in V(H)$ for which $N_{k-i}[x_i] \cap V(H) \subseteq N_{k-i}^H[f_{k-i}(x_i)]$. In this case, we define $f(x_i) = f_{k-i}(x_i)$. Since H is an isometric subgraph of G, then for each node x_i with $f(x_i) = x_i$, and for every node $v \in N_{k-i}[x_i] \cap V(H)$, we have that $d_H(x_i, v) = d_G(x_i, v) \le k - i$. Thus, if $f(x_i) = x_i$, then $N_{k-i}[x_i] \cap V(H) = N_{k-i}^H[x_i] = N_{k-i}^H[f(x_i)]$. Hence, we derive that

$$\begin{aligned} V(H) &= V(G) \cap V(H) \\ &= (N_{k-1}[x_1] \cup \ldots \cup N_0[x_k]) \cap V(H) \\ &= (N_{k-1}[x_1] \cap V(H)) \cup \ldots \cup (N_0[x_k] \cap V(H)) \\ &\subseteq N^H_{k-1}[f(x_1)] \cup \ldots \cup N^H_0[f(x_k)]. \end{aligned}$$

Therefore, $\{N_{k-i}[f(x_i)]\}_{i=1}^{k}$ forms a covering for the node set of H, with k closed neighbourhoods. Thus, by Corollary 6, we conclude that $b(H) \leq b(G)$. □

The following theorem shows that the isometric subtrees of a graph satisfy the conditions in Theorem 15.

Theorem 16. *For any isometric subtree H of a graph G, we have that $b(H) \leq b(G)$.*

Proof. By Theorem 15, it suffices to show that for any node $x \in V(G) \setminus V(H)$, and any positive integer r, there exists a node $f_r(x) \in V(H)$ for which $N_r[x] \cap V(H) \subseteq N^H_r[f_r(x)]$. Fix $x \in V(G) \setminus V(H)$.

Set $X_r = N_r[x] \cap V(H)$. If X_r is empty, then we can choose $f_r(x)$ to be any node in H. If $X_r = \{v\}$, then we take $f_r(x) = v$, and in this case, $N_r[x] \cap V(H) = \{v\} \subseteq N^H_r[v]$. Hence, we assume that $|X_r| \geq 2$. Since H is a tree, there is a unique path (consisting of the nodes in H only) between every pair of distinct nodes in $X_r \subseteq V(H)$. Let y_r and z_r be two nodes in X_r with the maximum distance over all possible pairs of nodes in X_r, and let w_r be a node in H that is of almost equal distance with respect to y_r and z_r. That is, $d(w_r, y_r) = d(w_r, z_r)$, if $d(y_r, z_r)$ is even, and $d(w_r, z_r) = d(w_r, y_r) + 1$ (without of loss of generality) in the case that $d(y_r, z_r)$ is odd. We claim that for each $v \in X_r$, $d(v, w_r) \leq r$.

Since H is an isometric subtree of G, the length of the path between y_r and z_r in H is equal to $d(y_r, z_r)$ in G. Thus, we have that $d(y_r, z_r) = d(w_r, y_r) + d(w_r, z_r) \leq d(x, y_r) + d(x, z_r)$. On the other hand, we have that $d(v, w_r) \leq d(z_r, w_r)$. To show this, we have to consider two possibilities; either v is on the path in H that connects z_r to w_r, or it is not. If the former holds, then $d(v, w_r) \leq d(z_r, w_r)$. If the latter holds, then suppose u is the first node that appears in both paths that connect v and z_r to

w_r. If $u = w_r$, then we have that

$$d(v, w_r) + d(w_r, z_r) = d(v, z_r) \leq d(y_r, z_r) = d(y_r, w_r) + d(w_r, z_r).$$

This implies that $d(v, w_r) \leq d(y_r, w_r) \leq d(z_r, w_r)$. If $u \neq w_r$, then we have that

$$d(v, w_r) + d(w_r, y_r) = d(v, y_r) \leq d(z_r, y_r) = d(z_r, w_r) + d(w_r, y_r).$$

Hence, we again conclude that $d(v, w_r) \leq d(z_r, w_r)$. Consequently, we have that

$$\begin{aligned} d(v, w_r) \leq d(z_r, w_r) &\leq \frac{d(y_r, z_r) + 1}{2} \\ &\leq \frac{d(y_r, x) + d(z_r, x) + 1}{2} \\ &\leq \frac{r + r + 1}{2} = r + \frac{1}{2}. \end{aligned}$$

Since $d(v, w_r)$ is an integer, it implies that $d(v, w_r) \leq r$. Therefore, if we define $f_r(x) = w_r$, then $X_r \subseteq N_r^H[f_r(x)]$. Thus, the proof follows. □

The above inequality however may fail for non-isometric subtrees. For example, let H be a path of order 5, and form G by adding a universal node (that is, one joined to all others) to H. Then $b(H) = 3$, but $b(G) = 2$.

The following corollary is a consequence of Theorem 16.

Corollary 17. *If H is a subtree of a tree T, then we have that $b(H) \leq b(T)$.*

Proof. It is derived from Theorem 16, since H is an isometric subtree of T. □

Here is a generalization of Theorem 16.

Corollary 18. *If H is an isometric subforest of a graph G, then we have that $b(H) \leq b(G)$.*

Proof. It is derived from Theorem 15, since H is an isometric subgraph of G with the desired property. Namely, every component of H is an isometric subtree of a component of G. Hence, by Theorem 16, we conclude that H satisfies the conditions in Theorem 15. Note that in proof of Theorem 15, we do not need the connectivity as part of the assumption. □

The burning number of paths is derived in the following result.

Theorem 19. *For a path P_n on n nodes, we have that $b(P_n) = \lceil n^{1/2} \rceil$.*

Proof. Suppose that $(x_1, x_2, \ldots, x_k)$ is an optimum burning sequence for P_n. By Equation (2.1), and the fact that for a node v in a path, $|N_i[v]| \leq 2i+1$ we derive that

$$\begin{aligned}
&(2(k-1) + 2(k-2) + \ldots + 2(1)) + k \\
&= 2\left(\frac{k(k-1)}{2}\right) + k \\
&= k^2 \geq n.
\end{aligned}$$

Since k is the minimum number satisfying this inequality, we conclude that $b(P_n) \geq \lceil n^{1/2} \rceil$.

Now, assume that $k = \lceil n^{1/2} \rceil$, and let $P_n : v_1, v_2, \ldots, v_n$. Since $n = 1 + 3 + \cdots + 2k - 3 + \left(n - (k-1)^2\right)$, then we have a partition for the nodes of P_n into paths of orders $1, 3, \ldots, 2k-3$, and $n-(k-1)^2$. We can choose the paths in such a partition to appear from the right to the left of P_n in the following order:

$$P_1, P_3, \ldots, P_{2k-3}, P_{n-(k-1)^2}.$$

Now, for $0 \leq i \leq k-2$, we choose $x_{k-i} = v_{n-i^2-i}$. Also, if $n \geq (k-1)^2 + k$, we take $x_1 = v_{n-(k-1)^2-(k-1)}$; otherwise we take $x_1 = v_1$. We can see that, every node x_i, for $2 \leq i \leq k$, is the centre of the path of order $2(k-i)+1$ that appears in the given partition for the nodes of P_n. Besides, the node x_1 is a node in the subpath $P_{n-(k-1)^2}$ such that $d(x_1, x_j) \geq j-1$, for $2 \leq j \leq k$, and every node in the path $P_{n-(k-1)^2}$ is within distance at most $(k-1)$ from x_1. Therefore, by Lemma 3, we conclude that $(x_1, x_2, \ldots, x_k)$ is a burning sequence for P_n. Hence, $b(P_n) \leq k$. Thus, $b(P_n) = \lceil n^{1/2} \rceil$. □

Note that since every subtree of a path is a subpath, then by Theorem 13, burning a path P_n is equivalent to decomposing P_n into some rooted subpaths that the root of each of them is a fire source. In the above proof, the optimum burning sequence that we defined is indeed the sequence of the roots of such a path-decomposition for P_n. More precisely, for $2 \leq i \leq k$, the fire source x_i is the middle node of a path of order $2(k-i)+1$, and x_1 is a node of a path of order at most $2(k-1)+1$ (depending on whether or not n is a square number) in such a path decomposition for P_n. In particular, if n is a square number, then burning P_n in $k = n^{1/2}$ steps is equivalent to decomposing P_n into paths of orders $1, 3, \dots, 2k - 1$.

We state the following lemma since we use it for proving some of the results on the burning problem. Assume that G and H are two disjoint graphs, and $u \in V(G)$ and $v \in V(H)$ are two nodes. We can make a new graph $G + uv + H$ by adding edge uv to $G \uplus H$.

Lemma 20. *If G and H are two disjoint non-empty graphs then we have that*

$$b(G + uv + H) \leq b(G \uplus H),$$

where $u \in V(G)$ and $v \in V(H)$.

Proof. Since $V(G + uv + H) = V(G \uplus H)$, then every burning sequence for $G \uplus H$ induces a covering for $G + uv + H$; in particular, any minimum burning sequence of $G \uplus H$ induces a covering for $G + uv + H$. Therefore, $b(G + uv + H) \leq b(G \uplus H)$. □

For example, we can combine Lemma 20 and Theorem 19 to conclude the following result for path-forests. As we mentioned in Chapter 1, by a path-forest we mean a graph that is a disjoint union of paths.

Theorem 21. *If G is a path-forest of order n and with $t \geq 1$ components, then we have that*

$$\lceil n^{1/2} \rceil \leq b(G) \leq \lceil n^{1/2} \rceil + t - 1.$$

Proof. Assume that $Q_1, Q_2, \dots, Q_t$ are the components of G. Then we glue the paths $Q_1, Q_2, \dots, Q_t$ together by adding edges $e_1, e_2, \dots, e_{t-1}$ in which each edge e_i connects a leaf of the path Q_i and a leaf of the path Q_{i+1}, for $1 \leq i \leq t - 1$. Thus, we have a path of order n, called P such that G is a subforest of P. Hence, by Theorem 19 and by applying Lemma 20, we conclude that $b(G) \geq b(P) = \lceil n^{1/2} \rceil = k$.

On the other hand, by Theorem 19, we know that burning P in k steps is equivalent to decomposing P into k subpaths such that the i-th subpath in such a partition is of order at most $2i - 1$, for $1 \leq i \leq k$. Thus, by deleting each edge e_i from P, we may break down at most one of the subpaths in this partition for P into two parts. Hence, by deleting $e_1, e_2, \dots, e_{t-1}$ from such a partition for P we will have at most

$k + t - 1$ subpaths that partition G such that the i-th subpath in this partition is of order at most $2i - 1$, where $1 \le i \le k + t - 1$. Thus, by Corollary 6, we conclude that $b(G) \le k + t - 1$. □

We also have the following corollaries from Theorem 19.

Corollary 22. *1. For a cycle C_n, we have that $b(C_n) = \lceil n^{1/2} \rceil$.*

2. For a graph G of order n with a Hamiltonian path, we have that $b(G) \le \lceil n^{1/2} \rceil$.

Proof. 1. The only spanning subtree of cycle C_n is the path P_n that is obtained by deleting an edge from C_n. Thus, by Corollary 14 and Theorem 19, we conclude that

$$b(C_n) = \min\{b(T) : T \text{ is a spanning subtree of } C_n\} = b(P_n) = \lceil n^{1/2} \rceil.$$

2. Suppose that G is a graph of order n with a Hamiltonian path P_n. Since P_n is a spanning subgraph of G, then by Theorem 19, we conclude that $b(G) \le b(P_n) = \lceil n^{1/2} \rceil$.

□

We recall from Chapter 1 that a spider graph is a tree that has only one node of degree at least three, and $SP(s, r)$ denotes a spider graph with s arms in which all arms are of the same length r. The following theorem gives sharp bounds on the burning number.

Lemma 23. *For any graph G with radius r and diameter d, we have that*

$$\lceil (d + 1)^{1/2} \rceil \le b(G) \le r + 1.$$

Proof. Assume that c is a central node of G with eccentricity r. Since every node in G is within distance r from c, then $N_r[c]$ forms a covering for $V(G)$. Hence, by Theorem 4, $r + 1$ is an upper bound for $b(G)$.

Now, let P be a shortest path connecting two nodes u and v in G with $d(u, v) = d$. Since P is an isometric subtree of G of order $d + 1$, by Theorem 16 and Theorem 19, we conclude that $b(G) \ge b(P) = \lceil (d + 1)^{1/2} \rceil$. □

The lower bound is achieved by paths as we show in Theorem 19, and the upper bound is achieved by spider graphs $SP(s, r)$, where $s \geq r$, and also by perfect binary trees (as we prove in Chapter 3).

We finish this section by providing some bounds on the burning number in terms of certain domination numbers. A *k-distance dominating set* D_k for G is a subset of nodes such that for every node $u \in V(G) \setminus D_k$, there exists a node $v \in D_k$, with $d(u, v) \leq k$. The number of nodes in a minimum k-distance dominating set of G is denoted by $\gamma_k(G)$ and we call it the *k-distance domination number* of G. For example, let G be the graph shown in Figure 2.4. Then the nodes with a circle around them form a minimum 2-distance dominating set for G.

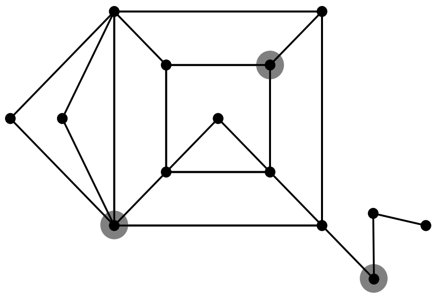


Figure 2.4: An optimum 2-distance dominating set for G.

We have the following results on connections between burning and distance domination.

Theorem 24. *If G has burning number k, then $k \geq \gamma_{k-1}(G)$.*

Proof. Assume that $b(G) = k$, for some positive integer k, and $(x_1, x_2, \ldots, x_k)$ is an optimum burning sequence for G. Then by (2.1), we know that every node v in G must be within the distance $k - i \leq k - 1$ from one of the x_i's. Hence, $D = \{x_1, x_2, \ldots, x_k\}$ forms a $(k-1)$-distance dominating set for G. □

We have the following lemma.

Lemma 25. *For any graph G, if $m = \min_{k \geq 1}\{\gamma_k(G) + k\}$, then $\frac{m+1}{2} \leq b(G) \leq m$.*

Proof. Assume that $m = \min_{k\geq 1}\{\gamma_k(G) + k\}$, and $b(G) = k_0$. Then by Theorem 24, $b(G) = k_0 \geq \gamma_{k_0-1}$. Hence,

$$k_0 + (k_0 - 1) \geq \gamma_{k_0-1} + k_0 - 1 \geq \min_{k\geq 1}\{\gamma_k(G) + k\} = m.$$

Therefore, $k_0 \geq \frac{m+1}{2}$.

On the other hand, assume that $D_k = \{x_1, x_2, \ldots, x_{\gamma_k}\}$ is a minimum k-distance dominating set for G. Then $\{S_1, S_2, \ldots, S_{\gamma_k}\}$, with $S_i = \{v \in V(G) : d(v, x_i) \leq k\}$, where $1 \leq i \leq \gamma_k$, is a covering for the nodes of G which consists of $\gamma_k(G)$ subsets each of radius at most k. Thus, by Theorem 4, we have that $b(G) \leq \gamma_k + k$. The result follows since this is true for any $k \geq 1$. Therefore, we have that

$$\frac{m+1}{2} \leq b(G) \leq m,$$

and the Theorem follows. □

We have the following fact about the k-distance domination number of graphs.

Theorem 26 ([45]). *If G is a connected graph of order n, with $n \geq k+1$, then we have that*

$$\gamma_k(G) \leq \frac{n}{k+1}.$$

We now provide the following general bound for the burning number of graphs.

Corollary 27. *If G is a connected graph of order n, with $n \geq k+1$, then we have that*

$$b(G) \leq 2\lceil n^{1/2}\rceil - 1.$$

Proof. By Lemma 25 and Theorem 26, we derive that for any positive integer $k \leq n-1$

$$b(G) \leq \min_{k\geq 1}\left\{\frac{n}{k+1} + k\right\}.$$

Now, the function $\frac{n}{k+1} + k$ is minimized for $k = \lceil n^{1/2}\rceil - 1$, and we note that $k \leq n-1$.

Therefore, we have that

$$\begin{aligned} b(G) &\leq \min_{k\geq 1}\left\{\frac{n}{k+1}+k\right\} \\ &\leq \frac{n}{(\lceil n^{1/2}\rceil - 1)+1} + \lceil n^{1/2}\rceil - 1 \\ &\leq 2\lceil n^{1/2}\rceil - 1, \end{aligned}$$

and the proof follows. □

We have the following conjecture.

Conjecture 28. *If G is a graph of order n with s components, then $b(G) \leq \lceil n^{1/2}\rceil + s - 1$. In particular, if G is a connected graph of order n, then $b(G) \leq \lceil n^{1/2}\rceil$.*

Note that by Theorem 19 and Theorem 21, we already know the truth of this conjecture for paths and path-forests. We will also see in Chapter 3 that the bound in Conjecture 28 is achieved by specific spider graphs.

2.2 Nordhaus-Gaddum Type Results

Nordhaus and Gaddum [47] gave bounds on the sum and product of the chromatic number of a graph and its complement, in terms of the order of the graph. Analogous relations have been discovered for many other graph parameters; see [4] for a survey. In this section, we present Nordhaus-Gaddum type results for the burning number.

We need first the following simple observation. Let G be a graph of order $n \geq 2$ with maximum degree Δ. If G does not have a universal node, then we have that $b(G) \leq n - \Delta$; otherwise, $b(G) = 2$. Namely, we can take a node such as v of degree Δ, and then by burning v and $V(G) \setminus N[v]$, respectively, we burn all nodes of G in at most $1 + |V(G) \setminus N[v]| = n - \Delta$ steps. If G contains a universal node v, then by Theorem 8, we have that $b(G) = 2$.

We first present some bounds on the sum of the burning numbers of a graph and its complement.

Theorem 29. *If G is a graph of order $n \geq 2$, then*

$$4 \leq b(G) + b(\overline{G}) \leq n + 2.$$

Proof. Suppose that $b(G) = k$, and $(x_1, x_2, \ldots, x_k)$ is a burning sequence for G. By Lemma 3, x_k cannot be adjacent to x_i, for $1 \le i \le k-2$. Therefore, $\Delta(\overline{G}) \ge d_{\overline{G}}(x_k) \ge k-2$. If G does not have an isolated node, then $\overline{G}$ does not have a universal node. Thus, by the above observation, $b(\overline{G}) \le n - \Delta(\overline{G}) \le n - (k-2)$, and consequently, we have that $b(\overline{G}) + b(G) \le n + 2$.

If G has an isolated node v, then $\overline{G}$ must have a universal node. Therefore, $b(\overline{G}) = 2$, and clearly, $b(G) \le n$. Thus, $b(\overline{G}) + b(G) \le n + 2$. Finally, since both G and $\overline{G}$ are of order at least two, then $b(\overline{G}) + b(G) \ge 2 + 2 = 4$. □

The upper bound in Theorem 29 is the best possible, since for the complete graph K_n we have that $b(\overline{K_n}) + b(K_n) = n + 2$. However, there are cases where the upper bound is strict; for example, using Corollary 22, we have that $b(C_n) + b(\overline{C_n}) = \lceil n^{1/2} \rceil + 3 < n + 2$. Also, the lower bound in Theorem 29 is achieved for the complete graph K_2, and star graph $K_{1,s}$, where $s \ge 2$.

We cite here two useful Nordhaus-Gaddum type results for distance domination.

Theorem 30 ([4])**.** *For any graph G of order $n \ge k+1$ with $k \ge 2$, we have that*

$$\gamma_k(G) + \gamma_k(\overline{G}) \le n + 1$$

and

$$\gamma_k(G)\gamma_k(\overline{G}) \le n.$$

Theorem 31 ([4])**.** *If G and $\overline{G}$ are both connected with $n \ge k+1$ nodes for integer $k \ge 2$, then $\gamma_k(G) + \gamma_k(\overline{G}) \le \frac{n}{k+1} + 1$ and $\gamma_k(G)\gamma_k(\overline{G}) \le \frac{n}{k+1}$.*

We now have the following result for the product of the burning numbers of a graph and its complement.

Theorem 32. *For any graph G of order n, we have $b(G)b(\overline{G}) \le 2n$, and the equality is achieved by complete graphs.*

Proof. First, by direct checking we can see that $b(G)b(\overline{G}) \le 2n$, for any graph G of order $n \le 5$. Now, assume that G is a graph of order $n \ge 6$. By Lemma 25, we have that $b(G) \leq \gamma_k(G) + k$, for $k \geq 2$ with $n \geq k+1$. Thus, we have that

$$\begin{aligned} b(G)b(\overline{G}) &\leq (\gamma_k(G)+k)(\gamma_k(\overline{G})+k) \\ &\leq \gamma_k(G)\gamma_k(\overline{G}) + k\gamma_k(G) + k\gamma_k(\overline{G}) + k^2. \end{aligned}$$

Now, if G and $\overline{G}$ are both connected, then using Theorem 31, we have that

$$\begin{aligned} b(G)b(\overline{G}) &\leq \gamma_k(G)\gamma_k(\overline{G}) + k\gamma_k(G) + k\gamma_k(\overline{G}) + k^2 \\ &\leq \frac{n}{k+1} + k\left(\frac{n}{k+1} + 1\right) + k^2 \\ &= n + k + k^2. \end{aligned}$$

By taking $k = 2$, the above inequality implies that $b(G)b(\overline{G}) \leq n + 6$.

If G is connected while $\overline{G}$ is disconnected, then either $\overline{G}$ has a component with at most two nodes, or every component of $\overline{G}$ has at least three nodes. If $\overline{G}$ has a component with at most two nodes, then G must have either a universal node, or it must contain two nodes such as u and v (corresponding to the component of $\overline{G}$ with exactly two nodes), such that $G = N[u] \cup \{v\}$. Thus, by Theorem 8, $b(G) = 2$, and obviously $b(\overline{G}) \leq n$. Hence, in this case, $b(G)b(\overline{G}) \leq 2n$.

Now, suppose that $G_1, G_2, \ldots, G_t$ are the components of $\overline{G}$ with $n_1, n_2, \ldots, n_t$ nodes, respectively, where each $n_i \geq 3$. By Theorem 26, and taking $k = 2$, we know that

$$\begin{aligned} \gamma_2(\overline{G}) &= \gamma_2(G_1) + \gamma_2(G_2) + \cdots + \gamma_2(G_t) \\ &\leq \frac{n_1}{3} + \frac{n_2}{3} + \cdots + \frac{n_t}{3} \\ &= \frac{n}{3}. \end{aligned}$$

Also, note that $b(G) \leq 3$, since in such a case the radius of G is at most 2. Namely, suppose that $x \in V(G_1)$ and $y \in V(G) \setminus V(G_1)$. We know that such nodes x and y exist since $\overline{G}$ is disconnected. Thus, in graph G, any node in $V(G_1 - x)$ is adjacent to y, and x is adjacent to all the nodes in $V(G) \setminus V(G_1)$. Hence, all the nodes in $V(G)$ are within distance at most 2 from x in graph G. Therefore, by Theorem 24 and Theorem 26, for $k = 2$, we have that

$$\begin{aligned} b(G)b(\overline{G}) &\leq 3(\gamma_2(\overline{G}) + 2) \\ &\leq 3\left(\frac{n}{3} + 2\right) \\ &= n + 6, \end{aligned}$$

and the proof follows. □

Corollary 33. *If graphs G and $\overline{G}$ are connected graphs of order $n \geq 6$, then $b(G) + b(\overline{G}) \leq 3\lceil n^{1/2} \rceil - 1$, and $b(G)b(\overline{G}) \leq n + 6$.*

Proof. First, by Lemma 24, we have that

$$b(G) + b(\overline{G}) \leq (\gamma_k(G) + k) + (\gamma_k(\overline{G}) + k) = \gamma_k(G) + \gamma_k(\overline{G}) + 2k.$$

By applying Theorem 31 with $k = \lceil n^{1/2} \rceil - 1$, we conclude that

$$b(G) + b(\overline{G}) \leq \frac{n}{k+1} + 1 + 2k \leq 3\lceil n^{1/2} \rceil - 1.$$

Finally, $b(G)b(\overline{G}) \leq n + 6$ follows from the proof of Theorem 32. □

We end this section by the following conjecture.

Conjecture 34. *If G and $\overline{G}$ are both connected graphs of order n, then we have that $b(G)b(\overline{G}) \leq n + 4$.*

It is straightforward to check that the bound in the conjecture is tight for the cycle C_5.

2.3 Burning in the ILT Model

In this section, we consider the burning number of the graphs generated by the ILT model. The *Iterated Local Transitivity* (ILT) model was introduced in [15], and simulates on-line social networks (or OSNs). The central idea behind the ILT model is what sociologists call *transitivity*: if u is a friend of v, and v is a friend of w, then

u is a friend of w. In its simplest form, transitivity gives rise to the notion of *cloning*, where u is joined to all of the neighbours of v.

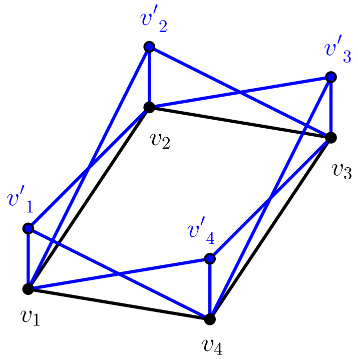


Figure 2.5: The graph G_1 generated from $G_0 = C_4$.

In the ILT model, given some initial graph as a starting point, nodes are repeatedly added over time which clone *each* node, so that the new nodes form an independent set. The only parameter of the model is the initial graph G_0, which is any fixed finite connected graph. Assume that for a fixed $t \geq 0$, the graph G_t has been constructed. To form G_{t+1}, for each node $x \in V(G_t)$, add its *clone* x', such that x' is joined to x and all of its neighbours at time t. Note that the set of new nodes at time $t+1$ form an independent set of cardinality $|V(G_t)|$. Figure 2.5 shows the graph G_1 generated by the ILT model starting from the graph $G_0 = C_4$. The nodes and edges with colour blue are those nodes and edges that we add to G_0 in order to make G_1.

The ILT model shares many properties with OSNs such as low average distance, high clustering coefficient densification, and bad spectral expansion; see [15]. The ILT model has also been studied from the viewpoint of competitive diffusion which is one model of the spread of influence; see [54].

We have the following theorem about the burning number of graphs obtained based on ILT model. Even though the graphs generated by the ILT model grow exponentially in order with t, we see that the burning number of such networks remains constant. First, we need the following elementary lemma for proving the main result of this section.

Lemma 35. *For each $i \geq 0$, let G_i be the graph generated at time i based on the ILT model with initial graph G_0. Then G_i is an isometric subgraph of G_j, for any $0 \leq i < j$.*

Proof. We need to prove that $d_{G_j}(u,v) = d_{G_i}(u,v)$, for every two nodes $u, v \in V(G_i)$. Assume that u and v are two distinct nodes in G_i. Since G_i is a subgraph of G_j, then

$$d_{G_j}(u,v) \leq d_{G_i}(u,v).$$

By contradiction, suppose that $d_{G_j}(u,v) = m < d_{G_i}(u,v)$, for some positive integer m. Thus, there must be a shortest path $P : u, u_1, u_2, \ldots, u_{m-1}, v$ between u and v in G_j, such that at least one of the nodes in P is not in G_i; otherwise, it implies that $d_{G_j}(u,v) = d_{G_i}(u,v)$, which is a contradiction.

For any shortest path (of length m) such as Q between u and v in G_j, let n_Q denote the number of the nodes in Q that are in $G_j \setminus G_i$. Assume that s is the smallest index for which $u_s \notin G_i$. Let u' be the clone of u_s in G_i. Note that by definition, we know that u_{s-1} and u_{s+1} are also both adjacent to u' in G_j.

First, we claim that $u' \neq u_t$, where $1 \leq t \leq m-1$. Otherwise, if $u' = u_t$, for some $1 \leq t \leq m-1$, we have two possibilities: either $t \leq s-1$, or $t \geq s+1$. If $1 \leq t \leq s-1$, then since $u' = u_t$ is adjacent to u_{s+1}, then the path $P' : u, u_1, \ldots, u_t = u', u_{s+1}, \ldots, u_{m-1}, v$ will be a shorter path between u and v in G_j, which is a contradiction. If $u' = u_t$, where $t \geq s+1$, then since $u' = u_t$ is adjacent to u_{s-1}, then the path $P' : u, u_1, \ldots, u_{s-1}, u_t = u', u_{t+1}, \ldots, u_{m-1}, v$ will be a shorter path between u and v in G_j, which is again a contradiction. Thus, $u' \neq u_t$, where $1 \leq t \leq m-1$.

Now, by above claim, we conclude that $Q : u, u_1, \ldots, u_{s-1}, u', u_{s+1}, \ldots, u_{m-1}, v$ is a path between u and v in G_j of length $m = d_{G_j}(u,v)$. However, $n_Q < n_P$. Therefore, by the same argument as above, after finite number of times, we can find a path of length m between u and v such that none of its nodes are in $G_j \setminus G_i$, which is a contradiction. Thus, we must have $d_{G_j}(u,v) = d_{G_i}(u,v)$. □

Theorem 36. *Let G_t be the graph generated at time $t \geq 1$ based on the ILT model with initial graph G_0. If G_0 has an optimum burning sequence $(x_1, x_2, \ldots, x_k)$ in which x_k has a neighbor that is burned in the $(k-1)$-th step, then $b(G_t) = b(G_0)$. Otherwise, $b(G_t) = b(G_0) + 1$.*

Proof. First, assume that $(x_1, x_2, \ldots, x_k)$ is an optimal burning sequence for G_0. Since every node $x' \in V(G_t) \setminus V(G_0)$, with $t \geq 1$, is adjacent to a node in G_0, then $(x_1, x_2, \ldots, x_k)$ is also a burning sequence for the subgraph of G_t induced by

$V(G_t) \smallsetminus (N^{G_t}[x_k] \smallsetminus N^{G_0}[x_k])$. Thus, $b(G_t) \leq b(G_0) + 1$. With a similar argument, we conclude that $b(G_t) \leq b(G_{t-1}) + 1$.

On the other hand, by Lemma 35, we know that G_{t-1} is an isometric subgraph of G_t, for any $t \geq 1$. Also, for any $x \in V(G_{t-1})$ and its clone $x' \in V(G_t)$, we have that $N^{G_t}[x] = N^{G_t}[x']$. Thus, for any $r \geq 1$, $N_r[x'] \cap V(G_{t-1}) = N_r^{G_{t-1}}[x]$. Therefore, by Theorem 15, $b(G_t) \geq b(G_{t-1})$. Hence, by induction we conclude that $b(G_t) \geq b(G_0)$, for any $t \geq 1$, and therefore, we have that either $b(G_t) = b(G_0)$, or $b(G_t) = b(G_0) + 1$. We now characterize where $b(G_t)$ equals $b(G_0)$ or $b(G_t) = b(G_0) + 1$ as follows.

Let $(x_1, x_2, \ldots, x_k)$ be an optimal burning sequence for G_t. By the following algorithm, we find a burning sequence $(y_1, y_2, \ldots, y_k)$ for G_t where at least all the first $k - 1$ fire sources are in G_{t-1}. Note that by above argument, either $b(G_{t-1}) = k$ or $b(G_{t-1}) = k - 1$.

Step 1. If $x_1 \in V(G_{t-1})$, then we take $y_1 = x_1$. Otherwise, we set $y_1 = x$, where x is a node in $V(G_{t-1})$.

Go to Step 2.

Step 2. For each $2 \leq i \leq k - 1$, we do the following steps.

Step 2.1. If $x_i \in V(G_{t-1})$ and x_i is not burned in step $i-1$ by burning $y_1, y_2, \ldots, y_{i-1}$, then we take $y_i = x_i$.

Step 2.2. If $x_i \in V(G_{t-1})$ and x_i is burned in step $i - 1$ by burning $y_1, y_2, \ldots, y_{i-1}$, then we set $y_i = x$, where x is a node in $V(G_{t-1})$ that is not burned in step $i - 1$. We are sure that such a node x exists, since $b(G_{t-1}) \geq k - 1$.

Step 2.3. If x_i is the clone of $x'_i \in V(G_{t-1})$, and x'_i is not burned in step $i - 1$ by burning $y_1, y_2, \ldots, y_{i-1}$, then we take $y_i = x'_i$.

Step 2.4. If x_i is the clone of $x'_i \in V(G_{t-1})$, and x'_i is burned in step $i - 1$ by burning $y_1, y_2, \ldots, y_{i-1}$, then we set $y_i = x$, where x is a node in $V(G_{t-1})$ that is not burned in step $i - 1$. We are sure that such a node x exists, since $b(G_{t-1}) \geq k - 1$.

Go to Step 3.

Step 3. We perform the following steps, and we return the sequence $(y_1, y_2, \ldots, y_k)$ as a burning sequence for G_t.

Step 3.1. If $x_k \in V(G_{t-1})$, and x_k is not burned in step $k - 1$ by burning $y_1, y_2, \ldots, y_{k-1}$, then we take $y_k = x_k$.

Step 3.2. If $x_k \in V(G_{t-1})$, and x_k is burned in step $k-1$ by burning $y_1, y_2, \ldots, y_{k-1}$, then we take $y_k = x$, where x is a node in $V(G_{t-1})$ that is not burned in step $k-1$, if such a node x is available; Otherwise, we choose x to be a node in $V(G_t)$ that is not burned in step $t-1$.

Step 3.3. If x_k is the clone of $x'_k \in V(G_{t-1})$, and x'_k is not burned in step $k-1$ by burning $y_1, y_2, \ldots, y_{k-1}$, then we take $y_k = x'_k$.

Step 3.4. If x_k is the clone of $x'_k \in V(G_{t-1})$, and x'_k is burned in step $k-1$ by burning $y_1, y_2, \ldots, y_{k-1}$, then we set $y_k = x$, where x is a node in $V(G_{t-1})$ that is not burned in step $k-1$, if such a node x is available; Otherwise, we choose x to be a node in $V(G_t)$ that is not burned in step $t-1$.

The sequence $(y_1, y_2, \ldots, y_k)$ obtained by the above algorithm is a burning sequence for G_t. Namely, by burning the nodes in the sequence $(y_1, y_2, \ldots, y_k)$, each node x_i or its clone x'_i is burning at stage i, for $1 \le i \le k$. Therefore, for $1 \le i \le k$, $N^{G_t}_{k-i}[x_i]$ or $N^{G_t}_{k-i}[x'_i]$ is burned by the end of the k-th step. Since for $1 \le i \le k$, $N^{G_t}[x_i] = N^{G_t}[x'_i]$, then we have $N^{G_t}_{k-i}[x_i] = N^{G_t}_{k-i}[x'_i]$. Hence, it implies that

$$\begin{aligned} V(G_t) &\supseteq N^{G_t}_{k-1}[y_1] \cup N^{G_t}_{k-2}[y_2] \cup \ldots \cup N^{G_t}_{0}[y_k] \\ &\supseteq N^{G_t}_{k-1}[x_1] \cup N^{G_t}_{k-2}[x_2] \cup \ldots \cup N^{G_t}_{0}[x_k] \\ &= V(G_t). \end{aligned}$$

Thus, by Equation (2.1), we conclude that $(y_1, y_2, \ldots, y_k)$ is a burning sequence for G_t.

Suppose that for every optimal burning sequence $(x_1, x_2, \ldots, x_k)$ of G_0 all the neighbours of x_k are burned in the k-th step. We claim that $b(G_1) = b(G_0)+1$. Assume not; that is, $b(G_1) = b(G_0)$. Let $(y_1, y_2, \ldots, y_k)$ be an optimal burning sequence for G_1 that is obtained from an optimal burning sequence $(z_1, z_2, \ldots, z_k)$ for G_1 by the algorithm above. Hence, $\{y_1, \ldots, y_k\} \subseteq G_0$. Otherwise, it implies that $b(G_0) = k-1$, which is a contradiction. But, then to burn $y'_k \in V(G_1)$ (the clone of y_k) by the end of the k-th step, one of the nodes in the neighbourhood of y_k must be burned in an earlier stage, which is a contradiction with the assumption. Therefore, in this case $b(G_1) = b(G_0)$ is impossible, and hence, $b(G_1) = b(G_0) + 1$.

Conversely, suppose that $b(G_1) = b(G_0) + 1$, and $(x_1, x_2, \ldots, x_k)$ is an optimal burning sequence for G_0. If x_k has a neighbour that is burned at stage $k-1$, then x'_k is also burned at stage k. Therefore, $(x_1, x_2, \dots, x_k)$ is also a burning sequence for G_1, and we have that $b(G_1) = b(G_0)$, which is a contradiction. Thus, $b(G_1) = b(G_0) + 1$, if and only if for every optimal burning sequence of G_0, say $(x_1, x_2, \dots, x_k)$, all the neighbours of x_k are burned in stage k. By induction, we can conclude that $b(G_t) = b(G_0) + 1$ if and only if for every optimal burning sequence of G_0, say $(x_1, x_2, \dots, x_k)$, all the neighbours of x_k are burned in stage k. Since starting from any graph G_0, for any $t \geq 1$, either $b(G_t) = b(G_0)$, or $b(G_t) = b(G_0) + 1$, we conclude that $b(G_t) = b(G_0)$ if and only if for every optimal burning sequence of G_0, say $(x_1, x_2, \dots, x_k)$ one of the neighbours of x_k is burned at stage $k-1$. □

We finish this section with an example that illustrates Theorem 36. Let P_n be a path on n nodes such that $\lceil n^{1/2} \rceil = k$, for a positive integer k. Then by Theorem 19, we know that $b(P_n) = k$. Moreover, if $(x_1, x_2, \dots, x_k)$ is an optimum burning sequence for P_n, then burning P_n is equivalent to decomposing P_n into paths of orders at most $1, 3, \dots, 2k-1$, in which each path is a rooted path of radius at most $k-i$ and with root x_i, for some $1 \leq i \leq k$. Thus, x_k is the path of order 1 in such a decomposition for P_n in terms of neighbourhoods of x_i's. There are two possibilities for n; either $n = k^2$, or $n \neq k^2$.

If $n = k^2$, then it implies that the order of each path in decomposing P_n is exactly equal to $2(k-i)+1$, for some $1 \leq i \leq k$. Therefore, the end points of such paths are burned in the k-th steps. Hence, both neighbours or the only neighbour of x_k must burn in the k-th step, depending on the position of x_k in P_n. Thus, by Theorem 36, if $G_0 = P_n$ in the ILT model, then we have that $b(G_t) = b(P_n) + 1 = k + 1$, for $t \geq 1$.

On the other hand, if $n \neq k^2$, then there is at least one i for which the rooted path with root x_i is of order less than $2(k-i)+1$. That is, one of the end points of this path called x is not burned at the k-th step. If in decomposing P_n, we choose x_k to be the neighbour of x, then we have a burning sequence for P_n such that at least one of the neighbours of x_k is not burned at step k. Therefore, by Theorem 36, if $G_0 = P_n$ in the ILT model, then we have that $b(G_t) = b(P_n) = k$.

2.4 Graph Products

In this section, we first state two simple bounds on the burning number of the Cartesian product of two graphs in general, and then we find the asymptotic value of the burning number for the Cartesian grids and toroidal grids. We also obtain the asymptotic value of the burning number of hypercube graphs, and we state a conjecture on the exact value of the burning number of hypercube graphs. We then consider the burning number of the strong product of two graphs in general, and the burning number of strong grids in particular. We finish this section by giving some bounds on the burning number of the lexicographic product of two graphs.

Here are some simple bounds on the burning number of the Cartesian product of two graphs.

Theorem 37. *If G and H are two connected graphs, then we have that*

$$\max\{b(G), b(H)\} \leq b(G \,\square\, H) \leq \min\{b(G) + \mathrm{rad}(H), b(H) + \mathrm{rad}(G)\}.$$

Proof. First, note that each of G and H is an isometric subgraph of $G\square H$ that satisfies the conditions in Theorem 15. Namely, suppose that $V(G) = \{u_1, u_2, \ldots, u_m\}$ and $V(H) = \{v_1, v_2, \ldots, v_n\}$ for some positive integers m and n. Let u_k, where $1 \leq k \leq m$, be a central node of G. By definition of the Cartesian product, the subgraph of $G \,\square\, H$ induced by the set of the nodes $\{(u_k, v_i) : 1 \leq i \leq n\}$ is isomorphic to H. We denote this subgraph by H_k. We can see that, for any node (u_i, v_j) in $G \,\square\, H$ and a positive integer r, we have that

$$N_r^{G\square H}[(u_i, v_j)] \cap H_k \subseteq N_r^{H_k}[(u_k, v_j)].$$

Namely, for any node $(u_k, v_l) \in G \,\square\, H$ (where $1 \leq l \leq n$), with $d_{G\square H}\left((u_k, v_l), (u_i, v_j)\right) \leq r$, we have that

$$\begin{aligned} r \geq d_{G\square H}\left((u_k, v_l), (u_i, v_j)\right) &= d_G(u_k, u_i) + d_H(v_l, v_j) \\ &\geq d_H(v_l, v_j) \\ &= d_{G\square H}\left((u_k, v_l), (u_k, v_j)\right). \end{aligned}$$

Hence, H is an isometric subgraph of $G \square H$ that satisfies the conditions in Theorem 15. Similarly, by symmetry of the Cartesian product, we can conclude that G is also an isometric subgraph of $G\square H$ that satisfies the conditions in Theorem 15. Therefore, we conclude that

$$b(G \square H) \geq \max\{b(G), b(H)\}.$$

Now, for proving the upper bound, let $r = \mathrm{rad}(G)$. Also, assume that $b(H) = s$, and $(x_1, x_2, \ldots, x_s)$ is an optimum burning sequence for H. We now define a burning process for $G \square H$ with a burning sequence of length $s + t$, where $t \leq r$ as follows. Suppose that u_k is a central node for G, and H_k is the subgraph of $G \square H$ that is isomorphic to H corresponding to u_k, as defined above. For $1 \leq i \leq s$, let $y_i = (u_k, x_i)$. Thus, $(y_1, y_2, \ldots, y_s)$ forms a burning sequence for H_k. For $i \geq s + 1$, let y_i be a node in $G \square H$ that is not burned by burning the nodes in the sequence $(y_1, y_2, \ldots, y_{i-1})$ in the first $i - 1$ steps, if such a node is available. Since every node in $G \square H$ is within distance r from a node in H_k, then for some $t \leq r$, we will have a burning sequence $(y_1, y_2, \ldots, y_{s+t})$ for $G \square H$. Thus, we conclude that $b(G \square H) \leq s + t \leq s + r$. Similarly, we have that $b(G \square H) \leq b(G) + \mathrm{rad}(H)$. Hence, the proof follows. □

Note that the lower bound in Theorem 37 is achieved by $K_1 \square G$, for any graph G. Also, it is achieved by $K_2 \square P_n$, where $n \in \{k^2 + 1, k^2 + 2\}$, for a positive integer k. The upper bound is tight if G is any graph of radius one and H is a path of square order. For example, let $G = P_3$ and $H = P_4$. Then by Theorem 8, we can show that $b(G \square H) > 2$. On the other hand, by Theorem 37, we have that $b(G \square H) \leq 3$. Therefore, we conclude that $b(P_3 \square P_4) = 3$, which is suggested by the bound in Theorem 37.

The *Cartesian* $m \times n$ *grid* is the Cartesian graph product $P_m \square P_n$. Note that the r-th closed neighbourhood of a node x in the Cartesian grid plane is a *diamond* of radius r with centre x. By direct checking, the number of nodes in such a diamond of radius r equals

$$1 + 4 + \cdots + 4r = 1 + 2r(r + 1). \tag{2.2}$$

We have the following asymptotic result for the burning number of the Cartesian grids.

Theorem 38. *Let G be a $m \times n$ Cartesian grid with $1 \leq m \leq n$, where $m = m(n)$ is a function of n. Then we have that*

$$b(G) = \begin{cases} \Theta(n^{1/2}) & \textit{if } m = O(n^{1/2}) \\ (1+o(1))(3/2)^{1/3}(mn)^{1/3} & \textit{if } m = \omega(n^{1/2}). \end{cases}$$

Proof. First, we find a general lower bound by applying the covering idea in Lemma 3 as follows. Let $(x_1, x_2, \dots, x_k)$ be a burning sequence for G. Thus, every node in G must be in the $(k-i)$-th neighborhood of a node x_i, for some $1 \leq i \leq k$. Note that the r-th closed neighbourhood of a node x in a grid is a subset of a diamond with centre x and with radius r. Hence, by (2.2), we have that $|N_r[x]| = |\{y \in G : d(x,y) \leq r\}| \leq 1 + 2r(r+1)$. Therefore, by Equation (2.1), we conclude that

$$\begin{aligned} mn &= |V(G)| \leq |N_{k-1}[x_1]| + |N_{k-2}[x_2]| + \cdots + |N_0[x_k]| \\ &\leq k + \sum_{i=1}^{k-1} 2i(i+1) = \frac{2k^3+k}{3}. \end{aligned}$$

Let $f(k) = \frac{2k^3+k}{3}$. It follows that for any k with $f(k) < mn$, we can not cover $V(G)$ with any collection of closed neighbourhoods $\{N_{k-i}[x_i]\}_{i=1}^{k}$. Therefore, for such a k, we must have $b(G) > k$. Let $k = (3/2)^{1/3}(mn)^{1/3} - 1$. Then we have that

$$\begin{aligned} f(k) = \frac{2k^3+k}{3} &< \frac{2}{3}(k+1)^3 \\ &= \frac{2}{3}((3/2)^{1/3}(mn)^{1/3})^3 \\ &= mn. \end{aligned}$$

Thus, we conclude that $b(G) > \lceil (3/2)^{1/3}(mn)^{1/3} - 1 \rceil$. Hence, if $m = \omega(n^{1/2})$, then this is the lower bound. Also, we know that the path P_n is an isometric subtree of G. Thus, by Theorem 16, we have that $b(G) \geq n^{1/2}$. This is the lower bound that we need for proving the case $m = O(n^{1/2})$ in the theorem statement.

Now, we prove the upper bounds using the covering idea in Theorem 7 and some technical counting arguments, as follows. By assumption, $m \leq n$. Let $m = \alpha_n n^{1/2}$, for some $1 \leq \alpha_n \leq n^{1/2}$ (note that $\alpha_n \leq n^{1/2}$, since otherwise, we must have $m > n$, which is a contradiction). In the following, we present a way for covering G with some diamonds of successive radii.

Here is a brief description of the idea for defining these strips and diamonds. Let a and b, with $a \geq b$, be two positive integers. Assume that we want to cover the area of a Cartesian grid with a set of diamonds with distinct successive radii, ranging from a to b. We may have trouble to set them up without overlapping. Without any effort to optimize the areas that overlap, playing with the diamonds like a child, a natural way for arranging the diamonds would be putting them diagonally in diagonal strips and ignoring the overlapping parts. This is the idea in the rest of the proof for covering the grid with a set of diamonds. We will present an algorithm that partitions the Cartesian grid G from right to left into some diagonal strips $S_1, S_2, \ldots, S_l$ (for some positive integer l), in which the width of the strips S_i's is increasing. We define the width of each strip S_i inductively. While defining each strip S_i, simultaneously we also cover S_i with disjoint diamonds (from top to bottom), even though they may overlap with the diamonds that are used for covering the other strips.

Algorithm 39. *Let G be the Cartesian $m \times n$ grid with $1 \leq m \leq n$, such that $m = \alpha_n n^{1/2}$, for some $1 \leq \alpha_n \leq n^{1/2}$. Suppose $k_1 = \frac{(mn)^{1/3}}{\alpha_n^{1/6}}$. Then we perform the following steps to cover G with a set of diamonds of successive radii ranging from k_1 to r_l for some integer $r_l \geq k_1$.*

Stage 1. *The rightmost diagonal strip S_1 is a diamond that contains the top right corner of G as its centre, and is of radius k_1 (equivalently of width $\sqrt{2}k_1$). Hence, the top-right part of this strip covers some areas out of G and only a quarter of S_1 falls in G (as we can see in the layout shown in Figure 2.6). The strips are shown Figure 2.6 with colour blue, and the black areas demonstrate the overlaps. Then go to Stage 2.*

Stage 2. *If the top-left corner of G is not covered by S_{i-1}, then we do the following steps; Otherwise go to Stage 3.*

Stage 2.1. *We take u_{i-1} to be the left-most node of S_{i-1}. Let v_{i-1} be the left neighbour of u_{i-1} on the top border line of G. Also, assume that the last diamond that we take for covering S_{i-1} is of radius r_{i-1}. Then we take $r = r_{i-1} + 1$, and we create a diagonal strip, called S_i, of width $\sqrt{2}r$ on the left side of S_{i-1} (parallel to S_{i-1}) such that v_{i-1} is on its border. Note that S_i contains exactly $2r + 1$ nodes of the top border line of G as it is shown in Figure 2.6.*

Stage 2.2. *We cover $V(G) \cap S_i$ from top to bottom with diamonds of successive increasing radiuses ranging from r to $r + j$, for some $j \geq 0$, respectively, such that the left side border of these diamonds is on the left border of S_i, and they all have non-empty intersection with G. The centre of the first (top-most) diamond is on the border line of G, and the last (bottom-most) diamond has non-empty intersection with the bottom boarder line of G.*

Stage 3. *If for some $i \geq 1$, the top-left corner of G is covered by S_{i-1}, and the left bottom corner of G is not covered yet, then we define S_i as follows; otherwise we stop making strips (as at this point, the entire G has been covered by the strips that we have defined till this step).*

Stage 3.1. *We define $r = r_{i-1} + 1$, where r_{i-1} is the radius of the last diamond that is used for covering S_{i-1}. Also, let u_{i-1} be the bottom most node of S_{i-1} that is on the left border line of G, and take v_{i-1} to be its neighbour on the left border line of G below S_{i-1}. We make a diagonal strip, called S_i, of width $\sqrt{2}r$, such that its top most node on the left border line of G is v_{i-1}, and it contains exactly $2r + 1$ nodes on this line. Then we put from top to bottom diamonds of successive increasing radii ranging from r to $r + j$, for some $j \geq 0$ on S_i, such that the centre of the first diamond is on the left border line of G. Also, the left border line of these diamonds is on the left border line of S_i.*

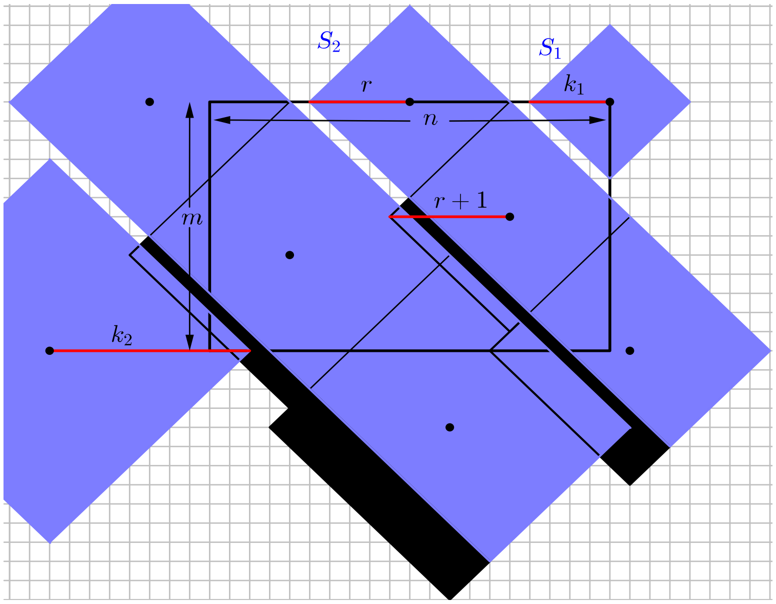


Figure 2.6: A covering of the Cartesian grid.

Suppose that by applying Algorithm 39, we have that $V(G) \subseteq S_1 \cup \ldots \cup S_l$, for some $l \geq k_1$. Let r_l be the radius of the last diamond that we put on S_l.

Thus, we have made a covering for G with diamonds of radii $k_1, k_1 + 1, \ldots, r_l$. We now estimate the number of the nodes in the diamonds defined above that are *wasted*; that is, those nodes that are either placed in the overlapping areas, or fallen out of G in the Cartesian plane. Let S be one of the strips we defined above, and r be the radius of its smallest diamond; that is, the first diamond that we place on S. Note that in the above algorithm, we place those diamonds vertically, from top to bottom. The order of the width of G is m, and the diamonds defined for covering S are of radii $r, \ldots, r + j$, for some $j \geq 0$. Since $(j-1)r \leq m$, then the total number of the diamonds that we put on S, which is $j+1$, is smaller than $\frac{2m}{r}$. On the other hand, the radius of each diamond is increased by 1. Therefore, the radius of the largest diamond we put on S is of order is smaller than $r + \frac{2m}{r}$.

Hence, the area that is overlapped in the largest diamond in S is at most

$$(r + \frac{2m}{r})\frac{2m}{r} = (1 + \frac{2m}{r^2})2m. \tag{2.3}$$

By assumption, we know that $r \geq k_1 = \frac{(mn)^{1/3}}{\alpha_n^{1/6}}$, and $m = \alpha_n \sqrt{n}$. Thus,

$$r \geq \frac{(mn)^{1/3}}{\alpha_n^{1/6}} = \alpha_n^{1/6}\sqrt{n}.$$

Since $\alpha_n \leq \sqrt{n}$, then we conclude that

$$\frac{m}{r} \leq \frac{m}{k_1} = \frac{m}{\alpha_n^{1/6}\sqrt{n}} = \frac{\alpha_n\sqrt{n}}{\alpha_n^{1/6}\sqrt{n}} = \alpha_n^{5/6} \leq n^{5/12}.$$

Therefore, by the above inequalities, we have that

$$0 \leq \frac{m}{r^2} \leq \frac{n^{5/12}}{r} \leq \frac{n^{5/12}}{\alpha_n^{1/6}\sqrt{n}} = \frac{1}{\alpha_n^{1/6}n^{1/12}}.$$

Since $\frac{1}{\alpha_n^{1/6}n^{1/12}}$ tends to zero, as n goes to infinity, we conclude that

$$\lim_{n\to\infty}\frac{m}{r^2} = 0. \tag{2.4}$$

Hence, by (2.3) and (2.4), the area that is overlapped in the largest diamond in S is asymptotically at most $2m$. Therefore, the number of the nodes that are wasted in other diamonds in S is also asymptotically at most $2m$. Thus, the total number of the nodes in S that are overlapped is asymptotically at most

$$2m\left(\frac{2m}{r}\right) = \frac{4m^2}{r}. \tag{2.5}$$

On the other hand, note that in each strip including S, there are at most three diamonds that contains some areas out of G; that is, the first diamond (only half of it) and probably the last two largest ones. Hence, the number of the nodes in covering S with diamonds that fall out of G is asymptotically at most

$$\lim_{n\to\infty} \frac{r^2}{2} + 2(r + \frac{2m}{r})^2 = \lim_{n\to\infty} \frac{r^2}{2} + 2r^2(1 + \frac{2m}{r^2})^2 = \frac{3r^2}{2}. \tag{2.6}$$

Now, we show that $r_l \leq k_2$, where $k_2 = (3/2)^{1/3}(mn)^{1/3}\left(1 + \frac{C}{\alpha_n^{1/6}}\right)$, and C is a large constant to be determined later. By (2.5) and (2.6), we conclude that the total number of the nodes that are wasted in covering S by the diamonds is asymptotically of order at

$$O(r^2) + O(\frac{m^2}{r}) = O(k_2^2) + O(\frac{m^2}{k_1}),$$

since by assumption, $k_1 \leq r \leq k_2$. Note that this bound is independent of S, and applies for all strips. Note that the strips S_i's are disjoint. Also, we put them diagonally on G, such that the diameter of the first diamond in each strip (which is a path of order $2r+1$ for some $r \geq k_1$) is on the top-left border line of G (which is a path of order $n + m \leq 2n$). Therefore, the total number of strips is of order $O(\frac{n}{k_1})$. Thus, for the total number of the nodes that are wasted in covering G by those diamonds, denoted by error$_1$, we have that

$$\text{error}_1 = O(\frac{n}{k_1})\left(O(k_2^2) + O(\frac{m^2}{k_1})\right) = O(\alpha_n^{1/6} n^{3/2}) + O(\alpha_n^{5/3} n).$$

On the other hand, the maximum number of the nodes in G that can be covered by those diamonds with successive radii ranging from k_1 to k_2 equals

$$\begin{aligned}\sum_{i=k_1}^{k_2}(1+2i(i+1)) &= f(k_2+1)-f(k_1)\\ &= \frac{2(k_2+1)^3}{3}+\frac{k_2+1}{3}-\frac{2k_1^3}{3}-\frac{k_1}{3}\\ &> \frac{2}{3}\left((k_2+1)^3-k_1^3\right)\\ &> \frac{2}{3}\left(k_2^3-k_1^3\right)\\ &= \frac{2}{3}\left(\frac{3}{2}mn(1+\frac{C}{\alpha_n^{1/6}})^3-\frac{mn}{\alpha_n^{1/2}}\right)\end{aligned}$$

Since for any positive number x, we have that $(1+x)^3 \geq 1+3x$, then we have that

$$\frac{2}{3}\left(\frac{3}{2}mn(1+\frac{C}{\alpha_n^{1/6}})^3-\frac{mn}{\alpha_n^{1/2}}\right) \geq \frac{2}{3}\left(\frac{3}{2}mn(1+\frac{3C}{\alpha_n^{1/6}})-\frac{mn}{\alpha_n^{1/2}}\right).$$

Therefore, we conclude that

$$\begin{aligned}\sum_{i=k_1}^{k_2}(1+2i(i+1)) &\geq \frac{2}{3}\left(\frac{3}{2}mn(1+\frac{3C}{\alpha_n^{1/6}})-\frac{mn}{\alpha_n^{1/2}}\right)\\ &= mn+\frac{2}{3}\left(\frac{9C}{2\alpha_n^{1/6}}mn-\frac{mn}{\alpha_n^{1/2}}\right)\\ &\geq mn+\frac{2}{3}\left(\frac{9C}{2\alpha_n^{1/6}}mn-\frac{mn}{\alpha_n^{1/6}}\right)\\ &= mn+\frac{9C-2}{3\alpha_n^{1/6}}mn\\ &= mn+\frac{9C-2}{3}n^{3/2}\alpha_n^{5/6}.\end{aligned}$$

We denote the term $\frac{9C-2}{3}n^{3/2}\alpha_n^{5/6}$ in the above equality by error$_2$. Hence, by this notation, the number of the nodes in G that can be covered by those diamonds is at least $mn+\text{error}_2$. We will show that by choosing an appropriate value for C, we can have error$_2 \geq$ error$_1$, and consequently, we can cover all the mn nodes in G. This implies that k_2 is an upper bound on r_l. If $m=\omega(\sqrt{n})$, then α_n must be a function of n that goes to infinity, as n grows (But still $\alpha_n \leq \sqrt{n}$).

Hence, by taking $C = 1$, we have that $\text{error}_1 = o(\text{error}_2)$. Thus, we showed that covering G by at most $t+1 = k_2 - k_1 + 1$ diamonds with distinct radii ranging from k_1 to k_2, is possible. Therefore, by Theorem 6, we conclude that $b(G) \leq k_2 + 1$. Thus, if $m = \omega(\sqrt{n})$, then we have that

$$b(G) = (1 + o(1))(\frac{3}{2})^{1/3}(mn)^{1/3}.$$

If $m < c\sqrt{n}$, where c is a constant, then by Theorem 19, we burn the path P_n on the top border of G in $\lceil\sqrt{n}\rceil$ steps. Since in this case, every node in G is within distance $c\sqrt{n}$ from a node in P_n, after at most $c\sqrt{n}$ more steps all the nodes in G must be burned. Therefore, we have that $b(G) = O(\sqrt{n})$. Thus, in this case $b(G) = \Theta(\sqrt{n})$, and the proof follows. □

By a $m \times n$ *toroidal grid* we mean the Cartesian product of C_m and P_n. Note that a $m \times n$ Cartesian grid is a spanning subgraph of a $m \times n$ toroidal grid. Also, as in the Cartesian grid plane, the r-th closed neighbourhood of a node x in a toroidal grid is a diamond with centre x and of radius r. Hence, we have the following corollary.

Corollary 40. *Let G be a $m \times n$ toroidal grid with $1 \leq m \leq n$, where $m = m(n)$ is a function of n. Then we have the following asymptotic results:*

$$b(G) = \begin{cases} \Theta(n^{1/2}) & \textit{if } m = O(n^{1/2}) \\ (1 + o(1))(3/2)^{1/3}(mn)^{1/3} & \textit{if } m = \omega(n^{1/2}). \end{cases}$$

Proof. For proving the upper bound, we just use the fact that a $m \times n$ Cartesian grid is a spanning subgraph of a $m \times n$ toroidal grid. To prove the lower bound, note that we can have exactly the same argument as in proof of Theorem 38 without any adjustment, since the whole argument was only based on the covering technique and counting the number of the nodes in each closed neighbourhood, which will be a subset of a diamond. Thus, the proof follows. □

The *hypercube graph*, or the *n-cube*, or the *n-dimensional hypercube*, denoted by Q_n, is a graph of order 2^n in which every node is corresponding to a binary string of length n, and two nodes are adjacent if and only if their corresponding binary strings differ in exactly one bit. It is known in graph theory that

$$Q_n = Q_{n-1} \square Q_1$$

where Q_1 is the path on two nodes. We have the following conjecture for the optimum burning of the hypercube graphs.

Conjecture 41. *Let n be a positive integer, and $k = \lceil \frac{n}{2} \rceil + 1$. Then for the hypercube Q_n we have that $b(Q_n) = k$. Moreover, if n is even, then in any optimal burning sequence $(x_1, x_2, \ldots, x_k)$ for Q_n we must have $d(x_1, x_2) = n = \operatorname{diam}(Q_n)$.*

Note that if we choose x_1 and x_2 as two nodes with $d(x_1, x_2) = n$, then every node in Q_n will be burned after $k = \lceil \frac{n}{2} \rceil + 1$ steps. Therefore, we have that $b(Q_n) \leq \lceil \frac{n}{2} \rceil + 1$. However, it seems challenging to prove the truth of the lower bound for the burning number of the n-cube. Note that Q_1 is a path of order 2, and by Theorem 19, we know that $b(Q_1) = b(P_2) = \lceil \sqrt{2} \rceil = 2$. Since $2 = \lceil \frac{1}{2} \rceil + 1$, the conjecture is true for Q_1. Also, Q_2 is a 4-cycle, and by Corollary 75, we know that $b(Q_2) = b(C_4) = \lceil \sqrt{4} \rceil = 2$. Since $2 = \lceil \frac{2}{2} \rceil + 1$, then we can see the truth of the conjecture for Q_2 as well.

Using an approximation in terms of the well-known *binary entropy function*, we can find a weaker lower bound on the burning number of the hypercube graph Q_n as follows. This lower bound leads to an asymptotic result for the burning number of the hypercube graph Q_n. The binary entropy function is defined by the rule:

$$H(x) = -x \log_2 x - (1 - x) \log_2 (1 - x),$$

where $0 < x \leq 1$ is a real number. The Taylor series of the entropy function is known to be

$$H(x) = 1 - \frac{1}{2 \ln 2} \sum_{n=1}^{\infty} \frac{(1 - 2x)^{2n}}{(2n - 1)},$$

for any x with $0 < x \leq 1$. It is also known that for an integer $1 \leq k \leq n$, if $\epsilon = \frac{k}{n}$, and $\epsilon \leq \frac{1}{2}$, then

$$\sum_{i=0}^{k} \binom{n}{i} \leq 2^{H(\epsilon) \cdot n};$$

see [28]. Here are the bounds that we can prove on the burning number of n-cubes.

Theorem 42. *For the hypercube graph Q_n, we have that*

$$b(Q_n) = (1 + o(1)) \left(\left\lceil \frac{n}{2} \right\rceil + 1 \right).$$

Proof. We will prove this by showing that

$$\left\lceil \frac{n}{2} + 1 - \sqrt{\ln 2 \left(\frac{n}{2} \log_2 \frac{n}{2} \right)} \right\rceil \leq b(Q_n) \leq \left\lceil \frac{n}{2} \right\rceil + 1. \tag{2.7}$$

As we mentioned earlier, by burning two nodes such as $x_1, x_2 \in V(Q_n)$ with $d(x_1, x_2) = n$ in the first and second steps of a burning process for Q_n, we will have every node burning at time $t = \lceil \frac{n}{2} \rceil + 1$. Therefore, we know that $b(Q_n) \leq \lceil \frac{n}{2} \rceil + 1$. On the other hand, if $b(Q_n) \geq \lceil \frac{n}{2} \rceil + 1$, then we conclude that $b(Q_n) = \lceil \frac{n}{2} \rceil + 1$, and theorem statement is clearly true in this case. Hence, without loss of generality assume that $b(Q_n) = k \leq \lceil \frac{n}{2} \rceil$, for some positive integer k. We consider this case as follows.

Since for $n \geq 1$, Q_n is of order at least two, then it implies that $b(Q_n) = k \geq 2$. Note that for a fixed node $u \in V(Q_n)$, the nodes that are at distance i from u in Q_n are those nodes whose corresponding n-binary strings differ in exactly i bits from the corresponding n-binary string of node u. Thus, the number of nodes in Q_n that are at distance i from node u equals $\binom{n}{i}$. Hence, the number of nodes in the r-th closed neighbourhood of u in Q_n equals

$$\sum_{i=0}^{r} \binom{n}{i}.$$

Now, suppose that $(x_1, x_2, \ldots, x_k)$ is an optimum burning sequence for Q_n. By Equation (2.1), we have that

$$\begin{aligned} 2^n = |V(Q_n)| &\leq \sum_{i=1}^{k} |N_{k-i}[x_i]| \\ &= \sum_{i=1}^{k} \sum_{j=0}^{k-i} \binom{n}{j} \\ &= \sum_{i=0}^{k-1} (k-i) \binom{n}{i}. \end{aligned}$$

By assumption, we know that $2 \leq k \leq \lceil \frac{n}{2} \rceil$. Thus, we have that $1 \leq k-1 \leq \frac{n}{2}$. Therefore, we conclude that $0 \leq \frac{k-1}{n} \leq \frac{1}{2}$. Hence, by the above inequalities, and the facts about the binary entropy function that we mentioned earlier, we have that

$$\begin{aligned}2^n &\leq \sum_{i=0}^{k-1}(k-i)\binom{n}{i}\\ &\leq \sum_{i=0}^{k-1}(k-1)\binom{n}{i}\\ &= (k-1)\sum_{i=0}^{k-1}\binom{n}{i}\\ &\leq (k-1)2^{H(\epsilon)\cdot n},\end{aligned}$$

where $\epsilon = \frac{k-1}{n}$. Thus, if $b(Q_n) = k$, then we must have that $(k-1)2^{H(\epsilon)\cdot n} \geq 2^n$. Therefore, we must have $k-1 \geq 2^{n(1-H(\epsilon))}$. On the other hand, by the Taylor series of the entropy function, we know that

$$1 - H(\epsilon) = 1 - 1 + \frac{1}{2\ln 2}\sum_{n=1}^{\infty}\frac{(1-2\epsilon)^{2n}}{(2n-1)} \geq \frac{(1-2\epsilon)^2}{2\ln 2}.$$

Hence, we must have that

$$k - 1 \geq 2^{n(1-H(\epsilon))} \geq 2^{n\frac{(1-2\epsilon)^2}{2\ln 2}}.$$

Since by assumption, $\epsilon = \frac{k-1}{n}$, then $k-1 = \epsilon n$. By substitution of this in the above inequality, we conclude that $\epsilon n \geq 2^{n\frac{(1-2\epsilon)^2}{2\ln 2}}$. By taking the logarithm from both sides of this inequality, we have that

$$\log_2 n\epsilon \geq n\frac{(1-2\epsilon)^2}{2\ln 2}.$$

Since $\epsilon \leq \frac{1}{2}$, then $\log_2 n\epsilon \leq \log_2 \frac{n}{2}$. Therefore, we have that

$$\log_2 \frac{n}{2} \geq \log_2 n\epsilon \geq n\frac{(1-2\epsilon)^2}{2\ln 2}.$$

Thus, we derive that

$$(1-2\epsilon)^2 \leq \frac{2\ln 2\log_2 \frac{n}{2}}{n}.$$

This implies that

$$\epsilon \geq \frac{1}{2} - \frac{1}{2}\sqrt{\frac{2\ln 2 \log_2 \frac{n}{2}}{n}}.$$

Since $\epsilon = \frac{k-1}{n} \leq \frac{1}{2}$, then we conclude that

$$k - 1 \geq \frac{n}{2} - \frac{n}{2}\sqrt{\frac{2\ln 2 \log_2 \frac{n}{2}}{n}} = \frac{n}{2} - \sqrt{\frac{\ln 2}{2} n \log_2 \frac{n}{2}}.$$

Therefore, if $b(Q_n) = k$, then we must have that

$$k \geq \left\lceil \frac{n}{2} + 1 - \sqrt{\ln 2\left(\frac{n}{2}\log_2 \frac{n}{2}\right)} \right\rceil.$$

Hence, (2.7) is true. Since $\lim_{n\to\infty} \frac{\sqrt{\ln 2(\frac{n}{2}\log_2 \frac{n}{2})}}{\frac{n}{2}} = 0$, we derive the desired result from the inequalities in (2.7). □

Now, we consider the burning number of the strong product of graphs. We first present some simple general bounds similar to Theorem 37.

Theorem 43. *If G and H are two connected graphs, then we have that*

$$\max\{b(G), b(H)\} \leq b(G \boxtimes H) \leq \min\{b(G) + \text{rad}(H), b(H) + \text{rad}(G)\}.$$

Proof. First, note that each of G and H is an isometric subgraph of $G \boxtimes H$ that satisfies the conditions in Theorem 15. Namely, suppose that $V(G) = \{u_1, u_2, \ldots, u_m\}$ and $V(H) = \{v_1, v_2, \ldots, v_n\}$ for some positive integers m and n. Let u_k, where $1 \leq k \leq m$, be a node in G. By definition of the strong product, the subgraph of $G \boxtimes H$ induced by the set of the nodes $\{(u_k, v_i) : 1 \leq i \leq n\}$ is isomorphic to H. We denote this subgraph by H_k. We can see that, for any node (u_i, v_j) in $G \boxtimes H$ and a positive integer r, we have that

$$N_r^{G\boxtimes H}[(u_i, v_j)] \cap H_k \subseteq N_r^{H_k}[(u_k, v_j)].$$

Namely, for any node $(u_k, v_l) \in G \boxtimes H$ (where $1 \leq l \leq n$), with $d_{G\boxtimes H}((u_k, v_l), (u_i, v_j)) \leq$

r, we have that

$$\begin{aligned} r \geq d_{G\boxtimes H}\left((u_k,v_l),(u_i,v_j)\right) &= \max\{d_G(u_k,u_i), d_H(v_l,v_j)\} \\ &\geq d_H(v_l,v_j) \\ &= d_{G\boxtimes H}\left((u_k,v_l),(u_k,v_j)\right). \end{aligned}$$

Hence, H is an isometric subgraph of $G \boxtimes H$ that satisfies the conditions in Theorem 15. Similarly, by symmetry of the Cartesian product, we can conclude that G is also an isometric subgraph of $G\boxtimes H$ that satisfies the conditions in Theorem 15. Therefore, we conclude that

$$b(G \boxtimes H) \geq \max\{b(G), b(H)\}.$$

For proving the upper bound, note that $G \square H$ is a spanning subgraph of $G \boxtimes H$. Thus, by Theorem 37, we conclude that

$$b(G \boxtimes H) \leq b(G \square H) \leq \min\{b(G) + \text{rad}(H), b(H) + \text{rad}(G)\}.$$

Hence, the proof follows. $\square$

We now consider the burning number of the strong grids. By a *$m \times n$ strong grid* we mean the strong product of P_m and P_n. Figure 2.7 shows an example of a 3×3 strong grid. Note that the r-th closed neighbourhood of a node x in a strong grid is a small strong subgrid which is a subset of a $(2r+1) \times (2r+1)$ strong grid with centre x. By direct checking, we can see that the number of the nodes in a $(2r+1) \times (2r+1)$ strong grid equals $(2r+1)^2$. We have the following theorem about the burning number of the strong grids.

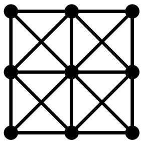


Figure 2.7: A 3×3 strong grid.

Theorem 44. *Let G be a $m \times n$ strong grid with $1 \leq m \leq n$, where $m = m(n)$ is a* *function of n. Then we have the following asymptotic results:*

$$b(G) = \begin{cases} \Theta(n^{1/2}) & \text{if } m = O(n^{1/2}) \\ \Theta((mn)^{1/3}) & \text{if } m = \omega(n^{1/2}). \end{cases}$$

Proof. Note that a $m \times n$ strong grid such as G contains a $m \times n$ Cartesian grid H as its spanning subgraph. Hence, we have that $b(G) \leq b(H)$, and we can apply Theorem 38 for finding upper bounds. That is, if $m = O(n^{1/2})$, then $b(G) = O(n^{1/2})$; while if $m = \omega(n^{1/2})$, then $b(G) = O((mn)^{1/3})$.

Now, we find a general lower bound by applying the covering idea in Lemma 3 as follows. Let $(x_1, x_2, \dots, x_k)$ be a burning sequence for G. Thus, every node in G must be in the $(k-i)$-th neighborhood of a node x_i, for some $1 \leq i \leq k$. Therefore, by Equation (2.1), we conclude that

$$\begin{aligned} mn = |V(G)| &\leq |N_{k-1}[x_1]| + |N_{k-2}[x_2]| + \cdots + |N_0[x_k]| \\ &\leq \sum_{i=1}^{k} (2i-1)^2 \\ &= \frac{k(2k-1)(2k+1)}{3}. \end{aligned}$$

Let $f(k) = \frac{k(2k-1)(2k+1)}{3}$. Then it implies that for any k with $f(k) < mn$, we can not cover $V(G)$ with any collection of closed neighbourhoods such as $\{N_{k-i}[x_i]\}_{i=1}^{k}$. Therefore, for such a k, we must have $b(G) \geq k$. Let $k = \lfloor (3/4)^{1/3}(mn)^{1/3} \rfloor$. Then we have that

$$\begin{aligned} f(k) = \frac{4k^3 - k}{3} &< \frac{4}{3}k^3 \\ &\leq \frac{4}{3}((3/4)^{1/3}(mn)^{1/3})^3 \\ &= mn. \end{aligned}$$

Hence, we conclude that $b(G) > \lceil (3/4)^{1/3}(mn)^{1/3} \rceil$. Also, we know that the path P_n is an isometric subtree of G. Thus, by Theorem 16, we have that $b(G) \geq n^{1/2}$. Therefore, the proof follows. □

We now present bounds on the burning number of the lexicographic product of two graphs in general.

Theorem 45. *For any two graphs G and H we have that*

$$b(G) \le b(G \cdot H) \le b(G) + 2.$$

Proof. Suppose that $V(G) = \{u_1, u_2, \ldots, u_n\}$, and $V(H) = \{v_1, v_2, \ldots, v_m\}$, where m and n are two positive integers. Note that the subgraph of $G \cdot H$ induced by the nodes $\{(u_1, v_1), (u_2, v_1), \ldots, (u_n, v_1)\}$ is isomorphic to G. We call this subgraph G_1, and we claim that G_1 is an isometric subgraph of $G \cdot H$ that satisfies the conditions in Theorem 15. Namely, suppose that (u_i, v_1) and (u_j, v_1) are two nodes in G_1, where $1 \le i < j \le n$. As we mentioned in Section 1.2, the graph distance between the nodes (u_i, v_1) and (u_j, v_1) in the graph $G \cdot H$ equals $d_G(u_i, u_j)$. Hence, G_1 is an isometric subgraph of $G \cdot H$. Now, suppose that (u_i, v_j) is a node in $G \cdot H$, with $j \ne 1$, and r is a positive integer. We have the following possibilities for $N_r^{G \cdot H}[(u_i, v_j)] \cap G_1$. If $r = 1$ and $v_j \in N(v_1)$, then $N_r^{G \cdot H}[(u_i, v_j)] \cap G_1 = N_r^{G_1}((u_i, v_1)) \subset N_r^{G_1}[(u_i, v_1)]$. If $r \ge 2$ or $v_j \notin N(v_1)$, then $N_r^{G \cdot H}[(u_i, v_j)] \cap G_1 = N_r^{G_1}[(u_i, v_1)]$. Thus, the claim is true. Therefore, by Theorem 15, we conclude that

$$b(G \cdot H) \ge b(G_1) = b(G).$$

Now, for proving the upper bound, assume that $b(G) = k$. Let $(x_1, x_2, \ldots, x_k)$ be an optimum burning sequence for G. We now define a burning process for $G \cdot H$ with a burning sequence of length $k + t$, where $t \le 2$ as follows. For $1 \le i \le k$, take $y_i = (x_i, v_1)$. Thus, $(y_1, y_2, \ldots, y_k)$ forms a burning sequence for G_1. For $i \ge k + 1$, let y_i be a node in $G \cdot H$ that is not burned by burning the nodes in the sequence $(y_1, y_2, \ldots, y_{i-1})$ in the first $i - 1$ steps, if such a node is available. Since every node (u_i, v_j) in $G \cdot H$ is within distance 2 from the node (u_i, v_1) in G_1, then we will have a burning $(y_1, y_2, \ldots, y_{k+t})$ for $G \cdot H$, for some $t \le 2$. Hence, we conclude that $b(G \cdot H) \le b(G) + 2$. □

Chapter 3

Complexity of the Graph Burning Problem

In this chapter, we consider the computational complexity of the burning problem.

3.1 Trees with Maximum Degree Three

In this section, we prove that the decision version of the burning problem is **NP**-complete even for trees with maximum degree three. We show this by a reduction from a variant of the 3-Partition problem which is also known to be strongly **NP**-complete. Here is the statement of this problem.

Problem: Distinct 3-Partition
Instance: A finite set $X = \{a_1, a_2, \ldots, a_{3n}\}$ of distinct positive integers, and a positive integer B where $\sum_{i=1}^{3n} a_i = nB$, and $B/4 < a_i < B/2$, for $1 \leq i \leq 3n$.
Question: Is there any partition of X into n triples such that in each triple the elements add up to B?

Note that in the original 3-Partition problem a_i's are not necessarily distinct; that is, X is a multi-set. However, in the Distinct 3-Partition problem, X is a set of distinct integers. A proof for the **NP**-completeness of the Distinct 3-Partition problem which is shown in fact to be strongly **NP**-complete is given in Corollary 7 from [36].

In the following, we show the **NP**-completeness of the Graph Burning problem by a reduction from Distinct 3-Partition. Here is the statement of the decision version of the burning problem.

Problem: Graph Burning
Instance: A finite simple graph G with $V(G) = \{v_1, v_2, \ldots, v_n\}$, and an integer $k \geq 2$.
Question: Is $b(G) \leq k$?

Note that the only graph with burning number 1 is the complete graph K_1. Also, by Theorem 8 in Chapter 2, we know that for a graph G of order n, $b(G) = 2$ if and only if $\Delta(G) \in \{n-1, n-2\}$. Since finding the maximum degree of a graph is a polynomial time problem, then the burning problem for $k = 2$ is in **P**. Now, we prove that the burning problem for $k \geq 3$ is **NP**-complete even for trees of maximum degree three, as follows.

A spider graph is a tree with only one node of degree at least three. Such a node in a spider graph is called the *spider head.* In the rest of this chapter, by O_i we mean the set of odd integers $\{1, 3, \ldots, 2i - 1\}$.

Theorem 46. *The Graph Burning problem is* ***NP****-complete for trees of maximum degree three.*

Proof. First, we need to show that the burning problem is in **NP**. Given a graph G of order n and a sequence $(x_1, x_2, \ldots, x_k)$ of the nodes in G, we can find $N_{k-i}[x_i]$ in polynomial time, for $1 \leq i \leq k$. Thus, we can check in polynomial time if $V(G) = \bigcup_{i=1}^{k} N_{k-i}[x_i]$. Hence, the burning problem is in **NP**.

Now, we show the **NP**-completeness of the burning problem for trees of maximum degree 3 by a reduction from the Distinct 3-Partition problem, as follows.

Suppose that we have an instance of the Distinct 3-Partition problem; that is, we are given a non-empty finite set $X = \{a_1, a_2, \ldots, a_{3n}\}$ of distinct positive integers, and a positive integer B such that $\sum_{i=1}^{3n} a_i = nB$, and $B/4 < a_i < B/2$, for $1 \leq i \leq 3n$. Since the Distinct 3-Partition problem is **NP**-complete in the strong sense, without loss of generality we can assume that B is bounded above by a polynomial in the length of the input. Assume that the maximum of the set X is m which is by assumption bounded above by $B/2$. We now construct a tree of maximum degree 3 as follows.

Let $Y = \{2a_i - 1 : 1 \leq i \leq 3n\}$. Hence, $Y \subseteq O_m$, and $2nB - 3n = \sum_{i=1}^{3n} 2a_i - 1$ is the sum of the numbers in Y. Let $Z = O_m \setminus Y$. Note that $1 \leq |Y| \leq m$, and consequently, $|Z| \leq m - 1$. Let $|Z| = k$, for some $k \leq m - 1$. For $1 \leq i \leq k$, let Q'_i be the path of order l_i, where l_i is the i-th largest element in Z. For $1 \leq i \leq m + 1$, we define T_i to be a spider $SP(3, 2m + 1 - i)$ with centre r_i. We also take Q_i to be a path of order $2B - 3$, for $1 \leq i \leq n$. Then we combine the graphs that we created above from left to right in the following order:

$$Q_1, T_1, Q_2, T_2, \ldots, Q_n, T_n, Q'_1, T_{n+1}, Q'_2, T_{n+2}, \ldots, Q'_k, T_{n+k}, T_{n+k+1}, \ldots, T_{m+1}$$

such that each graph in this order is joined by an edge from one of its leaves to a leaf of the next graph in the presented order. The resulting graph is called $T(X)$; note that it is a tree of maximum degree three.

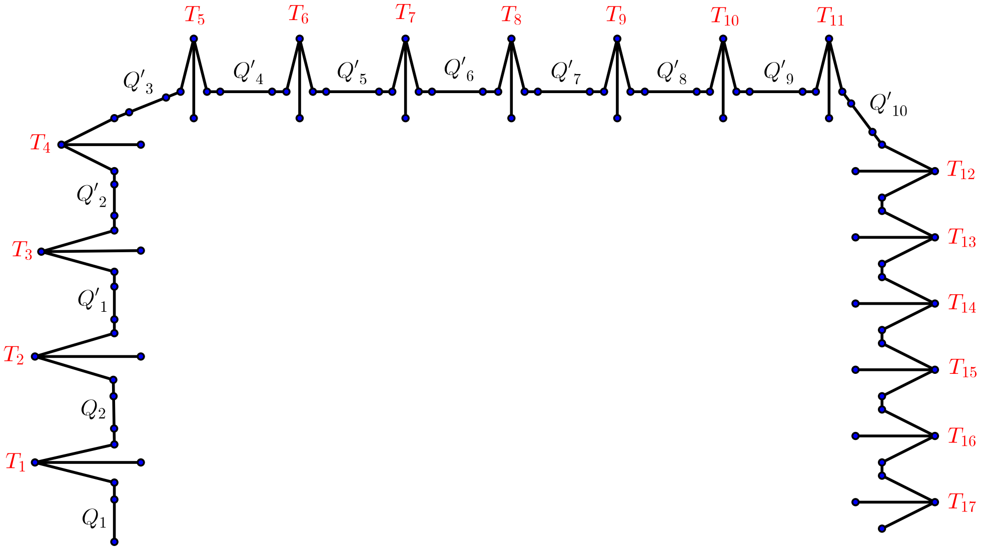


Figure 3.1: A sketch of the tree $T(X)$.

For example, let $X = \{10, 11, 12, 14, 15, 16\}$, and $B = 39$. Then $n = 2$, and $m = max\{a_i : a_i \in X\} = 16$. Therefore, $Y = \{19, 21, 23, 27, 29, 31\}$, and $Z = O_{16} \setminus Y = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 25\}$. Thus, $k = |Z| = |O_{16} \setminus Y| = 10$. The graph $T(X)$ is depicted in Figure 3.1. For simplicity, we do not draw the nodes in the paths Q_i and Q'_j, and the spiders T_i's in the figure.

For $1 \leq i \leq m+1$, let v_i be a leaf of $T(X)$ that is also a leaf of T_i, as a subgraph of $T(X)$. Note that for $1 \leq i \leq m+1$ the two arms of T_i that do not contain v_i, together with its centre r_i, form a path. We call this path T'_i. The order of T'_i is $2(2m+1-i)+1 \in O_{2m+1}$. Also, note that $\sum_{i=1}^{n} |Q_i| = \sum_{a_i \in X} 2a_i - 1$, where $|Q_i|$ is the order of each path Q_i. Moreover, $\sum_{i=1}^{n} |Q_i| + \sum_{i=1}^{k} |Q'_i| = \sum_{i=1}^{m} 2i - 1$. Hence, the subgraph of $T(X)$ induced by

$$\left(\bigcup_{i=1}^{m+1} T'_i\right) \bigcup \left(\bigcup_{i=1}^{n} Q_i\right) \bigcup \left(\bigcup_{i=1}^{k} Q'_i\right)$$

forms a path of order

$$\sum_{i=1}^{m}(2i-1) + \sum_{i=1}^{m+1} 2(2m+1-i) + 1 = \sum_{i=1}^{2m+1} 2i - 1 = (2m+1)^2,$$

that we denote it by P. Therefore,

$$T(X) - P = \bigcup_{i=1}^{m+1} (T_i \setminus P),$$

which is a disjoint union of paths of orders $\{2m+1-i\}_{i=1}^{m+1}$. Note that $T_i \setminus P$ is the arm of T_i that contains v_i. Thus, we have that

$$|V(T(X))| = (2m+1)^2 + \sum_{i=1}^{m+1} (2m+1-i) = (2m+1)^2 + \frac{3(m^2+m)}{2}.$$

Since m is bounded by B and by assumption, B is bounded above by a polynomial in the length of the input, then $T(X)$ is obtained in polynomial time in the length of the input.

We can see that, there is a partition of X into triples such that the elements in each triple add up to B if and only if we can decompose the paths $Q_1, Q_2, \ldots, Q_n$ into subpaths of orders $2a_i - 1 \in Y$. First, assume that there is a partition of X into triples such that the elements in each triple add up to B. Equivalently, we have a partition for the paths $Q_1, Q_2, \ldots, Q_n$ in terms of subpaths $\{P_l : l \in Y\}$. Hence, there is a partition for the subgraph $(\bigcup_{i=1}^{n} Q_i) \bigcup (\bigcup_{i=1}^{k} Q'_i)$ in terms of the subpaths $\{P_l : l \in O_m\}$. For $m+2 \le i \le 2m+1$, let x_i be the centre of a path P_l of order $l = 2(2m+2-i) - 1 \in O_m = X \cup Y$ in such a partition. For $1 \le i \le m+1$, let $x_i = r_i$ (the centre of T_i). Thus, we have that

$$V(T(X)) = \bigcup_{i=1}^{2m+1} N_{2m+1-i}[x_i].$$

Consequently, by Equation (2.1), we conclude that $(x_1, x_2, \ldots, x_{2m+1})$ forms a burning sequence of length $2m+1$ for $T(X)$. Therefore, $b(T(X)) \le 2m+1$.

Conversely, suppose that $b(T(X)) \le 2m+1$. Note that the path P of order $(2m+1)^2$ is a subtree of $T(X)$. Therefore, by Theorem 19 and Corollary 18, we have that

$$b(T(X)) \ge b(P) = 2m+1.$$

Thus, we conclude that $b(T(X)) = 2m+1$. Assume that $(x_1, x_2, \ldots, x_{2m+1})$ is an optimum burning sequence for $T(X)$.

We claim that each x_i must be in P. First, since $T(X) = P \bigcup(\bigcup_{j=1}^{m+1}(T_j \setminus P))$, every x_i is either in P or in $T_j \setminus P$, for some $1 \le j \le m+1$. On the other hand, every node in P must receive the fire from one of the x_i's. Note that the only connection of P to $T(X) \setminus P$ is through the nodes r_i's. Hence, for $1 \le i \le 2m+1$, $N_{2m+1-i}[x_i] \cap P$ must be a path of order at most $2(2m+1-i)+1$. If for some $1 \le i \le 2m+1$, the node x_i is out of P, then $N_{2m+1-i}[x_i] \cap P$ will be a path of order less than $2(2m+1-i)+1$. Therefore, the total sum of the orders of the subpaths $\{N_{2m+1-i}[x_i] \cap P\}_{i=1}^{2m+1}$ will be less than $(2m+1)^2 = |V(P)|$, which is a contradiction. Thus, every x_i must be selected from P.

Now, we claim that for $1 \le i \le m+1$, we must have $x_i = r_i$. We prove this by strong induction on i. Note that for $1 \le i \le m+1$, each v_i receives the fire from a fire source $x_j \in P$ (by the above argument), where $1 \le j \le 2m+1$. Therefore, $d(x_j, v_i) \le 2m+1-j$, for some $1 \le j \le 2m+1$. For $i = 1$, since the only node in P that is within distance $2m+1-i = 2m$ from v_1 is r_1, then we must have $x_1 = r_1$. Suppose that for $1 \le i \le m$ and for every $1 \le j \le i$, $x_j = r_j$. Since the only node in P within distance $2m+1-(i+1)$ from v_{i+1} is the node r_{i+1}, and by induction hypothesis, we conclude that $x_{i+1} = r_{i+1}$. Therefore, the claim is proved by induction.

Note that $N_{2m+1-i}[r_i] = V(T_i)$. Therefore, for $1 \le i \le m+1$, the fire started at $x_i = r_i$ will burn all the nodes in T_i. Hence, without loss of generality, we can assume that $x_i \in T(X) \setminus \bigcup_{i=1}^{m+1} T_i$, for $m+2 \le i \le 2m+1$. Now, the above argument implies that the nodes in $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ must be burned by receiving the fire started at $x_{m+2}, x_{m+3}, \dots, x_{2m+1}$ (the last m sources of fire). Since $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ is a disjoint union of paths, then we derive that for $m+2 \le i \le 2m+1$, $N_{2m+1-i}[x_i] \bigcap(T(X) \setminus \bigcup_{i=1}^{m+1} T_i)$ is a path of order at most $2m+1-i$ $(\le 2m-1)$. On the other hand, the path-forest $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ is of order

$$\sum_{i=1}^{m}(2m+1-i) = m^2.$$

Thus, we conclude that for $m+2 \le i \le 2m+1$, $N_{2m+1-i}[x_i] \bigcap(T(X) \setminus \bigcup_{i=1}^{m+1} T_i)$ is a path of order equal to $2m+1-i$; since otherwise, we can not burn all the nodes in $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ in m steps, which is a contradiction. Therefore, there must be a partition of $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ (induced by the burning sequence $(x_{m+2}, x_{m+3}, \dots, x_{2m+1})$) into subpaths $\{P_l : l \in O_m\}$.

Now, considering the partition described in the previous paragraph, we claim that there is a partition of $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ into subpaths of orders in O_m in which the paths $Q_1, Q_2, \ldots, Q_n$ are decomposed into paths of orders in Y, and each path Q'_i is covered by itself. Note that by definition, for $1 \leq i \leq k$, each path Q'_i is a component of $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$. Hence, it suffices to prove that there is a partition of $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ into subpaths of orders in O_m such that each Q'_i is covered by itself. Assume that in a partition of $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ into subpaths of orders in O_m, there is a path Q'_i of order $l \in O_m \setminus Y = Z$ that is partitioned by a union of paths of orders in O_m rather than by P_l itself. We know that P_l must have covered some part of a path Q'_j with $j \neq i$, or must be used in partitioning $Q_1, Q_2, \ldots, Q_n$. Hence, we can modify the partition by switching the place of P_l and those paths that have covered P_l (as they have equal lengths). Therefore, we have decreased the number of such displaced paths in our partition for $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$. Since the number of Q'_i's, where $1 \leq i \leq k$, is finite, we will end up after finite number of switching in a partition for $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ in which every Q'_i, $1 \leq i \leq k$, is covered by itself.

In the partition described above, every Q_i $(1 \leq i \leq n)$ is decomposed into paths of orders $\{l : l \in Y\}$. Since each Q_i is of order $2B-3$, there must be a partition of Y into triples such that the elements in each triple add up to $2B-3$. Equivalently, there must be a partition of X into triples such that the elements in each triple add up to B. Since $T(X)$ is a tree of maximum degree 3, then we have a polynomial time reduction from the Distinct 3-Partition problem to the Graph Burning problem for trees with maximum degree 3. □

Since any tree is a chordal graph, and also planar and bipartite, then we conclude the following corollary.

Corollary 47. *The burning problem is* ***NP****-complete for chordal graphs, planar graphs, and bipartite graphs.*

We have the following immediate corollary since a tree of maximum degree three is a graph of maximum degree three.

Corollary 48. *The burning problem is* ***NP****-complete even for graphs of maximum degree three.*

We also have the following corollary.

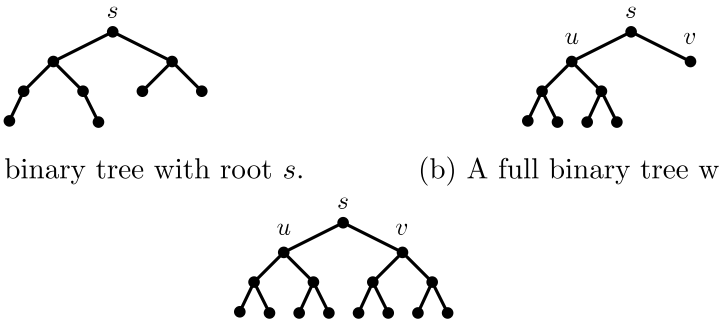
Corollary 49. (i) *The burning problem is **NP**-complete even for the forests of maximum degree three.*

(ii) *The burning problem is **NP**-complete even for the disconnected graphs of maximum degree three.*

Proof. First, if in the proof of Theorem 46 we keep the graphs Q_i's, Q_i''s, and T_i's disjoint, and take $T(X)$ as a forest with those graphs as its components, then still we can have the same argument and reduction from the Distinct 3-Partition problem. Thus, we conclude the result in part (i). Now, part (ii) is a corollary of part (i), since every forest (with at least two components) of maximum degree three is a disconnected graph of maximum degree three. □

A *binary tree* is a rooted tree in which every node has at most two children. In a *full binary tree* or a *proper binary tree*, every node that is not a leaf has two children. A *perfect binary* tree is a full binary tree in which all the leaves have the same depth (or they are all at the same level); that is, they are all at the same distance from the root.

Let T be a perfect binary tree of radius r. Assume that s is the root of T, and u and v are the neighbours of s. By definition, we can see that, there is only one node of degree two that is the root s, and every other node except for the leaves is of degree three. Moreover, s is the unique centre node of T, and each of u and v by itself is the root of a perfect binary tree of radius $r - 1$. Also, note that every perfect binary tree of radius r has 2^r leaves. In Figure 3.2 we have examples of binary trees.



(a) A binary tree with root s.

(b) A full binary tree with root s.

(c) A perfect binary tree with root s.

Figure 3.2: Some examples of binary trees.

Note that in the proof of Theorem 46, the gadget graph that we construct is a binary tree. To see this, we can take the end point of path Q_1 that is a leaf as a root for $T(X)$ and then every other node is a descendant of the root and has at most two children. Therefore, as another corollary of Theorem 46, we can say that the burning problem is **NP**-complete even for binary trees. However, we can find the burning number of the perfect binary trees as stated below.

Theorem 50. *If T is a perfect binary tree of radius r, then $b(T) = r + 1$.*

Proof. We prove the theorem statement by induction on r as follows. First, by definition, a perfect binary tree T of radius 1 consists of only a root node and its two neighbours. Thus, T is a path on three nodes, and hence, by Theorem 19 from Chapter 2, $b(T) = 2$. Now, assume that the burning number of a perfect binary tree of radius $r - 1$ equals r, and T is a perfect binary tree of radius r. Since the root of a perfect binary tree is its centre, then by burning the root at the first step, every other node will be burned after r more steps. Let s be the root of T with neighbours u and v. As we mentioned right after the definition of a perfect binary tree, u and v are the roots of two perfect binary trees of radii $r - 1$ that we call them T_1 and T_2, respectively. Since T includes a perfect binary tree of radius $r - 1$ as its isomorphic subtree, and by Corollary 17 and Lemma 23, we conclude that $r \leq b(T) \leq r + 1$.

By contradiction, suppose that $b(T) = r$, and $(x_1, x_2, \ldots, x_r)$ is an optimum burning sequence for T. Note that $V(T) = V(T_1) \uplus V(T_2) \uplus \{s\}$. Therefore, to burn T we have to burn T_1 and T_2 as well. Also, we have to consider two possibilities: $x_1 \notin T_1$, or $x_1 \notin T_2$. First, assume that $x_1 \notin T_1$. We know that T_1 has r levels (including u) and the only connection of T_1 to the rest of the nodes in T is through the node s. Hence, the distance between every leaf in T_1 and any node in $T \setminus T_1$ is at least r. Since T_1 has 2^{r-1} leaves, and every leaf in T_1 must be burned by the end of the r-th step, then at least one source of fire must be chosen from T_1.

Now, let L denote the set of the leaves in T_1. Since the number of the leaves in a perfect binary tree of radius $r - i$ equals 2^{r-i}, then we conclude that for $2 \leq i \leq r$, $|N_{r-i}[x_i] \cap L| \leq 2^{r-i}$ (where the equality holds only if we choose x_i from the $(i-1)$-th level of T_1). Hence, by following this greedy argument, even in the case that we choose all the x_i's, for $2 \leq i \leq r$, from T_1, we will burn at most $2^{r-2} + 2^{r-3} + \cdots + 1 = 2^{r-1} - 1$ leaves in L, which is a contradiction. By symmetry of T, we have a similar argument for not choosing x_1 from T_2. Hence, we conclude that burning T in r steps is impossible. Therefore, $b(T) = r + 1$, and the proof follows by induction. □

3.2 Path-forests and Spider Graphs

In this section, we prove that the Graph Burning problem is **NP**-complete even for spider graphs and path-forests. We first provide some background on the burning number of trees.

A *terminal path* in a tree T is a path P in T such that one of the end points of P is a leaf of T. The other end point of P, that is not necessary a leaf, is called the *non-terminal* end point of P (if P is of order one, then the non-terminal end point of P and the leaf in P coincide). Assume that $\{Q_i\}_{i=1}^t$ is a set of disjoint terminal paths in T, and let v_i denote the non-terminal point of the path Q_i, for $1 \leq i \leq t$. We call $\{Q_i\}_{i=1}^t$ a *decomposed spider* in T if the path between every pair v_i and v_j does not contain any node of Q_i and Q_j except v_i and v_j. In Figure 3.3, we see an example of a decomposed spider $\{Q_i\}_{i=1}^5$ in a tree. The paths shown with colour blue are the terminal paths.

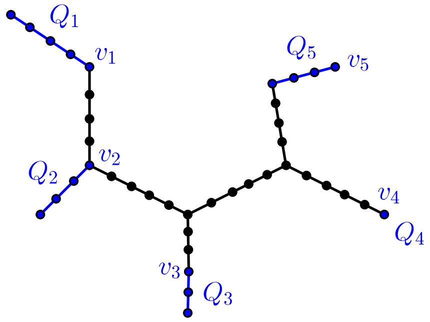


Figure 3.3: An example of a decomposed spider in a tree.

Theorem 51. *Suppose that $\{Q_i\}_{i=1}^t$, where $t \geq 3$, forms a decomposed spider in a tree T, and let v_i be the non-terminal end point of Q_i, for $1 \leq i \leq t$. If $d(v_i, v_j) \geq 2k$ for all $1 \leq i, j \leq t$, and $t \geq k$, then $b(T) \geq k + 1$.*

Proof. Let T' be the smallest connected subgraph of T that contains $\bigcup_{i=1}^t Q_i$. Since T' is an isometric subtree of T, to prove that $b(T) \geq k + 1$ it suffices to show that $b(T') \geq k + 1$, as follows.

First, we show that the burning number of T' is at least k. Let w_i denote the leaf of T' in Q_i. Note that we may have $w_i = v_i$ (in the case that Q_i is of order one). We claim that there is no fire source x_j that spreads the fire to two distinct leaves. By contradiction, suppose that there are two distinct leaves w_i and w_r, and a fire source x_j for which we have that $d(x_j, w_i) \leq k - j$ and $d(x_j, w_r) \leq k - j$ (that is, w_i and w_r both receive the fire started at x_j). By triangle inequality, we conclude that

$$2k \leq d(v_i, v_r) \leq d(w_i, w_r) \leq d(w_i, x_j) + d(x_j, w_r) \leq 2k - 2j < 2k,$$

which is a contradiction. Therefore, it implies that corresponding to every leaf w_i there is a unique fire source x_j such that the fire spread from x_j only burns one leaf of T, that is w_i. Thus, the number of fire sources must be at least as large as the number of the leaves in T' that is $t \geq k$. Hence, we must have $b(T') \geq k$.

Now, we claim that $b(T') \neq k$. By contradiction, suppose that $b(T') = k$, and $(x_1, x_2, \ldots, x_k)$ is an optimum burning sequence for T'. If $t > k$, then the above argument leads to the same contradiction, as the number of the fire sources has to be as large as the number of the leaves. If $t = k$, then let w_i be the leaf that receives the fire from x_k. Since $b(T') = k$, then it implies that $x_k = w_i$. We claim that there is no fire source $x_j \neq x_k$ with $d(v_i, x_j) \leq k$. By contradiction, suppose that there is a fire source $x_j \neq x_k$ with $d(v_i, x_j) \leq k$, and let w_r be the leaf of T' that receives the fire spread from x_j. Thus, we have that

$$2k \leq d(v_r, v_i) \leq d(w_r, v_i) \leq d(w_r, x_j) + d(x_j, v_i) \leq k - j + k < 2k,$$

which is a contradiction.

Let s be a neighbour of v_i that is not in the path between v_i and $w_i = x_k$. Since $t \geq 3$, we are sure that such a node s does exist. By assumption, we know that $x_k = w_i$, and therefore s can not receive the fire spread from x_k. On the other hand, the distance between s and any other fire source must be at least k. Thus, s can not be burned by the end of the k-th step, which is a contradiction. Hence, we have that $b(T') \geq k + 1$. □

By Theorem 19 from Chapter 2, we know that for a path P with $b(P) = k$, burning P in k steps is equivalent to decomposing P into k subpaths such that each of them is a closed neighbourhood of a fire source. Let $(x_1, x_2, \ldots, x_k)$ be an optimum burning sequence for such a path P. As we mentioned $N_{k-1}[x_1]$ is a subpath of P. If x_1 is close enough to the end points of P, then $P \setminus N_{k-1}[x_1]$ is path that can be burned in $k-1$ steps. If x_1 is faraway enough from the end points of P, then $N_{k-1}[x_1]$ is a path of order $2k-1$, and $P \setminus N_{k-1}[x_1]$ is a path-forest with two components that can be burned in $k-1$ steps. We would like to generalize this idea for an arbitrary tree, as follows.

Assume that we want to find the burning number and an optimum burning sequence for a given tree T. If there is an optimum burning sequence $(x_1, x_2, \ldots, x_k)$ for T such that $V(T) \subseteq N_{k-1}[x_1]$, then we claim that $b(T) = \text{rad}(T) + 1$. Namely, it implies that there is at least one node such as v for which $d(v, x_1) = k-1$; otherwise, we conclude that $V(T) \subseteq N_{k-2}[x_1]$, and therefore, $b(T) \leq k-1$, which is a contradiction. Thus, the eccentricity of the node v is $k-1$. Since $\text{rad}(T)$ is the smallest eccentricity of a node in T, then we conclude that $k-1 \geq \text{rad}(T)$. Therefore, $b(T) = k \geq \text{rad}(T)+1$. On the other hand, by Theorem 23 from Chapter 2, we know that $b(T) \leq \text{rad}(T) + 1$. Hence, we conclude that $b(T) = k = \text{rad}(T) + 1$.

If $T \setminus N_{k-1}[x_1]$ is non-empty, then this implies that

$$T \setminus N_{k-1}[x_1] \subseteq N_{k-2}[x_2] \cup N_{k-3}[x_3] \cup \ldots \cup N_0[x_k].$$

Note that it does not imply that $b(T \setminus N_{k-1}[x_1]) \leq k-1$. For example, let T be the tree shown in Figure 3.4. We can check that the sequence (x_1, x_2, x_3, x_4) is an optimum burning sequence for T. However, $T \setminus N_3[x_1]$ consists of 5 disjoint isolated nodes. Thus, $b(T \setminus N_3[x_1]) = 5 \nleq 2$.

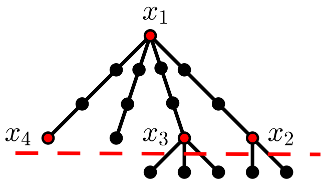


Figure 3.4: A tree T with an optimum burning sequence.

Hence, we are motivated to state the following conjecture.

Conjecture 52. *Suppose that $\{Q_i\}_{i=1}^t$, where $t \geq 3$, forms a decomposed spider in a tree T, and let v_i be the non-terminal end point of Q_i, for $1 \leq i \leq t$. If $b(\uplus_{i=1}^t Q_i) \geq k$, and $d(v_i, v_j) \geq 2k$ for all $1 \leq i, j \leq t$, then $b(T) \geq k+1$.*

Conjecture 52 may be helpful in finding a lower bound on the burning number of a tree T (as we can see the conjecture is true for paths by Theorem 19, and also we will see later on the conjecture is true for some specific spider graphs). In particular, if the burning number of a tree T is strictly less than $\mathrm{rad}(T)+1$, and the conjecture was true, then we guess that by starting from the leaves of T we could probably find a good lower bound on $b(T)$. Note that by Theorem 51, when $t \geq k$, the above conjecture is true. Also, we can prove the following lemma, since the leaves in any spider graph $SP(s,r)$, with $s \geq r$, form a decomposed spider that satisfies the conditions in Theorem 51. We use this lemma in the proof of the **NP**-completeness of the Graph Burning problem for spider graphs.

Lemma 53. *For a spider graph $SP(s,r)$, with $s \geq r$, we have that $b(SP(s,r)) = r+1$. Moreover, for $s \geq r+2$, every optimal burning sequence of $SP(s,r)$ must start by burning the central node.*

Proof. By Theorem 23, we know that $b(SP(s,r)) \leq r+1$, as $SP(s,r)$ has radius r. Since $SP(r,r)$ is an isometric subgraph of $SP(s,r)$ where $s \geq r$, then it suffices to show that $b(SP(r,r)) = r+1$.

First, we prove that $b(SP(r,r)) \geq r+1$. We index the leaves of $SP(r,r)$ with $w_1, w_2, \ldots, w_r$. For $1 \leq i \leq r$, let Q_i be the graph induced by w_i; that is, Q_i is a path of order one. Hence, every Q_i is a terminal path in $SP(r,r)$ with the non-terminal end w_i, and for every distinct pair $1 \leq i, j \leq r$, we have that $d(w_i, w_j) = 2r$. Therefore, by Theorem 51, we conclude that $b(SP(r,r)) \geq r+1$.

Now, suppose that $s \geq r+2$ and there exists an optimal burning sequence for $SP(s,r)$ such as $(x_1, x_2, \ldots, x_{r+1})$ in which x_1 is not the central node. Since $s \geq r+2$ and $b(SP(s,r)) = r+1$, then by Pigeonhole Principle, one of the arms does not include any source of fire, unless we choose the central node as a fire source. Note that by assumption, x_1 is not the central node. Since the only connection between the nodes in that arm to the rest of the nodes in $SP(s,r)$ goes through the central node, then in both cases, we need at least $1 + (r+1)$ steps for burning the leaf on that arm, which is a contradiction. Thus, every optimal burning sequence for $SP(s,r)$ starts by burning the central node, where $s \geq r+2$. □

Using the above lemma, we now prove that the burning problem is **NP**-complete for trees with an elementary-seeming structure as spider graphs. We note that Theorem 54 and Corollary 55 were proven independently in [12] and in a preprint of the paper [18] on arXiv.

Theorem 54. *The burning problem is* ***NP****-complete for spider graphs.*

Proof. By Theorem 46, the burning problem is in **NP**. As in the proof of Theorem 46, we give a reduction from the Distinct 3-Partition problem into the burning problem, in which the gadget graph that we construct is a spider graph.

Given an instance of the Distinct 3-Partition problem, that is, a set $X = \{a_1, \ldots, a_{3n}\}$ of positive distinct integers and a positive integer B such that each $B/4 < a_i < B/2$, we construct a graph G as follows. Since the Distinct 3-Partition problem is strongly **NP**-complete (as in the proof of Theorem 46), without loss of generality we assume that B is bounded above by a polynomial in the length of the input.

Suppose that $\max X = m+1$, and let $Y = \{2a_i - 1 : a_i \in X\}$. Therefore, $Y \subseteq O_{m+1}$. We then make a copy of the spider graph $SP(2m+5, m+1)$ with centre s, called G_s. Now, for any positive odd integer $l \in O_{m+1} \setminus Y$, we connect by an edge a leaf of a copy of P_l (a path on l nodes) to a distinct leaf of $SP(2m+5, m+1)$. We connect (by an edge) n copies of P_{2B-3}, called $Q_1, Q_2, \ldots, Q_n$ to distinct leaves of $SP(2m+5, m+1)$ that we have not used for attaching any other P_l, with $l \in O_{m+1} \setminus Y$. We call the resulting graph G. We can see that G is a spider tree with spider head s. Since $V(G)$ is the disjoint union of the spider graph $SP(2m+5, m+1)$ and the paths $Q_1, Q_2, \ldots, Q_n$, and the paths P_l, with $l \in O_{m+1} \setminus Y$, we have that

$$|V(G)| = \sum_{i=1}^{m+1} (2i-1) + (2m+5)(m+1) + 1 = (m+1)^2 + (2m+5)(m+1) + 1,$$

which is of order $O(B^2)$ in terms of B. Since B is bounded above by a polynomial in the length of the input, it implies that we construct graph G in polynomial time in the length of the input. We want to show that, there is a partition of X into n triples such that the numbers in each triple add up to B if and only if $b(G) \leq m+2$.

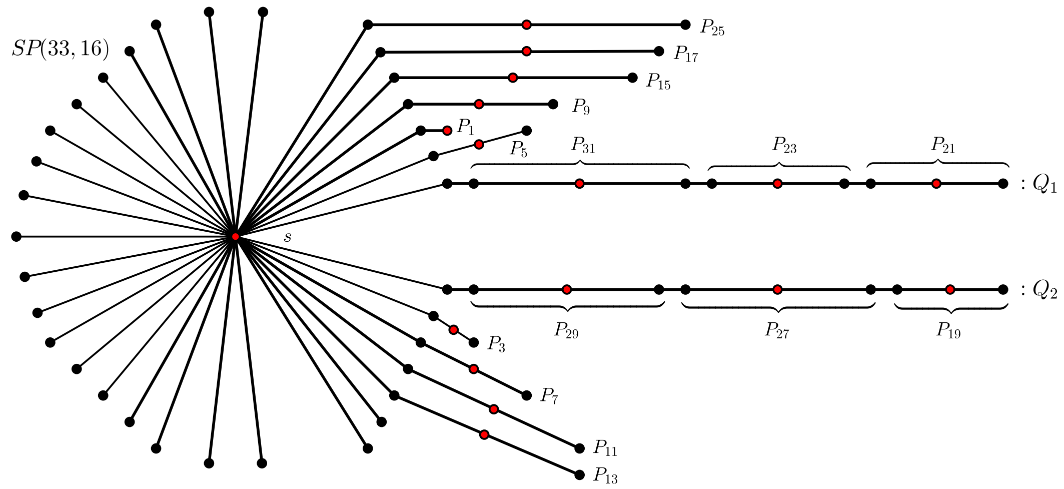


Figure 3.5: A sketch of the tree G.

First, assume that there is a partition of X into n triples such that the numbers in each triple add up to B. Consequently, paths $Q_1, Q_2, \ldots, Q_n$ can be partitioned into smaller paths of orders $\{2a_i - 1 : a_i \in X\}$. For $l \in Y$, we set $x_{m+2-(\frac{l-1}{2})}$ to be the middle node of the paths P_l, applied in such a partition of $Q_1, Q_2, \ldots, Q_n$. Then we take $x_1 = s$, and for any $l \in O_{m+1} \smallsetminus Y$, we set the middle node of P_l as $x_{m+2-(\frac{l-1}{2})}$. The sequence $(x_1, x_2, \ldots, x_{m+2})$ is a burning sequence for G. Thus, $b(G) \leq m+2$.

For example, let $X = \{10, 11, 12, 14, 15, 16\}$, and $B = 39$. Then the graph G is shown in Figure 3.5. Here, we have that $n = 2$, and $m = \max\{a_i : a_i \in X\} - 1 = 15$. Therefore, $Y = \{19, 21, 23, 27, 29, 31\}$, and $O_{16} \smallsetminus Y = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 25\}$. The red nodes in Figure 3.5 denote a burning sequence of length 17 for tree G.

Conversely, suppose that $b(G) \leq m+2$. Since G_s is an isometric subtree of G, then Theorem 18 and Lemma 53 imply that $b(G) = m+2$. Thus, G has a burning sequence $(x_1, x_2, \ldots, x_{m+2})$. We have to show that there is a partition of X into n triples such that the numbers in each triple add up to B. First, note that we use at most $m+1$ leaves of G_s for attaching the paths P_l, with $l \in O_{m+1}$, and the paths $Q_1, Q_2, \ldots, Q_n$. Thus, there is a copy of $SP(m+4, m+1)$ that is an isometric subtree of G and the only connection of its leaves to the rest of G is through node s. Therefore, by Lemma 53, we conclude that $x_1 = s$. On the other hand, by burning node s at the first step, all the nodes in G_s will be burned by the end of the $(m+2)$-th step. Thus, without loss of generality we can assume that for $2 \leq i \leq m+2$, all x_i's are selected from $G \smallsetminus G_s$.

Now, by Equation (2.1), we know that $G \smallsetminus G_s = \bigcup_{i=2}^{m+2} N_{m+2-i}[x_i]$. Since $G \smallsetminus G_s$ is a path-forest, then $N_{m+2-i}[x_i]$ must be a path of order at most $l = 2(m+2-i)+1$, for $2 \leq i \leq m+2$. Besides, we have that

$$\begin{aligned}|V(G \smallsetminus G_s)| &= 2nB - 3n + \sum_{l \in O_{m+1} \smallsetminus Y} l \\ &= \sum_{i=2}^{m+2} \left(2(m+2-i)+1\right).\end{aligned}$$

Therefore, it implies that $N_{m+2-i}[x_i]$ must be a path of order exactly equal to $l = 2(m+2-i)+1$, for $2 \leq i \leq m+2$. Hence, there must be a partition of $G \smallsetminus G_s$ by the set of paths of orders in O_{m+1}, in which the centre of each path in the partition is a fire source.

We claim that there is a burning sequence for G in which the central node of each P_l, $l \in O_{m+1} \smallsetminus Y$ (that we attached to a leaf of G_s), is selected as a fire source. We can prove this claim by switching the paths that are possibly displaced in the current partition for $G \smallsetminus G_s$. Thus, the closed neighbourhoods of the rest of the fire sources form a partition for $Q_1, Q_2, \ldots, Q_n$ in terms of paths of orders $2a_i - 1 \in Y$. Since each Q_i is of order $2B-3$, then it implies that there is partition for X into triples such that the elements in each triple add up to B. □

We conclude the following corollary.

Corollary 55. *The burning problem is* ***NP****-complete even for path-forests.*

Proof. If we delete the spider graph $SP(2m+5, m+1)$ in the proof of Theorem 54, and keep the rest of the parts of the gadget graph G the same, then we will still have the analogous argument for the disjoint union of the paths $Q_1, Q_2, \ldots, Q_n$, and the paths P_l with $l \in O_{m+1} \smallsetminus Y$. Thus, we can have a reduction from the Distinct 3-Partition problem to the Graph Burning problem for path-forests. Therefore, we conclude that the burning problem is strongly **NP**-complete for path-forests. □

Note that in Theorem 53 and the above corollary, we do not have any restriction on the number of the arms in $SP(2m+5, m+1)$ and on the length of the paths in constructing the gadget graphs. In other words, the parameter m is unbounded.

3.3 Algorithms for finding the Burning Number of Path-forests and Spider Graphs

In this section, we present a polynomial time algorithm that finds the burning number of path-forests when the number of components and their orders are restricted, and then we find another polynomial time algorithm that finds the burning number of spider trees with restricted number of arms and with restrictions on the length of the arms. We first provide some terminology and background.

From Theorem 19 in Chapter 2, we know that burning a path in k steps is equivalent to decomposing the path into k smaller subpaths of restricted orders. In particular, burning a path P of order k^2 in k steps is equivalent to decomposing P into paths of orders $1, 3, \ldots, 2k - 1$. Similarly, since the closed neighbourhood of a node in a path-forest G is a path, by Theorem 19 from Chapter 2, burning G in k steps is equivalent to decomposing G into k smaller paths of restricted orders. In particular, if G is of order k^2, and $b(G) = k$, then it means that we are lucky enough to have a decomposition of G into k paths of orders $1, 3, \ldots, 2k - 1$. Here in Figure 3.6 we see an example of a path-forest of order 25 that has been decomposed into paths of orders $1, 3, 5, 7$, and 9. The path of order 5 is shown with colour blue.

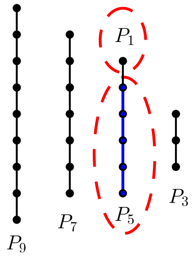


Figure 3.6: A path-forest of order 25.

As we saw in Theorem 19 from Chapter 2, a path such as P_8 that is not of square order has burning number 3, but the paths appearing in its decomposition are not necessary all of odd orders 1, 3, and 5. Similarly, there are path-forests like G with burning number k that can not be decomposed into paths of the odd orders $1, 3, \ldots, 2k - 1$, as we see an example in Figure 3.7. However, such a path-forest is a subgraph of a path-forest that can be partitioned into paths of orders in O_k.

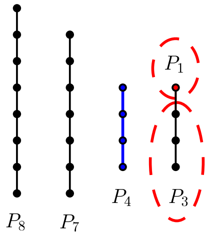


Figure 3.7: A path-forest of order 23.

That is the key idea for our algorithm. We say that G is a *maximal path-forest* if it can be decomposed into paths of orders $1, 3, \ldots, 2t-1$ for some positive integer t. In such a case, it is clear that G is of order t^2. For example, the graph G in Figure 3.6 is a maximal path-forest of order 25. In general, if G is a path-forest with s components and with burning number t, then G corresponds to a sequence of positive integers such as $(l_1, l_2, \ldots, l_s)$, where $l_1 \geq l_2 \geq \cdots \geq l_s$, and l_i denotes the order of the i-th component of G. Note that by Theorem 9 from Chapter 2, we know that $b(G)$ is at least at large as the number of the components in G. Hence, $s \leq t$. Therefore, from now on we represent a path-forest with burning number t by a sequence of integers as defined above.

Here is briefly what we do in the algorithm, and the detailed explanation follows after. For each $k \geq 1$, we first build up all the maximal path-forests with burning number k that have at most $t-1$ components, where t is a fixed parameter. Then recursively, we construct all the maximal path-forests with exactly t components, and at each stage we check if G is a subgraph of them or not.

We denote the set of maximal path-forests with t components and with burning number k by MPF_k^t. For example, for $k = t = 1$, as we mentioned in Chapter 2, the only graph with burning number one is $K_1 = P_1$, which is a maximal path-forest. Thus, $\mathrm{MPF}_1^1 = \{P_1\} = \{(1)\}$. In general, we can see that for any $k \geq 1$, $\mathrm{MPF}_k^k = \{(2k-1, 2k-3, \ldots, 1)\}$. This is true, since first by Theorem 9 from Chapter 2, we know that $b(H)$ is at least at large as the number of the components that is k. On the other hand, we can cover the node set of $(2k-1, 2k-3, \ldots, 1)$ by supaths of orders $1, 3, \ldots, 2k-1$. Hence, $b(H)$ is at most k. Therefore, $b(H) = k$. Also, note that for any $k \geq 1$, $\mathrm{MPF}_k^1 = \{P_{k^2}\} = \{(k^2)\}$.

Algorithm 56. *Suppose that $G = (s_1, s_2, \dots, s_t)$, for a constant $t \geq 1$, represents a path-forest in which s_i denotes the order of the i-th component of G, and $s_1 \geq s_2 \geq \cdots \geq s_t$. Let m be a positive integer such that $s_1 \leq m$; that is, the order of the components of G is bounded above by m. Then we perform the following steps.*

Stage 1. *First, for each $1 \leq r \leq t-1$, we perform Stages 1.1 and 2.1:*

Stage 1.1. *We set $\mathrm{MPF}_r^r = \{(2r-1, 2r-3, \dots, 1)\}$.*

If $(s_1, s_2, \dots, s_r) \notin \mathrm{MPF}_r^r$, then go to the next stage.

Stage 1.2. *For $k \geq r+1$, we perform the following steps:*

Stage 1.2.1. *For each $H = (l_1, l_2, \dots, l_{r-1}) \in \mathrm{MPF}_{k-1}^{r-1}$, we form the sequence $H' = (2k-1, l_1, \dots, l_{r-1})$.*

We rearrange the numbers in the sequence H' if they do not appear in a decreasing order, and we add it to the set MPF_k^r.

If $(s_1, s_2, \dots, s_r) \subseteq H'$, then finish Stage 1.2.1.

Stage 1.2.2. *For each $H = (l_1, l_2, \dots, l_r) \in \mathrm{MPF}_{k-1}^r$, and each $1 \leq i \leq r$, we form the sequence $H_i = (l_1, \dots, l_{i-1}, l_i + 2k - 1, l_{i+1}, \dots, l_r)$.*

We rearrange the numbers in the sequences H_i if they do not appear in a decreasing order, and we add them to the set MPF_k^r.

If $(s_1, s_2, \dots, s_r) \subseteq H_i$, then finish Stage 1.2.2.

Stage 2. *For $r = t$, we perform the following steps:*

Stage 2.1. *We set $\mathrm{MPF}_t^t = \{(2t-1, 2t-3, \dots, 1)\}$.*

If $G \subseteq (2t-1, 2t-3, \dots, 1)$, then stop and return $b(G) = t$.

Stage 2.2. *For $k \geq t+1$, we perform the following steps:*

Stage 2.2.1. *For each $H = (l_1, l_2, \dots, l_{t-1}) \in \mathrm{MPF}_{k-1}^{t-1}$, we form the sequence $H' = (2k-1, l_1, \dots, l_{t-1})$.*

We rearrange the numbers in the sequence H' if they do not appear in a decreasing order, and we add it to the set MPF_k^t.

If $G \subseteq H'$, then stop and return $b(G) = k$.

Stage 2.2.2. *For each $H = (l_1, l_2, \dots, l_t) \in \mathrm{MPF}_{k-1}^t$, and for each $1 \leq i \leq t$, we form the sequence $H_i = (l_1, \dots, l_{i-1}, l_i + 2k - 1, l_{i+1}, \dots, l_t)$.*

We rearrange the numbers in the sequence H_i if they do not appear in a decreasing order, and we add it to the set MPF_k^t.

If $G \subseteq H_i$, then stop and return $b(G) = k$.

The algorithm works since every graph G that is not a subgraph of a graph in MPF_i^t, for all $1 \le i < k$, but G is a subgraph of a graph in MPF_k^t, has burning number k. This is true since first, by Corollary 18 from Chapter 2, $b(G) \le k$. On the other hand, if $b(G) = i \le k$, then it implies that we can cover the nodes set of G by paths of orders $1, 3, \dots, 2i-1$; that is, G is a subgraph of a graph in MPF_i^t, which is a contradiction. Namely, in such a case the algorithm should have stopped at Stage 2 for value i, and would not have proceeded till value k. Therefore, we must have $b(G) = k$. We have the following fact about Algorithm 56.

Theorem 57. *Suppose that $G = (s_1, s_2, \dots, s_t)$, for an integer constant $t \ge 1$, represents a path-forest in which s_i denotes the order of the i-th component of G, and $s_1 \ge s_2 \ge \dots \ge s_t$. Let m be a positive integer such that $s_1 \le m$; that is, the order of the components of G is bounded above by m. If t is a fixed constant in terms of m, then Algorithm 56 finds the burning number of G in polynomial time in terms of the input.*

Proof. Given the graph G, suppose that for some $k \ge t$, Algorithm 56 stops by recognizing G as a subgraph of a graph in MPF_k^t. Note that t is a fixed constant in terms of m. Thus, by Theorem 21 from Chapter 2, we derive that if $H = (l_1, l_2, \dots, l_r)$ is a graph in MPF_i^r (generated by Algorithm 56), with $1 \le r \le t$ and $i \ge r$, then

$$b(H) \le \left\lceil \sqrt{\sum_{j=1}^{r} l_j} \right\rceil + r - 1 \le \sqrt{mt} + t - 1 = O(\sqrt{m}).$$

On the other hand, since $b(H) = i$, then there is a partition of the set O_i into subsets $\{A_j\}_{j=1}^r$ such that $l_j = \sum_{a \in A_j} a$, for $1 \le j \le r$. It implies that $l_j \le \sum_{a \in O_i} a = i^2 = O(m)$, for $1 \le j \le r$. Hence, the length of the longest l_j that appears in the representation of such a graph H is of order m. Let $l = O(m)$ be the length of the longest component in a graph H generated by Algorithm 56. Thus, any graph H generated by Algorithm 56 is a subgraph of the graph $G_0 = (l, l, \dots, l)$ with t components. Since these graphs are distinct, then the total number of the graphs that is generated by Algorithm 56 is of order $O(m^t)$.

Moreover, note that for $r = t$ and $k \geq t$, each time that we add a new graph $H = (l_1, l_2, \ldots, l_t)$ to MPF_k^t, we check to see if G is a subgraph of H or not. We simply can do this comparison by checking if $s_i \leq l_i$, for $1 \leq i \leq t$. Thus, the total number of the steps that we perform in Algorithm 56 is bounded above by $O(tm^t)$. Since t is a fixed constant in terms of m, then Algorithm 56 is a polynomial time algorithm in terms of the input. □

In the following, we try to find the burning number of spider graphs, again using a bottom-up dynamical programming approach. First we need some facts to use for this algorithm. The following theorem is a key in the algorithm that we will present, and shows that for a spider tree we always can have an optimum burning sequence in which the first source of fire is close to the spider head.

Theorem 58. *If G is a spider graph with $s \geq 3$ arms and the spider head c, then there is an optimum burning sequence $(x_1, x_2, \ldots, x_k)$ for G such that $d(x_1, s) \leq k-1$.*

Proof. We prove this by strong induction on the number of the nodes in G. The smallest order spider graph is a star with three leaves. By Theorem 8 from Chapter 2 we know that the burning number of such a star equals 2 and in every optimum burning sequence for this graph the first fire must be the centre that is the spider head. Hence, the theorem statement is true for this spider.

Now, suppose that the theorem statement is true for every spider graph of order at most $n-1$, and G is a spider graph of order n with $s \geq 3$ arms and spider head c. Also, assume that $L_1, L_2, \ldots, L_s$ are the arms of G, and $v_1, v_2, \ldots, v_s$ are their corresponding leaves. Finally, suppose that the order of each arm L_i is denoted by l_i. Let $(x_1, x_2, \ldots, x_k)$ be an optimum burning sequence for G. By Equation (2.1), we know that

$$V(G) = N_{k-1}[x_1] \cup N_{k-2}[x_2] \cup \ldots \cup N_0[x_k].$$

If $d(x_1, c) \leq k-1$, then we are done. Hence, let $d(x_1, c) \geq k$, and $x_1 \in L_i$ where $1 \leq i \leq s$. We have two possibilities for l_i: either $l_i \leq 2k-2$ or $l_i \geq 2k-1$.

Case 1. If $l_i \leq 2k-2$, then it implies that $d(c, v_i) \leq 2k-2$. Let x be the node in L_i for which $d(x, v_i) = k-1$. Therefore, we have that $d(c, x) \leq k-1$. Note that we can cover all the nodes in $L_i \cup \{c\}$ with $N_{k-1}[x]$. Hence, $V(G) \setminus N_{k-1}[x] \subseteq V(G) \setminus N_{k-1}[x_1]$. Thus, we still have that

$$V(G) = N_{k-1}[x] \cup N_{k-2}[x_2] \cup \ldots \cup N_0[x_k].$$

Note that some of the fire sources x_j's, with $j \geq 2$, might be in $N_{k-1}[x] \cap L_i$. Therefore, we have that $b(G \setminus N_{k-1}[x]) = t \leq k-1$. Hence, we can find a burning sequence of length t such as $(x'_2, x'_3, \ldots, x'_t)$ for $G \setminus N_{k-1}[x]$. Also, for $t+1 \leq j \leq k$, we define x'_j to be a node of distance $j-1$ from x. Thus, for $t+1 \leq j \leq k$, $d(x'_j, x) \geq j-1$, and $d(x'_j, x_r) \geq r-1+j-1 \geq j-r$, for any $2 \leq r \leq t$. Therefore, the sequence $(x'_1 = x, x'_2, \ldots, x'_k)$ forms a desired optimum burning sequence for G.

Case 2. If $l_i \geq 2k-1$, then either $d(v_i, x_1) \leq k-1$ or $d(v_i, x_1) \geq k$. We claim that there is a burning sequence for G such as $(x'_1, x'_2, \ldots, x'_k)$ such that $x'_1 \in L_i$ and $d(x'_1, v_i) \leq k-1$, or equivalently, $G \setminus N_{k-1}[x'_1]$ is connected. If $d(x_1, v_i) \leq k-1$, then we are done. If $d(v_i, x_1) \geq k$, then $G \setminus N_{k-1}[x_1]$ is the disjoint union of a spider graph G' and a path P, such that P is a subpath of L_i containing v_i. Let u be the leaf of G' that is in L_i, and v be the other end point of P that probably is different from v_i. We know that $b(G \setminus N_{k-1}[x_1]) \leq k-1$. Hence, by Lemma 20, we have that

$$t = b(G' + uv + P) \leq b(G' \cup P) = b(G \setminus N_{k-1}[x_1]) \leq k-1.$$

Note that $G' + uv + P$ is a subtree of G that is (isomorphic to) a spider of the same number of arms as G. In fact, the i-th arm of $G' + uv + P$ is (isomorphic to) a subpath of L_i with exactly $2k-1$ less nodes than L_i. Also, note that some of the fire sources x_j's, with $j \geq 2$, might be in $N_{k-1}[x_1] \cap L_i$. Let $(x'_2, x'_3, \ldots, x'_t)$ be an optimum burning sequence for $G' + uv + P$, and x'_1 be the node in L_i with $d(x'_1, v_i) = k-1$. Also, for $t+1 \leq j \leq k$, we take x'_j to be a node of distance j from x'_1 that is on the path connecting x'_1 and v_i. Thus, the sequence $(x'_1, x'_2, \ldots, x'_k)$ forms a burning sequence for G, such that $x'_1 \in L_i$, and $G \setminus N_{k-1}[x'_1]$ is connected. Hence the claim is true.

Now, by above claim, without loss of generality, we assume that $N_{k-1}[x_1]$ contains v_i. That is, we have a burning sequence $(x_1, x_2, \ldots, x_k)$ for G such that $G' = G \setminus N_{k-1}[x_1]$ is a spider graph with smaller number of nodes than G, and with the same number of arms and the same spider head c. In fact, for $j \neq i$, and $1 \leq j \leq s$, L_j is the

j-th arm of G' too, and the i-th arm of G' is a subset of L_i that contains exactly $2k-1$ nodes less than L_i. Hence, we have that $b(G') = t \leq k-1$, and by induction hypothesis, G' must have a burning sequence $(x'_2, x'_3, \ldots, x'_t)$ such that $d(x'_2, c) \leq t-1 \leq k-2$. We have two possibilities: either $x'_2 \in L_i$, or $x'_2 \in L_j$ for some $j \neq i$.

If $j = i$, then let x be the neighbour of x'_2 that is on the path which connects x'_2 to v_i. Also, let x' be the neighbour of x_1 that is on the path connecting x_1 to v_i. Hence, we have that

$$G \smallsetminus (N_{k-1}[x] \cup N_{k-2}[x']) = G \smallsetminus (N_{k-1}[x_1] \cup N_{k-2}[x'_2]).$$

Now, for $t+1 \leq r \leq k$, we take x'_r to be the node in L_i on the path connecting v_i to x' that is of distance $r-2$ from x'. Finally, we take $x'_1 = x$, and we redefine $x'_2 = x'$. Thus, the sequence $(x'_1, x'_2, \ldots, x'_k)$ forms a burning sequence for G in which $d(x'_1, c) \leq k-1$.

If $j \neq i$, then let x be the neighbour of x'_2 that is on the path connecting x'_2 to c. Also, let x' be the neighbour of x_1 that is closer to v_i. Hence, we have that

$$L_i \smallsetminus (N_{k-1}[x] \cup N_{k-2}[x']) = L_i \smallsetminus (N_{k-1}[x_1] \cup N_{k-2}[x'_2]),$$

(by isomorphism). Also,

$$L_j \smallsetminus (N_{k-1}[x] \cup N_{k-2}[x']) = L_j \smallsetminus (N_{k-1}[x_1] \cup N_{k-2}[x'_2]).$$

But,

$$G \smallsetminus (N_{k-1}[x] \cup N_{k-2}[x']) \subseteq G \smallsetminus (N_{k-1}[x_1] \cup N_{k-2}[x'_2]),$$

and we know that $N_{t-2}[x'_3] \cup N_{t-3}[x'_4] \cup \ldots \cup N_0[x'_t]$ forms a covering for $G \smallsetminus (N_{k-1}[x] \cup N_{k-2}[x'])$. In fact, $G \smallsetminus (N_{k-1}[x] \cup N_{k-2}[x'])$ is an isometric subforest of $G \smallsetminus (N_{k-1}[x_1] \cup N_{k-2}[x'_2])$. Thus, by Corollary 18 from Chapter 2, we have that

$$\begin{aligned} b(G \smallsetminus (N_{k-1}[x] \cup N_{k-2}[x'])) &\leq b(G \smallsetminus (N_{k-1}[x_1] \cup N_{k-2}[x'_2])) \\ &\leq t-1 \leq k-2. \end{aligned}$$

Hence, there must be an optimum burning sequence $(x''_3, x''_4, \ldots, x''_{t'})$, where $t' \leq t$ for $G \smallsetminus (N_{k-1}[x] \cup N_{k-2}[x'])$. Now, for $t'+1 \leq r \leq k$, we take x''_r to be the node in L_i on the path connecting v_i to x' that is of distance $r-2$ from x'. Finally, we take $x''_1 = x$, and we define $x''_2 = x'$. Thus, the sequence $(x''_1, x''_2, \dots, x''_k)$ forms a burning sequence for G in which $d(x''_1, c) \leq k-1$. □

The following lemma provides us with another key tool for finding the burning number of spider graphs.

Lemma 59. *Let G be a spider graph with spider head c. Also, suppose that for a positive integer k and a node $x \neq c$ in G, $G \smallsetminus N_{k-1}[x]$ is a path-forest (that is, $d(x,c) \leq k-1$) with at least two components, and $b(G \smallsetminus N_{k-1}[x]) \leq k-1$. If $x \neq c$, and the neighbour of x on the path connecting x to c is x', then we have that $b(G \smallsetminus N_{k-1}[x']) \leq k-1$.*

Proof. Assume that a spider graph G with the above conditions is given, and we have the nodes x and x' as mentioned in the lemma's statement. Let x be in an arm of G called L_s. We have two possibilities for L_s: either $L_s \smallsetminus N_{k-1}[x]$ is empty or not.

First, suppose that $L_s \smallsetminus N_{k-1}[x]$ is not empty. Hence, since in this case one of the components of $G \smallsetminus N_{k-1}[x]$ is contained in L_s, then it implies that L_s is of order at least $k + d(x,c)$. By assumption, we know that each component of $G_1 = G \smallsetminus N_{k-1}[x]$ is a subset of one of the arms in G. Let G_2 be the path-forest $G \smallsetminus N_{k-1}[x']$. We know that G_2 is a path-forest since by assumption, $d(x',c) \leq d(x,c) \leq k-1$. Hence, each component of G_2 is also a subpath of an arm in G. We call the components of G_1 and G_2 that are subpaths of L_s by P and P', respectively. In fact, P' is a superset of P with exactly one more node. Also, each non-empty component of G_2 such as $Q' \neq P'$ is a subset of the corresponding component $Q \neq P$ of G_1, and has exactly one node less than Q.

Since by assumption, $b(G_1) = t \leq k-1$, then there must be a burning sequence $(x_1, x_2, \dots, x_t)$ for G_1. Note that each $N_{t-j}[x_j]$ is a path of order at most $2(t-j)+1$. Therefore, the closed neighbourhoods of the x_i's cover all the nodes in G_2, except for probably the extra node in P' that is a superset of P. We have two possibilities: either there is a component $Q \neq P$ in G_1 that is of order one, or the order of each component of G_1 is of order at least two.

If there is a component $Q \neq P$ of G_1 that is of order one and is burned by x_i, then let x'_i be the extra node in $P' \smallsetminus P$. If $(x_1, x_2, \dots, x_t)$ does not burn x'_i, then the sequence

$(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_t)$ is a burning sequence for G_2. Thus, $b(G_2) \leq k - 1$.

If every component of G_1 like $Q \neq P$ is of order at least two, then we have again two possibilities: either x_t is in P, or $x_t \notin P$.

If x_t is in P, then let i be the smallest index for which $x_i \in P$, but x_{i-1} is not in P. We know that such an index i does exist, since otherwise, it means that all the x_i's must be in P, and consequently, it implies that P is the only component of G_1, which is a contradiction. Thus, there must an index i such that $x_i \in P$, but x_{i-1} is in a component of G_1 that we call it Q, with $Q \neq P$.

Since each $N_{t-j}[x_j]$ is a path of order at most $2(t-j)+1$, then without loss of generality we can assume that $N_{t-i}[x_i]$ covers at least two nodes less than $N_{t-(i-1)}[x_{i-1}]$. Now, let $x'_i = x_{i-1}$ and $x'_{i-1} = x_i$. Therefore, we have a new covering for G_2 induced by $(x_1, \ldots, x_{i-2}, x'_{i-1}, x'_i, \ldots, x_t)$ in which all the nodes of P' plus one extra node of $L_s \smallsetminus P'$ is covered, while we may have lost covering one node in Q. Now, by moving x_t to cover such a uncovered node in Q, and shifting the place of the fire sources used for covering P without changing their order (if it is necessary), we find a covering for G_2 with t closed neighbourhoods of restricted radii. Hence, by Corollary 6 from Chapter 2, $b(G_2) \leq k - 1$.

If $x_t \notin P$, then there must be a component Q of G_1 for which $x_t \in Q$. Let Q' be the corresponding component of G_2 that has exactly one node less than Q. By moving x_t to cover the extra node in P' (and shifting the place of the fire sources used for covering Q without changing their order, if it is necessary), we find a covering for G_2 by t closed neighbourhoods with restricted radii. Hence, again in this case, $b(G_2) \leq k - 1$.

Now, assume that $L_s \smallsetminus N_{k-1}[x]$ is empty, and $G_1 = G \smallsetminus N_{k-1}[x]$, and $G_2 = G \smallsetminus N_{k-1}[x']$. If $L_s \smallsetminus N_{k-1}[x']$ is empty, then G_2 is an isometric subforest of G_1, and therefore $b(G_2) \leq b(G_1) \leq k - 1$.

If $L_s \smallsetminus N_{k-1}[x']$ is non-empty, then it means that $P' = L_s \smallsetminus N_{k-1}[x']$ contains exactly one node. Also, we know that all the non-empty components of G_2 are subsets of the corresponding components of G_1, with exactly one less node. Assume that $(x_1, x_2, \ldots, x_t)$ is an optimum burning sequence for G_1. Since $L_s \smallsetminus N_{k-1}[x]$ is empty, then there must be non-empty component of G_1 like Q for which $x_t \in Q$. By moving x_t to cover the extra node in P' (and shifting the place of the fire sources used for covering Q without changing their order, if it is necessary), we find a covering for G_2 by t closed neighbourhoods with restricted radii. Hence, again in this case we conclude that $b(G_2) \leq k-1$. □

As a consequence of the above lemma we have the following result.

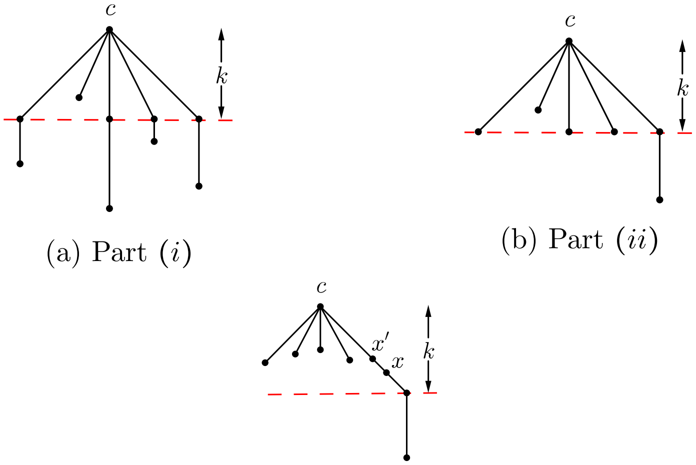
Lemma 60. *Let G be a spider graph with spider head c. Also, suppose that for a positive integer k and a node x in G, $G \smallsetminus N_{k-1}[x]$ is a non-empty path-forest (that is, $d(x,c) \leq k-1$) with at least one component, and $b(G \smallsetminus N_{k-1}[x]) \leq k-1$. If $x \neq c$, then we have one of the following possibilities:*

(i) The graph $G \smallsetminus N_{k-1}[c]$ has at least two components, and $b(G \smallsetminus N_{k-1}[c]) \leq k-1$.

(ii) There is a leaf in G that is of distance $k-1$ from c, and $G \smallsetminus N_{k-1}[c]$ has only one component, and $b(G \smallsetminus N_{k-1}[c]) \leq k-1$.

(iii) There is a node $x' \notin \{x, c\}$ on the path connecting x to c for which $G \smallsetminus N_{k-1}[x']$ has only one component (that is a subset of L_s), and $b(G \smallsetminus N_{k-1}[x']) \leq k-1$, and there is a leaf of G that is of distance $k-1$ from x'.

Proof. Since the spider graph G satisfies the conditions in Lemma 59, after finite number of times of applying Lemma 59, we derive the desired result. Here, in Figure 3.8 we see a layout of the three different cases stated in the lemma.



(a) Part (i)

(b) Part (ii)

(c) Part (iii)

Figure 3.8: Three possible cases for Lemma 60.

□

A *perfect spider of radius* r is a spider graph G with a unique centre node c such that $d(v,c) = r$ for every leaf $v \in G$. We denote the set of all perfect spider trees of radius k with PS_k.

A *k-burning maximal spider graph*, is a spider graph with spider head c that its node set can be decomposed into a perfect spider graph $F = N_{k-1}[s] \in \mathrm{PS}_{k-1}$, where s is a node with $d(s,c) \leq k-1$, and a graph $H \in \mathrm{MPF}_{k-1}$. We denote the set of all k-burning maximal spider graphs by k-BMS. By above Lemma, we can see that there are two different types of the graphs in k-BMS like G: either G is a graph for which the centre node s of the perfect spider graph in the decomposition of G is the spider head, or G is a graph such that the centre node s is not the spider head. If the latter holds, then by Lemma 60 part *(iii)*, we conclude that the path-forest $G \smallsetminus N_{k-1}[s]$ must be a single path of order $(k-1)^2$.

Note that the path-forest that appears in decomposing a k-BMS forms a decomposed spider as described in Conjecture 52. Now, we have the following useful theorem that proves the truth of Conjecture 52 for k-BMS trees.

Theorem 61. *If G is a k-*BMS *with spider head c, then $b(G) = k$.*

Proof. Let G be a k-BMS with spider head c. If the centre of the perfect spider in decomposing G equals c, then it implies that $b(G \smallsetminus N_{k-1}[c]) = k-1$. In this case, $G_1 = G \smallsetminus N_{k-1}[c]$ is in MPF_{k-1}. By contradiction suppose that $b(G) = t \leq k-1$. Thus, by Theorem 58, there must be an optimum burning sequence for G like $(x_1, x_2, \ldots, x_t)$ such that $d(x_1, c) \leq t-1 \leq k-2$.

If $x_1 = c$, then $G \smallsetminus N_{k-1}[c]$ is an isometric subforest of $G \smallsetminus N_{t-1}[c]$, and therefore, $b(G \smallsetminus N_{t-1}[c]) \leq t-1 \leq k-2$, which is a contradiction.

If $x_1 \neq c$, then note that all the arms of G are of length at least $k-1$, and since $b(G \smallsetminus N_{k-1}[c]) = k-1$, there must be at least one arm of G that is of length at least $k-1+2k-3 = 3k-4$. Thus, $G \smallsetminus N_{t-1}[c]$ must have at least two non-empty components, and therefore, by Lemma 59, we have that $b(G \smallsetminus N_{t-1}[c]) \leq t-1 \leq k-2$, which is a contradiction, as $G \smallsetminus N_{k-1}[c]$ is an isometric subforest of $G \smallsetminus N_{t-1}[c]$. Hence, in both cases we find a contradiction, and therefore, $b(G) \geq k$.

If the centre of the perfect spider in decomposing G is a node $s \neq c$, then as we discussed before the theorem statement, the graph $G_1 = G \smallsetminus N_{k-1}[s]$ is a single path of order $(k-1)^2$. Let L_s be the arm of G with $s \in L_s$, and assume that v_s is the node in L_s such that $d(s, v_s) = k - 1$. Also, assume that P is the path between v_s and s, and P' is the path connecting s to c excluding s.

By contradiction suppose that $b(G) = t \leq k - 1$. Thus, by Theorem 58, there must be an optimum burning sequence for G like $(x_1, x_2, \dots, x_t)$ such that $d(x_1, c) \leq t - 1 \leq k - 2$. We consider different possibilities for x_1 as follows:

If x_1 is in $G \setminus (L_s \setminus P')$, then $G \setminus N_{k-1}[s]$ is an isometric subforest of $G \setminus N_{t-1}[c]$, and therefore, we must have $b(G_1) \leq t - 1 \leq k - 2$, which is a contradiction.

If x_1 is in P, then let x' be the neighbour of s on the path connecting s to c. Note that all the leaves of G, except for the leaf in L_s, are of distance $k - 1$ from s. Thus, $G \setminus N_{t-1}[x_1]$ must have at least two non-empty components, and therefore, by applying Lemma 59 for a finite number of times, we have that $b(G \setminus N_{t-1}[x']) \leq t - 1 \leq k - 2$, which is a contradiction, as $G \setminus N_{k-1}[s]$ is an isometric subforest of $G \setminus N_{t-1}[x']$. Hence, in both cases we find a contradiction, and therefore, $b(G) \geq k$. □

By the argument in proof of Theorem 61, we can conclude that the spider graphs in 2-MBS can be decomposed into a perfect spider $SP(s, 1)$ and a single node P_1, with $s \geq 3$; that is, a spider with $s - 1$ arms of length one and an arm of length two. Now, we can present an algorithm for finding the burning number of a spider tree with a constant number of arms and a restriction on the length of the arms, as follows. Note that the burning number of every spider graph is at least two. We show the set of all perfect spider trees with t arms and of radius k by PS_k^t. We denote the set of all k-burning maximal spider graphs with t arms by k-BMS^t.

Algorithm 62. *Suppose that G is a spider tree with arms $L_1, L_2, \dots, L_t$, for a constant $t \geq 1$, such that the length of each arm L_i is denoted by l_i, and $l_1 \geq l_2 \geq \dots \geq l_t$. Let m be a positive integer for which $l_1 \leq m$; that is, the length of each arm in G is bounded above by m. Then we perform the following steps until $G \subseteq H$, for some $H \in k$-BMS^t where $k \geq 2$.*

Stage 1. *For the initial case $k = 2$, we perform the following steps:*

Stage 1.1. *We put the graph $SP(t, 1)$ in PS_1^t.*

Stage 1.2. *We add a single node to one of the arms in $SP(t, 1) \in \mathrm{PS}_1^t$, and we put the resulting graph H in 2-MBS^t.*

If $G \subseteq H$, then return $b(G) = 2$; otherwise, go to Stage 2.

Stage 2. *For $k \geq 3$, we perform the following steps:*

Stage 2.1. *For $0 \leq i \leq k-2$, we make a spider graph with $t-1$ arms of length $k-1-i$, and then we add an additional arm of length $i+k-1$ to it. We call the resulting spider (with t arms) by H_i and we put it in PS^t_{k-1}.*

Stage 2.2. *For $1 \leq s \leq k-1$, and each $F \in \mathrm{MPF}^s_{k-1}$ (generated by Algorithm 56 for the graph $G' = (l_1, l_2, \dots, l_t)$), we join an end point of each component of F to a distinct leaf of $H_0 \in \mathrm{PS}^t_{k-1}$, and we call the resulting graph by F'. Then we add F' to k-MBS^t.*

If $G \subseteq F'$, then stop and return $b(G) = k$.

Stage 2.3. *For $1 \leq i \leq k-2$, we join the end point of longest arm of H_i to a path of order $(k-1)^2$ in MPF^1_{k-1}, and we call the resulting graph by H'_i. Then we add H'_i to k-MBS^t.*

If $G \subseteq H'_i$, then stop and return $b(G) = k$.

If Algorithm 62 stops at $i = k$, then it means that G is a subgraph of a graph in k-MBS^t. By Theorem 61, we know that the burning number of a graph in i-MBS^t equals i. Hence, by Corollary 18 from Chapter 2, we conclude that $b(G) = k$.

On the other hand, if $b(G) = k$, then by Lemma 58, we know that there is an optimum burning sequence $(x_1, x_2, \dots, x_k)$ for G such that $d(x_1, c) \leq k-1$. Hence, $b(G \smallsetminus N_{k-1}[x_1]) \leq k-1$. It implies that $G \smallsetminus N_{k-1}[x_1]$ is a subgraph of a path-forest F in MPF^s_{k-1}, for some $1 \leq s \leq t-1$. Let H be a spider tree with t components that is a supergraph of G, such that $N^H_{k-1}[x_1]$ forms a perfect spider in PS^t_{k-1}. Moreover, $H \smallsetminus N^H_{k-1}[x_1] = F$. Hence, $H \in k$-MBS^t_k. Therefore, the algorithm presented above works, and we have the following theorem about its complexity.

Theorem 63. *Suppose that G is a spider tree with arms $L_1, L_2, \dots, L_t$, for a constant $t \geq 1$, in which the length of each arm L_i is denoted by l_i, and $l_1 \geq l_2 \geq \cdots \geq l_t$. Let m be a positive integer for which $l_1 \leq m$; that is, the length of each arm in G is bounded above by m. If t is a constant in terms of m, then Algorithm 62 finds the burning number of G in polynomial time in terms of the input.*

Proof. Given the graph G, suppose that for some $k \geq t$, Algorithm 62 stops by recognizing G as a subgraph of a graph in MBS^t_k; that is, $b(G) = k$. In Algorithm 62, we first generate all the perfect spider graphs of radius i with t arms, for $1 \leq i \leq k$. Then at Stage 2.2, we need to perform Algorithm 56 for the graph $(l_1, l_2, \ldots, l_t)$ which satisfies all the conditions in Theorem 57. Hence, we perform at most $O(tm^t)$ steps to find all the maximal path-forests generated by Algorithm 56 at Stage 2.2.

On the other hand, by Theorem 23 from Chapter 2, we know that $k = b(G) \leq \mathrm{rad}(G) + 1 \leq m + 1 = O(m)$. Thus, $\frac{k(k+1)}{2} = O(m^2)$. Note that the number of the perfect spider graphs that we generate in Algorithm 62 for each $1 \leq i \leq k$ equals i. Therefore, the total number of the graphs that we create and consider by Algorithm 62 is of order

$$\sum_{i=1}^{k} kO(tm^t) = \frac{k(k+1)}{2} O(tm^t) = O(tm^{t+2}).$$

Finally, note that each time that we add a new spider graph F to MBS_k^t, for $k \geq 2$, we compare G with F. We can simply do this comparison by comparing the lengths of the arms between G and F. Since G and F both have t arms, then the total number of the steps that we perform in Algorithm 62 is bounded above by $O(t^2 m^{t+2})$. Since t is a fixed constant in terms of m, then Algorithm 62 is a polynomial time algorithm in terms of the input. □

Chapter 4

Graph Burning and Random Models

In this chapter, we consider the burning number of the *random geometric graphs.* Also, we define a random version of the burning process for the graphs which we call the *Poisson burning process.* We present some results on the Poisson burning process of graphs in general, and for the paths, in particular. For more probabilistic results on graph burning, see [46].

4.1 Random Geometric Graphs

In this section we present our result on the burning number of random geometric graphs. We need first to provide some background and preliminaries.

Suppose that n is a positive integer and r is a nonnegative real number. Also, let S be the unit square $[0,1]^2$ in the 2-dimensional Cartesian plane. A *random geometric graph* G is defined as follows. First, we choose n points independently and uniformly at random from S as the node set of G, denoted by $V(G)$. That is, the probability of choosing each point from a subset A of S equals to the area of the set A. We connect two nodes u and v in $V(G)$ if and only if the Euclidean distance between u and v in the plane is less than or equal to r.

Note that since by definition, the probability of choosing a point in S more than once is zero, then we can assume here that $|V(G)| = n$. Also, each node u in $V(G)$ can be represented by (u_x, u_y), where u_x is the first coordinate of u in the Cartesian plane, and u_y is the second coordinate of u in the plane. Note that here we have two metrics on the node set of G: The *Euclidean metric* and the *geodesic metric.* Based on the definition of the Euclidean metric on $V(G)$, the *Euclidean distance* between two nodes such as $u = (u_x, u_y)$ and $v = (v_x, v_y)$, denoted by $d_E(u,v)$, equals

$$\left(|u_x - v_x|^2 + |u_y - v_y|^2\right)^{1/2}.$$

According to the geodesic metric on $V(G)$, the *geodesic distance* or *graph distance* between u and v is the length of a shortest path between u and v in G (as we defined in Chapter 1). Hence, we can say that two nodes u and v are adjacent in the geometric graph G if and only if $d_E(u,v) \leq r$.

The set of all random geometric graphs as defined above on S is denoted by $\mathscr{G}(n,r)$. As usual for the random graph models, here we consider asymptotic properties of $\mathscr{G}(n,r)$ as $n \to \infty$, where $r = r(n)$ may and usually does depend on n. Assume that $\{G_n\}_{n=1}^{\infty}$ is a sequence of graphs in which the n-th graph is of order n, for $n \geq 1$. Also, let P be a property of graphs (for instance, connectivity). In graph theory, we say that *asymptotically almost surely* a graph in the sequence $\{G_n\}_{n=1}^{\infty}$ has property P if the probability of the event "G_n has property P" converges to 1 as n goes to infinity. We sometimes, shortly use the notation *a.a.s.* instead of asymptotically almost surely.

It is known that $r_c = \sqrt{\log n/n\pi}$ is a sharp threshold for the connectivity of $G \in \mathscr{G}(n,r)$. That is, for every $\varepsilon > 0$, if $r \leq (1-\varepsilon)r_c$, then the graph G is a.a.s. disconnected; while if $r \geq (1+\varepsilon)r_c$, then G is a.a.s. connected (see [50, 31]). Here, we consider the burning number of the random geometric graph where it is connected.

Assume that u is a node in a random geometric graph $G \in \mathscr{G}(n,r)$. Let B be a ball with centre u and radius kr in the plane, for some positive integer k. Then by definition of $\mathscr{G}(n,r)$, we can see that $N_k[u]$ is a subset of B. In other words, any node of G that is of geodesic distance at most k from u can not be at Euclidean distance more than kr from u in the plane. However, there may be some other nodes in B that are for instance of graph distance $s \geq k+1$ from u. Figure 4.1 shows an example of such a case.

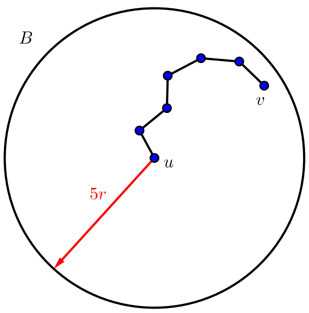


Figure 4.1: In this figure, $d_E(u,v) \leq 5r$, while $d_G(u,v) = 6$.

The following result from [23] shows the relation between the geodesic distance and the Euclidean distance between the nodes of a random geometric graph under some constraints. The results in [23] are stated in the model of a square of side length $\sqrt{n}$, but they can be easily translated to our setting. In fact, in [23], a more precise result was shown, but for our purpose the following version is enough.

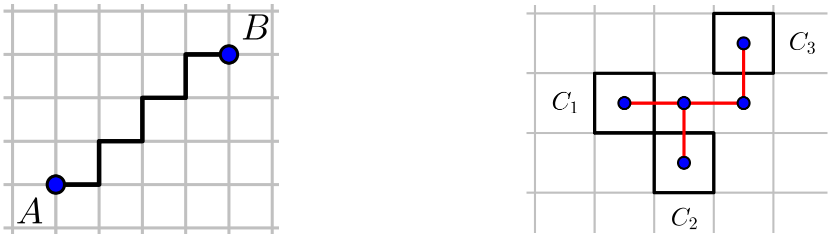
Theorem 64 ([23])**.** *There exists a constant c_0 such that for any $c \geq c_0$, there is a constant $c' = c'(c)$ for which if $r \geq cr_c$, then a.a.s. we have that*

$$d_G(u,v) \leq c' \frac{d_E(u,v)}{r},$$

for every pair of nodes $u, v \in V(G)$, where $G \in \mathscr{G}(n,r)$.

For more details on the random geometric graphs, see [49].

In the proof of the main result of this section, we use the notion of *d-cell-distance* that we define here. Assume that we partition the 2-dimensional Euclidean plane into squares of equal sides, each called a *cell*, that appear in rows and columns. Then we say that two points such as A and B are of *d-cell-distance* if the sum of the absolute values of the differences between the first coordinates and second coordinates of A and B equals d times the width of a cell. Figure 4.2a shows an example of such two points and the distance between them. Similarly, we say that two cells such as C_1 and C_2 are of d-cell-distance if the centres of C_1 and C_2 are of d-cell-distance. Figure 4.2b demonstrates an example of such two cells.



(a) Point A is of 7-cell-distance from point B.

(b) C_1 is of 2-cell-distance from C_2 and of 3-cell-distance from C_3.

Figure 4.2: d-distance-cell.

We need the following consequence of the Chernoff's bound for the proof of the main result in this section; see [38, Corollary 2.3].

Lemma 65 ([38]). *If X is a binomial random variable, then for any $0 < \varepsilon < 3/2$, we have that*

$$\mathbb{P}\Big(|X - \mathbb{E}[X]| \geq \varepsilon\mathbb{E}[X]\Big) \leq 2\exp\left(-\frac{\varepsilon^2\mathbb{E}[X]}{3}\right)$$

Here is our result for the burning number of random geometric graphs.

Theorem 66. *There exists a constant c_0 such that for any $c \geq c_0$, if $r \geq cr_c$, then a.a.s. we have that*

$$b(G) = \Theta\left(r^{-2/3}\right),$$

where $G \in \mathscr{G}(n,r)$.

Proof. We start by proving the lower bound as follows. Let $c = \max\{c_0, 32\}$, where c_0 is the constant that satisfies the conditions in Theorem 64, and $r \geq cr_c$. First, starting from the left side of S, we tessellate the area of S with squares of width $s = 4\sqrt{\log n/n}$ which we call *cells*. Since $\frac{1}{s}$ might not be an integer, for the convenience of our computations in the rest of proof, we assume that the rightmost column and the lowest row might consist of slightly larger cells. Figure 4.3 shows such a tessellation for the unit square S with cells of width s or more.

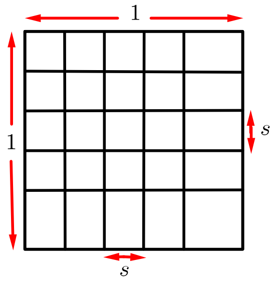


Figure 4.3: A tesllation of the unit square.

We claim that every cell a.a.s. contains at least one node. To prove this, for each cell C, we assign a random variable X_C that counts the number of nodes of G inside C. For each cell C, let E_C represent the event $X_C \leq \mathbb{E}[X_C]/2$. Also, assume that A_C denotes the area of the cell C. By definition of G, since we choose the nodes of

G independently and uniformly at random, then we can see that

$$\mathbb{P}(X_C = i) = \binom{n}{i}(A_C)^i(1 - A_C)^{n-i}.$$

Thus, X_C must be a binomial random variable with parameters n and $p = A_C$. Hence, the expected value of X_C equals $np = nA_C \geq ns^2$. Now, by choosing $\varepsilon = 1/2$ in Lemma 65, we have that

$$\begin{aligned}\mathbb{P}\Big(E_C\Big) = \mathbb{P}\Big(X_C \leq \mathbb{E}\left[X_C\right]/2\Big) &\leq \mathbb{P}\Big(|X_C - \mathbb{E}\left[X_C\right]| \geq (1/2)\mathbb{E}\left[X_C\right]\Big)\\ &\leq 2\exp\left(-\frac{(1/2)^2\mathbb{E}\left[X_C\right]}{3}\right)\\ &\leq 2e^{-(1/2)^2 s^2 n/3}\\ &= 2e^{-(1/2)^2\left(\frac{16\log n}{n}\right)n/3}\\ &= 2e^{-4/3\log n}\\ &= 2n^{-4/3} = o(n^{-1}).\end{aligned}$$

Therefore, we have that

$$\begin{aligned}\mathbb{P}\Big(\bigcap_C E_C^c\Big) &= 1 - \mathbb{P}\Big(\bigcup_C E_C\Big)\\ &\geq 1 - \sum_C \mathbb{P}\Big(E_C\Big)\\ &\geq 1 - \frac{1}{s^2}(2n^{-4/3})\\ &= 1 - \frac{1}{8n^{1/3}\log n}\\ &= 1 - o(n^{-1/3}).\end{aligned}$$

Thus, a.a.s., none of the E_C's occur. In other words, a.a.s., each cell C contains at least $\mathbb{E}\left[X_C\right]/2 + 1$ nodes. Hence, a.a.s., each cell C contains at least one node.

Now, suppose that we choose a sequence of t arbitrary nodes in $V(G)$ such as $(x_1, x_2, \ldots, x_t)$. By Equation (2.1) from Chapter 1, we know that $(x_1, x_2, \ldots, x_t)$ forms a burning sequence for G if and only if $V(G) = N_{t-1}[x_1] \cup N_{t-2}[x_2] \cup \ldots \cup N_0[x_t]$. For $1 \leq i \leq t$, let B_i denote the intersection of S and the ball of radius $r(t-i)$ centred at x_i. As we discussed at the beginning of this section, we know that $N_{t-i}[x_i] \subseteq B_i$. Hence, if

$(x_1, x_2, \ldots, x_t)$ is a burning sequence for G, then we must have $V(G) \subseteq B_1 \cup B_2 \cup \ldots \cup B_t$.

In the following, we want to show that for some specific t (that will be determined in the proof) there exists at least one cell such as C that does not have an intersection with $\bigcup_{i=1}^{t} B_i$. This consequently shows that a.a.s., there is one node (in C) that is not burned by the time t, and therefore, we must have $b(G) > t$. To do so, we find an upper bound for the number of the cells that intersect with the union of B_i's, as follows.

First, note that by assumption, $c \geq 32$. Therefore, it implies that

$$r \geq cr_c > \frac{32}{\sqrt{\pi}} \sqrt{\frac{\log n}{n}} > \frac{32}{2} \sqrt{\frac{\log n}{n}} = 4s.$$

For $1 \leq i \leq t-1$, the radius of B_i is at least r. Since $r > 4s$, and the largest cell in the given tessellation is possibly of length at most $2s$, then we conclude that there must be a cell that falls completely inside B_i. Also, the area of B_i is larger than the area of any cell in the tessellation. Note that we have two types of cells that intersect B_i: those cells that fall completely inside of B_i, called *regular cells*, and those cells that also intersect $S \setminus B_i$, called *boundary cells*.

We claim that for $1 \leq i \leq t-1$, the total number of the cells that intersect B_i is asymptotically at most equal to

$$c_1 \left(\frac{\pi(t-i)^2 r^2}{s^2} \right),$$

where c_1 is a sufficiently large constant that we find in the following. By the above argument, we know that for $1 \leq i \leq t-1$, there is at least one regular cell in B_i. Note that every boundary cell is within 4-cell-distance from a regular cell. Figure 4.4 gives us an intuition of this fact.

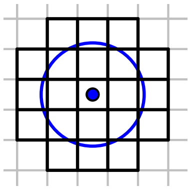


Figure 4.4: Every boundary cell is within 4-cell-distance from a regular cell.

Therefore, if we denote the total number of regular cells with M_i, then the total number of cells that intersect with B_i is at most $25M_i$. On the other hand, since every regular cell falls completely inside of B_i, then it implies that

$$M_i s^2 < |B_i|,$$

where $|B_i|$ is the area of B_i. Let T_i denote the total number of the cells that intersect B_i. Thus, by the above argument we have that

$$\begin{aligned} T_i s^2 &\leq 25 M_i s^2 \\ &\leq 25|B_i|. \end{aligned}$$

Therefore, $T_i \leq 25\left(\frac{|B_i|}{s^2}\right)$. Hence, any constant $c_1 \geq 25$ is sufficient for the truth of the claim. Thus, we have that asymptotically

$$T_i \leq c_1 \left(\frac{\pi(t-i)^2 r^2}{s^2}\right).$$

For $i = t$, the ball B_t is of radius zero; that is, it contains only the point x_t. Hence for $i = t$, the total number of cells that intersect B_t is at most 4, in a case that x_t is a corner point in the given tessellation. Thus, the total number of cells that intersect with at least one of the B_i's, denoted by T, is asymptotically at most

$$T \leq \sum_{i=1}^{t-1} c_1 \frac{(t-i)^2 r^2 \pi}{s^2} + 4 \leq c_1 \frac{t^3 r^2 \pi}{3s^2}.$$

We want to have at least one cell that does not intersect with any of the B_i's. The total number of cells in the tessellation of S is $\left(\lfloor \frac{1}{s} \rfloor\right)^2$. Therefore, if we take $t \leq \left(\frac{2}{c_1 \pi r^2}\right)^{1/3}$, then asymptotically

$$T \leq c_1 \frac{t^3 r^2 \pi}{3s^2} \leq \frac{2}{3s^2} < \left(\left\lfloor \frac{1}{s} \right\rfloor\right)^2.$$

Thus, there must be a cell C that does not intersect with any of B_i's. Consequently, we conclude that burning G in t steps is impossible. Hence, $b(G) > \left(\frac{2}{c_1 \pi r^2}\right)^{1/3}$; in other words, asymptotically $b(G) = \omega(\frac{1}{r^{2/3}})$.

Now, we prove the upper bound using Theorem 64 as follows. First, remember that c is sufficiently large such that the condition in Theorem 64 is satisfied. That is, there exists $c' = c'(c)$ for which

$$d_G(u,v) \leq c' \frac{d_E(u,v)}{r},$$

for every pair of nodes u and v in G. Then we tessellate the unit square S into cells of width $ar^{1/3}$, where $a = (3\sqrt{2}c')^{-1/3}$. Since $\frac{1}{ar^{1/3}}$ may not be an integer, for the convenience of our computations, we assume that the rightmost column and the lowest row might consist of slightly larger cells (similar to the tessellation shown in Figure 4.3). Note that the side lengths of the larger cells would be at most $2ar^{1/3}$, and the area of each cell is at least $a^2r^{2/3}$.

Again we claim that each cell in such a tessellation contains at least one node. To show this, assume that X_C denotes the number of nodes of G that fall in the cell C. As we discussed in the proof of the lower bound, X_C is a binomial random variable with parameters n and $p = A_C$, where A_C is the area of C. Moreover, the expected value of X_C equals $np = nA_C \geq na^2r^{2/3}$. Let E_C represent the event $X_C \leq \mathbb{E}[X_C]/2$, for each cell C. By choosing $\varepsilon = 1/2$ in Lemma 65, we have that

$$\begin{aligned}
\mathbb{P}\Big(E_C\Big) = \mathbb{P}\Big(X_C \leq \mathbb{E}[X_C]/2\Big) &\leq \mathbb{P}\Big(|X_C - \mathbb{E}[X_C]| \geq (1/2)\mathbb{E}[X_C]\Big) \\
&\leq 2e^{-(1/2)^2 \frac{a^2r^{2/3}n}{3}} \\
&\leq 2e^{-(1/12)\left((3\sqrt{2}c')^{-2/3} nC^{2/3} \frac{(\log n)^{1/3}}{(\pi n)^{1/3}}\right)} \\
&= 2e^{-\frac{C^{2/3}}{12(3\sqrt{2\pi}c')^{-2/3}}(n^2 \log n)^{1/3}}.
\end{aligned}$$

Note that the total number of cells in the tessellation is $T = \left(\lfloor \frac{1}{ar^{1/3}} \rfloor\right)^2 \leq \frac{1}{a^2r^{2/3}}$. Let $k = \frac{1}{a^2r^{2/3}}$ (which is not necessarily an integer, but it is a bound on T). Therefore, we have that

$$\begin{aligned}
\mathbb{P}\Big(\bigcap_C E_C^c\Big) = 1 - \mathbb{P}\Big(\bigcup_C E_C\Big) &\geq 1 - \sum_C \mathbb{P}\Big(E_C\Big) \\
&\geq 1 - \frac{1}{a^2r^{2/3}} \mathbb{P}\Big(E_C\Big) \\
&\geq 1 - \frac{(3\sqrt{2}c')^{2/3}}{(cr_c)^{2/3}} \mathbb{P}\Big(E_C\Big).
\end{aligned}$$

Since $\mathbb{P}\big(E_C\big) \leq 2e^{-\frac{C^{2/3}}{12(3\sqrt{2\pi}c')^{-2/3}}(n^2 \log n)^{1/3}}$, and by replacing r_c in the above inequality, we conclude that

$$\mathbb{P}\Big(\bigcap_C E_C^c\Big) \geq 1 - 2\left(\frac{3\sqrt{2\pi}c'}{c}\right)^{2/3}\left(\frac{n}{\log n}\right)^{1/3} e^{-\frac{c^{2/3}}{12(3\sqrt{2\pi}c')^{-2/3}}(n^2 \log n)^{1/3}}.$$

Note that in the last inequality above, both terms $\left(\frac{n}{\log n}\right)^{1/3}$ and $(n^2 \log n)^{1/3}$ (inside the exponential function) go to infinity as n grows. However, since the term $\left(\frac{n}{\log n}\right)^{1/3}$ asymptotically is much smaller than the term $(n^2 \log n)^{1/3}$, then it implies the whole expression in the last line approaches to 1, as n goes to infinity. Therefore, a.a.s., none of the E_C's occur. In other words, a.a.s., each cell C contains at least $\mathbb{E}\left[X_C\right]/2 + 1$ nodes. Hence, a.a.s., each cell C contains at least one node.

Note that the node set of G is covered by $T \leq k$ cells, denoted by $C_1, C_2, \ldots, C_T$. Assume that for $1 \leq i \leq T$, we choose a node in each cell C_i, called u_i (we know by the above argument that such a node does exist). Now, since the maximum side length of a cell is at most $2ar^{1/3}$, then the maximum Euclidean distance between any two points in a cell C_i is at most $2\sqrt{2}ar^{1/3}$. Thus, by Theorem 64, for any node such as v in the cell C_i, we have that

$$d_G(u_i, v) \leq c' \frac{d_E(u_i, v)}{r} \leq 2\sqrt{2}c'ar^{-2/3} \leq k.$$

This is true since $2\sqrt{2}c'ar^{-2/3} \leq k = \frac{1}{a^2 r^{2/3}}$ if and only if $2\sqrt{2}ac' \leq \frac{1}{a^3} = 3\sqrt{2}c'$, which is correct. Therefore, $\{N_k[u_i]\}_{i=1}^T$ is a covering for the node set of G. Hence, by Theorem 4 from Chapter 2, we conclude that $b(G) \leq T + k \leq 2k$. Thus, the upper bound follows. □

4.2 Poisson Burning Process

In this section we define the Poisson burning process and we consider this process for the graphs in general, and for the paths in particular. We start by providing some background and terminology.

A *Homogeneous Poisson process* with parameter λ is a counting (or point) stochastic process that counts the number of events that occur in the given intervals of time.

We denote a Poisson process by $\{N(t)\}_{t\geq 0}$, where the parameter t represents the time, and $N(t)$ denotes the number of events that happen in the time interval $[0,t]$. The sequence $\{N(t)\}_{t\geq 0}$ satisfies the following properties:

(i) $N(0) = 0$.

(ii) The number of events in non-overlapping time intervals are independent.

(iii) For a positive real number h, we have that

$$\mathbb{P}\left(N(t+h) - N(t) = 1\right) = \lambda h + o(h),$$

and

$$\mathbb{P}\left(N(t+h) - N(t) \geq 2\right) = o(h).$$

The properties (i), (ii), and (iii) imply that

$$\begin{aligned}\mathbb{P}\left(N(t+h) - N(t) = k\right) &= \mathbb{P}\left(N(t) - N(0) = k\right)\\ &= \mathbb{P}\left(N(t) = k\right).\end{aligned}$$

Let T_1 denote the time of the occurrence of the first event in a Poisson process with parameter λ, and for $i \geq 2$, suppose that T_i denotes the time that we wait after the occurrence of the $(i-1)$-th event until the i-th event happens. Since the occurrence of each event happens randomly, each T_i is a random variable that is called the *i-th inter-arrival time*. The time that the i-th event occurs is called the *i-th waiting time* and is denoted by S_i. Thus, by definition, $S_1 = T_1$, and for $j \geq 2$, $S_j = \sum_{i=1}^{j} T_j$. Moreover, for $j \geq 1$, $T_j = S_j - S_{j-1}$.

It is known in probability theory that for every natural number n, the random variables $T_1, T_2, \ldots, T_n$ are independent and identically distributed exponential random variables with rate λ; that is, the probability density function of each T_i is defined by the function $f(x) = \lambda \exp(-\lambda x)$, where $x \geq 0$. Hence, the expectation or mean of each T_i equals $\frac{1}{\lambda}$, and its variance equals $\frac{1}{\lambda^2}$. Moreover, each S_n is a gamma random variable $\Gamma(n,\lambda)$; that is, the probability density function of S_n is defined by $f(x) = \frac{\lambda^n x^{n-1}\exp(-\lambda x)}{\Gamma(\lambda)}$, for $x \geq 0$. In this formula, $\Gamma(\lambda)$ is the *gamma function* that is defined as follows:

$$\Gamma(t) = \int_0^\infty x^{t-1}\exp(-x)dx,$$

for $t > 0$. In fact, the gamma function is an extension of the factorial function; that is, for a positive integer n, we have that

$$\Gamma(n) = (n-1)!$$

Also, $N(t)$ is a Poisson random variable with rate λt; that is, the probability distribution function of S_n is defined by

$$\mathbb{P}(N(t) = i) = \frac{\exp(-\lambda t)(\lambda t)^i}{i!},$$

where $i \geq 0$ is an integer.

Hence, each Poisson process is uniquely determined by one of the sequences $\{N(t)\}_{t\geq 0}$, or $\{T_i\}_{i=1}$, or $\{S_i\}_{i=1}$. We can also see that the event $N(t) < i$ happens if and only if the event $S_i > t$. Thus, we have that

$$\mathbb{P}(N(t) < i) = \mathbb{P}(S_i > t). \tag{4.1}$$

For more details on Poisson process see [52]. The following concentration inequality is called the *Chebyshev's inequality*; see [53].

Theorem 67 ([53]). *Suppose that X is a random variable with finite mean μ, and finite non-zero variance σ^2. Then for any $t > 0$, we have that*

$$\mathbb{P}\left(|X - \mu| \geq t\right) \leq \frac{\sigma^2}{t^2}.$$

Now, we are ready to define the *Poisson graph burning process* as follows. Suppose that we consider the burning process for a graph G of order n, with $b(G) = k$. However, the time for choosing the i-th source of fire is the waiting time for the i-th event in a Poisson process with parameter λ, and we choose the i-th source of fire uniformly at random to be any node in G. We continue this until at some time t the whole graph G is burning. We may or may not consider to continue the Poisson process after time t as every node in G will be burning after t. Such a burning process on G is called a Poisson burning process for G. We call the time t at which the whole graph G is burning in a Poisson burning process the *burning time* of G, and we denote it by

$b_{p,\lambda}(G)$, or simply by $b_p(G)$ when we know the parameter λ.

Note that in a regular random burning process for G, we choose the i-th source of fire at time $t = i$. However, in a Poisson burning process the time for choosing the i-th source of fire can be any time t either before i or after i. Moreover, here $b_p(G)$ is a random variable as it depends on the Poisson process, and also on the way that we choose the fire sources. Our goal is here to consider the asymptotic behaviour of $b_p(G)$.

Theorem 68. *If G is a graph of order n with $b(G) = k$, then in a Poisson burning process with $\lambda = 1$ on G we have that*

$$b_p(G) = O(k\sqrt{\log n}).$$

Proof. Assume that $(x_1, x_2, \ldots, x_k)$ is a fixed burning sequence for G (in the regular burning process), and $t = k + k\sqrt{\log n}$. In a Poisson burning process for G let A be the event that "all the nodes in G are burning at time t". Also, let B be the event "$\bigcap_{i=1}^{k} A_i$", where A_i is the event that "$S_i \leq t - (k-i)$ and the node x_i is burned in the i-th step" (or equivalently, $t - S_i \geq (k-i)$ and the node x_i is burned in the i-th step), for $1 \leq i \leq k$. We can see that B is a special case of the event A (or equivalently, $B \subseteq A$), and hence, $\mathbb{P}(B) \leq \mathbb{P}(A)$. Namely, if B occurs, then it implies that $S_k \leq t$, and for $1 \leq i \leq k$, the node x_i is burning at time $t - (k-i)$. Therefore, $\{N_{t-S_i}[x_i]\}_{i=1}^{k}$ forms a covering for the node set of G. Hence, G must be burning at time t in such a case.

Thus, if we show that $\lim_{n\to\infty} \mathbb{P}(B) = 1$, then it implies that $\lim_{n\to\infty} \mathbb{P}(A) = 1$. Consequently, it shows that a.a.s. $b_p(G) \leq t$. For this, it suffices to prove that $\lim_{n\to\infty} \mathbb{P}(\overline{B}) = 0$. In the rest of the argument, we will try to prove this by Chebyshev's inequality.

Assume that $\lambda = 1$. Therefore, $\mathbb{E}[T_i] = \mathrm{Var}(T_i) = 1$, for each $1 \leq i \leq k$. Note that $S_i = \sum_{j=1}^{i} T_j$. Since T_j's are independent exponentially distributed random variables, then for $1 \leq i \leq k$,

$$\mathrm{Var}(S_i) = \sum_{j=1}^{i} \mathrm{Var}(T_j) = i = \left(\sqrt{i}\right)^2.$$

Moreover,

$$\mathbb{E}\left[S_i\right] = \sum_{j=1}^{i} \mathbb{E}\left[T_j\right] = i.$$

Thus, by Chebyshev's inequality, for $1 \le i \le k$, we have that

$$\begin{aligned} \mathbb{P}(\overline{A_i}) &= \mathbb{P}(S_i > t - (k - i)) \\ &= \mathbb{P}\left(S_i - i > k\sqrt{\log n}\right) \\ &\le \mathbb{P}\left(\left|S_i - \mathbb{E}\left[S_i\right]\right| \ge k\sqrt{\log n}\right) \\ &\le \frac{i}{k^2 \log n}. \end{aligned}$$

Since $B = \bigcap_{i=1}^{k} A_i$, then we have that

$$\begin{aligned} \mathbb{P}(\overline{B}) &= \mathbb{P}(\bigcup_{i=1}^{k} \overline{A_i}) \\ &\le \sum_{i=1}^{k} \mathbb{P}(\overline{A_i}) \\ &\le \sum_{i=1}^{k} \frac{i}{k^2 \log n} \\ &= \frac{k(k+1)}{2k^2 \log n}. \end{aligned}$$

Therefore, we conclude that

$$\lim_{n\to\infty} \mathbb{P}(\overline{B}) = \lim_{n} \frac{k(k+1)}{2k^2 \log n} = 0.$$

Thus,

$$\mathbb{P}(A) = \mathbb{P}(B) = 1,$$

a.a.s.; that is, a.a.s., all the nodes of G are burning at time $t = k + k\sqrt{\log n}$. Since $k \le k\sqrt{\log n}$, then the proof follows. □

Note that the whole probability argument above is only dependent on the Poisson process and a fixed optimum burning sequence for G. Since by Corollary 27 from Chapter 2, the burning number of every connected graph G of order n, is of order

$O(\sqrt{n})$, then we conclude the following result on the Poisson burning for $b_p(G)$.

Theorem 69. *If G is a connected graph of order n, then in a Poisson burning process with $\lambda = 1$ on G we have that*

$$b_p(G) = O(\sqrt{n \log n}).$$

4.3 Poisson Burning for Paths

In this section, we consider the Poisson random burning process for paths, and we find the asymptotic order of $b_p(P_n)$.

Theorem 70. *In a Poisson burning process with $\lambda = 1$ on the path P_n we have that*

$$b_p(P_n) = \Theta(\sqrt{n \log n})$$

Proof. Suppose that we burn path P_n according to a Poisson burning process with mean λ. Assume that we decompose path P_n into smaller paths of order d, where $d \leq n$ is a positive integer. Then we can decompose the current Poisson process on P_n into $N = \lceil \frac{n}{d} \rceil$ disjoint Poisson processes each with mean $\frac{1}{N}\lambda$, since the smaller paths are all disjoint.

Intuitively, if we choose d in a way that, with high probability, there is no fire source chosen from each smaller path of order d at time $t_0 = \sqrt{n \log n}$, then it implies that asymptotically almost surely, $b_p(P_n) > t_0$.

For $1 \leq i \leq N$, let A_i be the event "no fire source is chosen from the i-th path (in the given partition of P_n) at time t_0". Equivalently, A_i is the event "$T_1 > t_0$". By taking $\lambda = 1$ and $d = \lceil \frac{\sqrt{n}}{\log n} \rceil$, we will have that

$$\begin{aligned}
\mathbb{P}(A_i) &= \mathbb{P}(T_1 > t_0) \\
&= \int_{t_0}^{\infty} \frac{1}{N} e^{-\frac{1}{N}x} dx \\
&= e^{-\frac{t_0}{N}}.
\end{aligned}$$

Note that

$$\lim_{n\to\infty} e^{-\frac{t_0}{N}} = \lim_{n\to\infty} e^{-\frac{t_0}{\lceil \frac{n}{d} \rceil}} \geq \lim_{n\to\infty} e^{-\frac{\left(\sqrt{n\log n}\right)\left\lceil \frac{\sqrt{n}}{\log n} \right\rceil}{n}} = 1.$$

Therefore, we conclude that a.a.s., there is no fire started at any of the small paths of order d in the partition for P_n. This implies that a.a.s. $b_p(P_n) \geq t_0 = \sqrt{n\log n}$. On the other hand, by Theorem 69, we have the upper bound. Thus, we conclude that $b_p(P_n) = \Theta(\sqrt{n\log n})$. □

Chapter 5

The Competitive Diffusion Game in Classes of Graphs

In this chapter, we consider a game on graphs that is based on a model for the spread of influence in social networks. Indeed, this game was the first motivation for us to define the burning problem (as a one player game). First, we explain some preliminaries for the results in this chapter, the model and the game.

A *game* consists of a set of N players denoted by $[N] = \{1, 2, \ldots, N\}$ (for a positive integer N), a set of *strategy profiles* $S = S_1 \times S_2 \times \cdots \times S_N$, where S_i is the strategy set of player i, and a pay-off function defined on S, defined by $f(x) = (f_1(x), f_2(x), \ldots, f_N(x))$, where $x \in S$ and $f_i(x)$ describes the out-come of player i for strategy x. In such a game, a *strategy* for each player is a plan that is determined in the beginning of each play of the game, and describes what the player does at every possible situation. The *strategy space* or *strategy set* of each player is the set of all possible strategies available for the player in the game. A *strategy profile* x in a N-player game, is a vector $x = (x_1, x_2, \ldots, x_N)$ in which $x_i \in S_i$ denotes the strategy of the i-th player, for $1 \leq i \leq N$.

In 2009, Alon *et al.* [2], introduced a new model for the competitive diffusion process in social networks. Their approach is a novel way of modelling the spread of influence as a game, where the aim of this game is to influence users in the network through "infection" with a particular brand, spreading through the links of the network. In other words, suppose that we have a set of firms that want to advertise their products. Initially they target a small group of people that they hope will extend into a larger group of society. Any individual, who has learned about a product brand from one of these firms first, either directly or through a social link, will be biased in favour of that brand. However, if a user is getting the influence from different products, she becomes confused and we cancel her out of the game. The gain of each firm is the total number of users that at the end of the diffusion process are biased towards its brand. In the language of mathematics, we can model this competitive propagation process as a game on an undirected finite graph, representing the underlying graph of a social network, in which the users form the node set of the graph, and the agents are the players. Also, the product of each agent is denoted by a distinct colour.

We now provide a formal definition of the game based on the competitive diffusion model. A game $\Gamma = \langle G, N \rangle$ as a discrete-time deterministic process is played on a graph G with a set of N players. We identify each player with a distinct number i, where $1 \leq i \leq N$. The strategy space of each player is the set $V(G)$. That is, at time $t = 0$, each player i, where $1 \leq i \leq N$ selects a single node and colours it in colour i, and the rest of the nodes are uncoloured. If two or more agents select the same node at time $t = 0$, then that node becomes gray, and those players automatically leave the game. If S_t is the set of coloured nodes at time $t \geq 0$, then at time $t + 1$ each player colours the uncoloured nodes in the neighbourhood of S_t. If v is an uncoloured node that has coloured neighbours only in colour i, then v receives colour i. If v is an uncoloured node that has coloured neighbours with different colours, then it becomes gray. The players continue until no one can colour any uncoloured node. At the end, the *pay-off* of the i-th player is the number of nodes in G with colour i. Note that in this game, after choosing the strategies of the players everything in the process is deterministic. Hence, in the rest of the chapter, we represent each play of the game by the corresponding strategy profile of the players in that play.

As an example, let G be a graph as shown in Figure 5.1, and take $N = 2$.

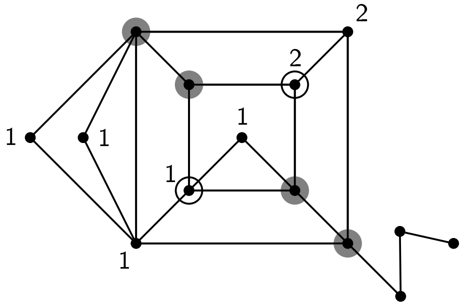


Figure 5.1: An example of the two-player game on a graph.

If the first player with colour 1, and the second player with colour 2, choose the two nodes with the circles around them at the beginning, then the pay-off of them will be the number of the nodes which are indexed by 1 and 2 in the figure, respectively.

As we can see in Figure 5.1, there are four nodes that become gray by the rules of the game, and three nodes which are not reachable by any player and therefore, remain uncoloured at the end.

Note that throughout this game, it is as if we delete all the gray nodes, so the metric of the graph is changing within the rounds of the game. This is unlike the Voronoi games [24], in which the gain of each agent is the number of individuals whose distance to the agent representative is less than the other agents.

In real networks finding a kind of stable situation in which every agent is satisfied is called a Nash-equilibrium. This is often of more interest than finding the winner of the game. In a N-player game, a *pure Nash-equilibrium* is a strategy profile such as $x = (x_1, x_2, \ldots, x_N)$ in which the strategy of each player is the best against the strategies of the others. That is, for every $1 \leq i \leq N$, and for each $x'_i \in S_i$,

$$f_i(x) \geq f_i(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_N).$$

In other words, in a pure Nash-equilibrium $(x_1, x_2, \ldots, x_N)$ by choosing x_i the i-th player has maximized her pay-off with respect to the strategy of the other players in this profile; that is, no player can gain more by changing only her own strategy unilaterally. In the rest of this chapter, we simply use the word Nash-equilibrium instead of pure Nash-equilibrium. Also, note that we do not care about the order of players in this game, as they are supposed to choose their strategies simultaneously at the beginning of the game. In other words, we use the numbers in the set $[N]$ only to distinguish the players not to order them. Hence, in some of the results, for instance in Theorem 77, we say that there is a unique Nash-equilibrium (c_1, c_2), where c_1 and c_2 are the centroidal nodes of the graph T, as we do not consider (c_1, c_2) different from (c_2, c_1). For further information about game theory concepts we refer the reader to [9].

Note that not every game necessarily admits a Nash-equilibrium in its strategy profile. For example, assume that we have the 3-player case of the competitive diffusion game on a path of order 6. Then there can not be any Nash-equilibrium for this game, as we see in Theorem 89.

Alon *et al.* [2] in their paper, proved the existence of Nash-equilibriums for the game on graphs of diameter 2, and gave an example of a graph with diameter more than 2 which does not admit a Nash-equilibrium in the two-player case of the game. Takehara *et al.* [57] however, provided a counter example with a graph of diameter 2 that does not admit a Nash-equilibrium, and presented a restatement of the theorem (about graphs with diameter at most 2) in [2] by putting some restrictions on the graph structure. Recently, Small and Mason [55] considered the existence of Nash-equilibriums for the two-player game on trees, and also for the ILT model of online social networks, with focus on utility functions. In [39], Janssen and Vautour consider some results on the safe strategies (see Definition 3.1.3 from [9]) of the game on trees. In [10], Etesami and Basar show that the problem of determining whether there exist a Nash-equilibrium or not for this game is **NP**-hard. They also prove that there is a Nash-equilibrium for the two-player case of the game on lattices and hypercubes.

In this thesis, we will consider the above game for some families of graphs. However, we take a novel approach based on the graph properties of these families. In Section 5.1, we prove the existence of Nash-equilibriums for the special two-player case of the game for trees, paths, cycles, and unicyclic graphs. Our proof for trees is much simpler and shorter compared to previous work [55]. In Section 5.2, we show that the two-player game on the Cartesian grids always admit a Nash-equilibrium. In Section 5.3, we consider the game with at least three players. Throughout this chapter, we assume that the graphs are connected.

5.1 Trees, Paths, Cycles, and Unicyclic Graphs

In this section, we consider some simple facts about the game, and use them to find Nash-equilibriums for different known families of graphs. The following definitions help us to describe the obtained results.

Assume that we are playing the game on a graph G. Suppose that in some round of the game there is a shortest path $P : u, v_1, \ldots, v_{n-1}, v$ between two distinct nodes u and v such that the only coloured nodes of P are u and v with two different colours. If for each $1 \le i \le n-1$, and every coloured node $w \notin \{u, v\}$, $\min\{d(v_i, u), d(v_i, v)\} < d(v_i, w)$, then we call path P a *blocked path* induced by the nodes u and v, or simply, a blocked path. In other words, in a blocked path P, the closest coloured nodes to each node in P are the end points of P.

We need the following lemma to find a better understanding of the dynamic of a path between a pair of nodes with different colours throughout the game.

Lemma 71. *Suppose that we have a k-player game on a graph G. If P is a blocked path of length n induced by nodes u and v in G, such that u is coloured by the i-th player and v is coloured by the j-th player, where $1 \leq i < j \leq k$. Then each of the i-th and the j-th players wins the first $\lfloor (n+1)/2 \rfloor$ nearest nodes in path P by the end of the game. Moreover, if the length of P is even, then the middle node of P becomes gray.*

Proof. Since by assumption, the closest coloured nodes to any node in P are u and v, then the nodes in P will be coloured only by the colours of u or v, or they may become gray throughout the rounds of the game. Without loss of generality, assume that the colour of u is 1 and the colour of v is 2. By definition of the game, we can see that a node $z \in P$ receives colour 1 from u if and only if $d(z,u) < d(z,v)$. Moreover, z becomes gray if and only if $d(z,u) = d(z,v)$. Therefore, each of the i-th and the j-th players wins the first $\lfloor (n+1)/2 \rfloor$ nearest nodes in path P by the end of the game. If the length of P is even, then the middle node of P becomes gray. □

Before going through the next results, note that the following lemma is true for any graph.

Lemma 72. *If we play the k-player game on a graph G of order n, where $n \leq k$, then there is always a Nash-equilibrium.*

Proof. Assume that $V(G) = \{u_1, u_2, \ldots, u_n\}$. Then we define a strategy profile $(x_1, x_2, \ldots, x_k)$ as follows. For each $1 \leq i \leq n$, we take $x_i = u_i$. For $n \leq i \leq k$, we define $x_i = u_j$ in which j is the smallest integer that is congruent to i mod n. We claim that $(x_1, x_2, \ldots, x_k)$ forms a Nash-equilibrium. Note that by definition, every node is a strategy that is either achieved by only one player, or becomes gray, when at least two players have selected that node. Hence, the gain of each player is either one or zero, accordingly. If any player changes her strategy to a non-gray node, then the new node (strategy) will become gray, as it was selected by another player before. If a player changes her strategy to any gray node, then she does not gain anything. Thus, in such a case no one can increase her pay-off by changing her strategy. Therefore, $(x_1, x_2, \ldots, x_k)$ is a Nash-equilibrium for this game. □

A node v of a graph G is called a *cut node* if removing v from G results in a graph which is not connected. An edge uv is a *cut edge* if deletion of uv from G is a disconnected graph. The following lemma is useful for some of the results as we will see later on.

Lemma 73. *Suppose that graph $G = G_1 \cup G_2$ is the union of two subgraphs G_1 and G_2 such that $V(G_1) \cap V(G_2) = \{v\}$, where v is a cut node of G. Then any possible Nash-equilibrium of the two-player game on G consists of either two nodes in G_1 or two nodes in G_2.*

Proof. Suppose that (u_1, u_2) is a strategy profile such that $u_1 \in V(G_1) \setminus V(G_2)$ and $u_2 \in V(G_2) \setminus V(G_1)$. Note that by assumption $u_1 \neq v$ and $u_2 \neq v$. There are three possibilities: (i) the first player wins the node v, (ii) the second player gains v, or (iii) the node v becomes gray throughout the game. We claim that each player by changing her strategy to v will increase her pay-off. To show this, first we consider the case in which the first player changes her strategy to v. Let P_1 be the set of the nodes that the first player gains in (u_1, u_2), and P_1' denote the set of the nodes that she wins in game (v, u_2).

If case (i) happens, then it implies that the first player now gains v in round $t = 0$, while before she was gaining v in a round $t \geq 1$. Thus, not only P_1' contains all the nodes in G_1, but also she gains the nodes in $P_1 \cap V(G_2)$ in at least one round earlier than before. Hence, $P_1 \subset P_1'$.

If case (ii) or (iii) happens, then it implies that $P_1 \subset G_1 \subseteq P_1'$, since P_1 does not contain v while P_1' contains v.

Hence, (u_1, u_2) can not be a Nash-equilibrium. By symmetry, with a similar argument, we prove that the second player also can increase her pay-off by choosing v instead of u_2. Thus, such a (u_1, u_2) can not be a Nash-equilibrium in any case. Therefore, in any possible Nash-equilibrium strategies of the players should either both be in G_1 or both be in G_2. □

Assume that P is a path of order n with centre C. If n is odd, then C consists of the node in the middle of P that is of equal distance with respect to the end points. If n is even, then C consists of two adjacent nodes in the middle of P such that deleting them from P results in two isomorphic paths. We now state and prove our first result on the competitive diffusion game for paths.

Theorem 74. *Suppose that we play the two-player game on a path P of order n with centre C.*

(i) If n is even, then the two adjacent central nodes in C form the only possible Nash-equilibrium of this game, and the equilibrium pay-offs are equal to $\frac{n}{2}$.

(ii) If n is odd and $N(v) = \{u_1, u_2\}$, where $C = \{v\}$, then the set of the Nash-equilibriums of this game consists of (u_1, v) and (u_2, v). Moreover, the equilibrium pay-offs are $\frac{n+1}{2}$ and $\frac{n-1}{2}$.

Proof. If $n = 1$ or $n = 2$, then by Lemma 72, the theorem statement is true. Hence, assume that P is a path of order $n \geq 3$. Note that any non-pendant node of P is a cut-node. First, we claim that in a possible Nash-equilibrium, the strategies of the players must be two adjacent nodes. In other words, if the strategy of one of the players is node u, then the best strategy for her opponent with respect to v is to choose a node adjacent to v which separates v from a larger number of the nodes in P.

Namely, suppose that each player $i \in \{1, 2\}$ chooses node u_i as her strategy. If (u_1, u_2) forms a Nash-equilibrium, and u_1 and u_2 are not adjacent, then there must be a node $v \notin \{u_1, u_2\}$ in the path between u_1 and u_2. Since v is a cut-node, we find a contradiction by Lemma 73. Therefore, such a strategy profile (u_1, u_2) is not a Nash-equilibrium.

However, if the players choose two adjacent nodes as their strategies which are not selected as in (i) or (ii), then the player who is closer to one of the end points can improve her pay-off by changing her strategy to another neighbour of her opponent's strategy. Hence, such a case is not a Nash-equilibrium.

Finally, if they both choose their strategies as in (i) or (ii), then no one can improve her pay-off by changing her strategy. Therefore, (i) and (ii) form the only possible Nash-equilibriums of this game. □

Theorem 75. *In a two-player game on cycle C_n we have the following statements.*

(i) If n is odd, then every two nodes on C_n form a Nash-equilibrium, and the pay-offs are equal to $\frac{n-1}{2}$.

(ii) If n is even, then two nodes on C_n form a Nash-equilibrium if and only if they are of odd distance, and the equilibrium pay-offs are equal to $\frac{n}{2}$.

Proof. When we have a two player game on a cycle C_n, the strategies of the players divide the cycle into two blocked paths. If n is odd, then one of the blocked paths is always of odd length and the other one is of even length. Hence, by Lemma 71, every player wins $(n-1)/2$ nodes, and one node in the middle of the even subpath becomes gray. Since this happens for any selection of the nodes, then any two nodes form a Nash-equilibrium when n is odd.

If n is even, then the two blocked paths are both even or odd. If they are both of odd length, then by Lemma 71, each player wins exactly half of the nodes on C_n, and no one can improve this. If the blocked paths are both of even length, then every player wins $(n/2)-1$, and one of the nodes in each blocked path becomes gray. Thus, each player can improve her pay-off by changing her strategy to a neighbour of her first strategy. Hence, two nodes of C_n form a Nash-equilibrium if and only if they are of odd distance. □

A maximal subtree which contains a node v of a tree T as a leaf is called a *branch* of T at v. The *weight* of a node v of T, denoted by $wt(v)$ is the maximum number of nodes in a branch at v (not including v). A node u is a *centroid node* of T if it has the minimum weight among all nodes. The *centroid* of T is the set of all centroid nodes of T. The following theorem about the centroid of trees is known in graph theory.

Theorem 76 ([59]). *If C is the centroid of a tree T of order n, then we have the following statements.*

(i) C consists of either a single node or two adjacent nodes. Moreover, $wt(u) \leq \frac{n}{2}$ if and only if $u \in C$.

(ii) If $C = \{c_1, c_2\}$, then $wt(c_1) = wt(c_2) = n/2$.

(iii) $C = \{c\}$ if and only if $wt(c) \leq (n-1)/2$.

Note that according to case (i) of the above theorem, if $v \notin C$, then $wt(v) > n/2$. In the following theorem, we use centroidal nodes and we apply Theorem 76 to find the Nash-equilibriums of the two-player game on a tree.

Theorem 77. *In a two-player game on a tree T of order n with centroid C, we have the following statements.*

(i) If $C = \{c_1, c_2\}$, then (c_1, c_2) is the unique Nash-equilibrium of the game, and the equilibrium pay-offs are equal to $n/2$.

(ii) If $C = \{c\}$, and v is a neighbour of c in a branch with maximum weight attached at c, then (c, v) forms a Nash-equilibrium. Moreover, any Nash-equilibrium for this game consists of such two nodes.

Proof. Suppose that for $i \in \{1, 2\}$, u_i is the strategy of the i-th player, and g_i denotes her pay-off. Also, assume that (u_1, u_2) forms a Nash-equilibrium for this game. Note that there exists a unique path between any two nodes in a tree. First, we claim that (u_1, u_2) must be two adjacent nodes such that $g_1 = wt(u_2)$ and $g_2 = wt(u_1)$. Namely, if u_1 and u_2 are not adjacent, then there must be a node $v \notin \{u_1, u_2\}$ in the path between u_1 and u_2. Since v is a cut-node, then we find a contradiction by Lemma 73. Now, let e be the edge between u_1 and u_2 in T. We can see that the pay-off of the first player is a subset of a branch attached at u_2 called B_2 which contains e. Thus, we have that

$$g_1 \leq |B_2| \leq wt(u_2), \tag{5.1}$$

and the equality holds only if u_1 is the neighbour of u_2 in a branch with maximum weight attached at u_2. Namely, if u_1 is in a branch at u_2 that its weight does not equal $wt(u_2)$, then the inequality (5.1) will be strict. Similarly, we can see that the gain obtained by choosing u_2 is a subset of a branch attached at u_1 called B_1 which contains e. Therefore, $g_2 \leq wt(u_1)$, and the equality holds only if u_2 is the neighbour of u_1 in a branch with maximum weight attached at u_1. Therefore, the best strategy for each player is to choose a node adjacent to the strategy of her opponent that is in a branch with maximum weight attached to it. Hence, in a possible Nash-equilibrium the strategies of the players must be adjacent, and $g_1 = wt(u_2)$ and $g_2 = wt(u_1)$.

Now, we assume that u_1 and u_2 are adjacent nodes with $g_1 = wt(u_2)$ and $g_2 = wt(u_1)$. We claim that at least one of them must be in C. By contradiction, suppose that both u_1 and u_2 are not in C. Thus, by Theorem 76, $wt(u_1) > \frac{n}{2}$, and $wt(u_2) > \frac{n}{2}$. Note that $g_1 + g_2 \leq n$, since we may have some gray nodes. Therefore, we conclude that $wt(u_1) = g_2 \leq n - g_1 < \frac{n}{2}$, which is a contradiction. Therefore, in a possible Nash-equilibrium at least one of the players' strategies must be in C.

Hence, without loss of generality, assume that $u_1 \in C$. Thus, by Theorem 76, $wt(u_1) = \frac{n}{2}$. By Theorem 76, we also know that either C consists of two adjacent nodes, or C contains a single node. If C consists of two adjacent nodes, and u_2 is not in C, then by Theorem 76, $g_1 = wt(u_2) > \frac{n}{2}$. Therefore, we conclude that $wt(u_1) = g_2 \leq n - g_1 < \frac{n}{2}$, which is a contradiction. On the other hand, if $\{u_1, u_2\} = C$, then each player has the best strategy with respect to the strategy of her opponent. Thus, in such a case, C forms the only Nash-equilibrium of this game. If C contains a single node, then by the above argument, the best strategy for the second player is to choose a node adjacent to $u_1 \in C$ that is in a branch with maximum weight at u_1. Also, note that in such a case, the strategy of each player is the best with respect to the strategy of her opponent. Hence, if $C = \{c\}$, then (u_1, u_2) forms a Nash-equilibrium if and only if $u_1 = c$ and u_2 is a node adjacent to c in a branch with maximum weight at c. □

Suppose that G is a *unicyclic* graph, that is, G has only one cycle C. We can see that $G \smallsetminus C$ is a forest such that each component of this forest is adjacent to exactly one node on C. For each node $v \in C$, if there are $t = d(v) - 2 > 0$ different components in $G \smallsetminus C$ that are connected to v, then for each of these trees, we label the union of the tree together with v and the edge connecting v to the tree by T_{iv}, for $1 \leq i \leq t$; that is, all T_{iv}'s share v.

Suppose that we have a two-player game on a unicyclic graph G with cycle C. By the above definition, we can assume that every node v on C has a weight $wt_C(v) = |V(\bigcup_{i=1}^{d(v)-2} T_{iv})|$. As we will see, sometimes we play the game on the weighted cycle C (instead of G) with a slightly different rule; that is, we only choose the nodes on C, and the gain of each player after taking node $v \in C$ is increased by the weight of v. In such cases, we replace G by C_W. Hence, when we refer to C_W, we mean that instead of playing on G we are playing the game only on C with weighted nodes.

We use the above notations for the results on unicyclic graphs. We use the following lemma, which is an immediate result of Theorem 76, to prove the next theorem.

Lemma 78. *Suppose that T is a tree with centroid C. Then for any node u which is not in C the maximum branch attached at u is the one that contains C (which is the only branch attached at u with weight more than $\frac{n}{2}$).*

Proof. Suppose that u is a node in T that is not in C. We want to show that any branch attached at u which does not contain C is not maximum. We know that there is a unique path between u and the closest node in C which goes through a branch B attached at u. Hence, B contains C. Now, assume that $B' \neq B$ is a branch attached at u. We can see that B' is a proper subset of a branch attached at a node c in C. Therefore, by Theorem 76 part (i), we have that

$$|B'| < wt(c) \leq \frac{n}{2}.$$

On the other hand, by Theorem 76 part (i), we know that the weight of any node outside C is greater than $n/2$. Since u is not in C, then by the above argument, we conclude that B must be the maximum branch attached at u. □

In general, we have two possibilities for a unicyclic graph G with cycle C; either there is a node v in C with $|T_{iv}| \geq \frac{n}{2} + 1$, for some $1 \leq i \leq d(v) - 2$, or $|T_{iv}| \leq \frac{n}{2}$ for all $v \in C$, and $1 \leq i \leq d(v) - 2$. Hence, we have the following theorem.

Theorem 79. *Suppose that G is a unicyclic graph with cycle C. If there is a node v on C with $|T_{iv}| \geq \frac{n}{2} + 1$, for some $1 \leq i \leq d(v) - 2$, then there exists a Nash-equilibrium with both players playing on T_{iv}. Otherwise, if there exists a Nash-equilibrium for this game, then it must consist of a set of two nodes either on C or on a T_{iv}, for some $v \in C$ and $1 \leq i \leq d(v) - 2$.*

Proof. First, suppose that there is a node v on C with $|T_{iv}| \geq \frac{n}{2} + 1$, for some i. We show that in a possible Nash-equilibrium (u_1, u_2), both u_i's must be in T_{iv} as follows. If none of u_i's are in T_{iv}, then the player with the smaller gain can improve her pay-off by changing her strategy to v. Namely, this way she wins more than half of the nodes in G. Hence, in a possible Nash-equilibrium for this game, at least one of the players must choose her strategy on a node in T_{iv}.

Note that node v has a unique neighbour in T_{iv}, called v'. Without loss of generality, suppose that $u_1 = v$, and u_2 is in $G \setminus T_{iv}$. Then the second player can improve her pay-off by changing her strategy to v'. To see this, suppose that g_1 and g_1' denote her pay-off before and after changing her strategy, respectively. We can see that $g_1 < |G \setminus T_{iv}| < \frac{n}{2}$, since her first gain is a proper subset of the complement of T_{iv} in G, while she gains all the nodes in $T_{iv} - v$ by changing her strategy to v'. Hence,

$g_1' = |T_{iv} - v| \geq \frac{n}{2}$. Thus, such a strategy profile does not form a Nash-equilibrium. Therefore, by Lemma 73 (since v is a cut node), we conclude that both u_i's must be in T_{iv}.

Now, we show that in such a case, there is always a Nash-equilibrium. Note that since v is a cut node, then any player who gains v will gain all the nodes in $G \smallsetminus T_{iv}$. Now, we delete $G \smallsetminus T_{iv}$ and connect node v to a path P consisting of $|G \smallsetminus T_{iv}|$ nodes. We call the resulting graph T. We can see that T is a tree, since it is connected and has no cycle. Let u be the neighbour of v on P. Since more than half of the nodes of T are in T_{iv}, then the maximum-weighted branch at u is the one that contains v; that is, we have that

$$wt_T(u) = |T_{iv}| \geq \frac{n}{2} + 1 > \frac{n}{2}.$$

Thus, by Lemma 78, we conclude that the centroid of T, called $C(T)$, must be a subset of T_{iv}. On the other hand, by Theorem 77, we know that tree T always has a Nash-equilibrium which involves $C(T)$. Also, note that the weight of the branch at node v which contains $T_{iv} - v$ is at least $\frac{n}{2}$. Hence, $T_{iv} - v$ must be a maximum branch attached at v. Therefore, we conclude that (u_1, u_2) forms a Nash-equilibrium for the game on T if and only if it is a Nash-equilibrium for the game on G. Hence, in such a case, there exists a Nash-equilibrium for the game on G, that is obtained by finding the Nash-equilibriums of T, according to Theorem 77.

Now, assume that for every $v \in C$ and each $1 \leq i \leq d(v) - 2$, $|T_{iv}| \leq \frac{n}{2}$, and there exists a Nash-equilibrium (u_1, u_2) for this game. Since every node v of weight greater than one in C is a cut node, then by Lemma 73, both u_i's must be either in a T_{iv}, or in $G \smallsetminus T_{iv}$. Hence, in this case, if the equilibrium nodes both are not included simultaneously in any T_{iv}, for a node v on C, and some $1 \leq i \leq d(v) - 2$, then we conclude that both u_i's must be selected on C_W. □

Note that for the game on a unicyclic graph G when the weight of all the nodes in cycle C is at most half of the order of G, Theorem 79 does not guarantee the existence of Nash-equilibrium. It only says where we should be looking for a possible Nash-equilibrium in case that there exist one for this game. The unicyclic graph G in Figure 5.2, with 6-cycle C is an example of a unicyclic graph that does not admit any Nash-equilibrium, as we explain below.

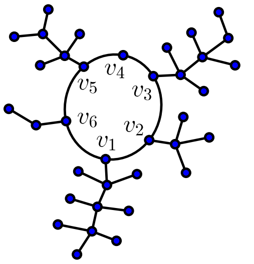


Figure 5.2: The game on the unicyclic graph G does not admit a Nash-equilibrium.

Note that in the graph of Figure 5.2, the weight of each tree attached at a node in cycle C is less than half of the whole number of nodes. By Theorem 79, we know that if there is a Nash-equilibrium (u_1, u_2) for the game on G, then both u_i's must be either in T_v, for some $v \in C$, or in C. By direct checking, we can consider all different possibilities to conclude that there can not be any Nash-equilibrium in which one of the players chooses a strategy out of C. Hence, by Theorem 79, we have to consider the game on C_W.

Now, we have the following bimatrix as the pay-off matrix (see [9], Chapter 3) of the players for the game on C_W; That is, each row of this matrix corresponds to a strategy of the first player, and each column of this matrix corresponds to a strategy of the second player. Note that it is a symmetric game and the columns correspond to nodes v_1, v_2, v_3, v_4, v_5, and v_6, as well as the rows, respectively. Moreover, the (ij)-th entry of this matrix has two components in which the first component shows the pay-off of the first player and the second component shows the pay-off of the second player when they choose the strategy profile (v_i, v_j).

$$\begin{bmatrix} (0,0) & (21^*,15) & (14,10) & (19,17^*) & (16,8) & (25^*,11) \\ (15,21^*) & (0,0) & (19,17) & (16,8) & (25^*,11) & (14,10) \\ (10,14) & (17,19) & (0,0) & (25^*,11) & (14,10) & (15,21^*) \\ (17^*,19) & (8,16) & (11,25^*) & (0,0) & (15,21) & (10,14) \\ (8,16) & (11,25^*) & (10,14) & (21,15) & (0,0) & (17,19) \\ (11,25^*) & (10,14) & (21^*,15) & (14,10) & (19,17) & (0,0) \end{bmatrix}$$

From game theory (see [9], Chapter 3, Definition 3.1.2), we know that a possible Nash-equilibrium for such a game is determined by an entry of this matrix in which the first component is the largest in the same column and the second component is the largest in the row. Here, for each column and each row we indicate such components with a star. As we can see, there is no entry with a star on both components. Thus, there is no Nash-equilibrium for this game.

If in a unicyclic graph the weight of every node on the cycle is at most half of the number of nodes, and there is a Nash-equilibrium for the game on the weighted cycle, then the following lemma could be helpful.

Lemma 80. *Assume that G is a unicyclic graph with weighted cycle C_W such that $wt(v) \leq \frac{n}{2}$ for all $v \in C$. If there is a Nash-equilibrium (u, v) for the two-player game on C_W, then we have that,*

(i) either (u, v) is a Nash-equilibrium for the regular game on G, or

(ii) one of the neighbours of u together with v, or one of the neighbours of v together with u forms a Nash-equilibrium for the game on G.

Proof. Assume that (u, v) is a Nash-equilibrium for the game on C_W. Also, suppose that g_x denotes the pay-off of a player who takes node x as her strategy, where $x \in G$. By assumption, we know that no one can increase her pay-off by changing her strategy to another node on C_W. Hence, we must have that $g_u \geq g_z$, for any $z \in C_W$. Now, we consider the changes in the pay-off of the first player after changing her strategy to a node such as $w \in G - C$ while the strategy of the second player is the fixed node v. By the structure of G, such a node w must be in a tree attached at a node $z \in C$. There are two possibilities for z: either $z \neq v$, or $z = v$.

If $z \neq v$ (note that $w \neq z$), and the first player changes her strategy to node w, then $g_w < g_z$. Namely, this way she gains the nodes on C_W at a later time, since $d(w, v) > d(z, v)$. Therefore, she loses at least one of the nodes that she was able to take by choosing z. Thus, we have that $g_u \geq g_z > g_w$. Hence, the only possible way for the first player to increase her pay-off is to change her strategy to a node w in T_{iv}, for some $1 \leq i \leq d(v) - 2$. We consider such a case as follows.

If $z = v$, that is, $w \in (\bigcup_{i=1}^{d(v)-2} T_{iv}) - v$, and the first player changes her strategy to w, then assume that P is a path of order $|G - \bigcup_{i=1}^{d(v)-2} T_{iv}|$. Let T be the tree obtained by deleting $G \smallsetminus (\bigcup_{i=1}^{d(v)-2} T_{iv})$, and connecting P to $\bigcup_{i=1}^{d(v)-2} T_{iv}$ via v. Since the second player's gain includes node v and all the nodes in $G \smallsetminus (\bigcup_{i=1}^{d(v)-2} T_{iv})$, then finding the best strategy for the first player with respect to v among the nodes in $\bigcup_{i=1}^{d(v)-2} T_{iv}$ for the game on G, is equivalent to finding such a strategy for the game on T. Hence, we consider the best possible strategy for the first player with respect to v among the nodes in $\bigcup_{i=1}^{d(v)-2} T_{iv}$, as follows.

By the proof of Theorem 77, we know that in a game on a tree always the best strategy against an opponent is to choose the neighbour of her strategy that is in a branch with maximum weight. Hence, if for some neighbour of v such as $w \in T_{iv}$, for some $1 \leq i \leq d(v) - 2$, $g_w > g_u$ and g_w is the maximum over such neighbours of v, then the best strategy for the first player (against v) is to change her strategy to w. We claim that in such a case, $\{w, v\}$ forms a Nash-equilibrium for the game on G. Namely, in this case, in one side w is the best strategy against v in game on G (not C_W). On the other side, if the second player changes her strategy to a node $z \neq w$ in T_{iv}, then she will gain $g_z < |T_{iv}| \leq \frac{n}{2} \leq |G - T_{iv}| = g_v$. If she changes her strategy to a node z in T_{jv}, with $j \neq i$, then her gain will be a proper subset of T_{jv} (since she does not gain v any more) while her gain included T_{jv} when her strategy was v. Thus, $g_z < g_v$. If she changes her strategy to a node $z \in G \setminus (\bigcup_{j=1}^{d(v)-2} T_{jv})$, then she does not gain node v, and therefore, she achieves a proper subset of her previous gain (since v is a cut node). Thus, the claim is true; that is, (v, w) forms a Nash-equilibrium in this case.

However, if for every neighbour of v, like w, $g_w \leq g_u$, then we conclude that u is the best strategy against v in G. We can do the same discussion for the second player, and conclude that, either u together with one of its neighbours form a Nash-equilibrium for the game on G, or otherwise, v is the best strategy against u in the game on G. If u and v are the best strategies with respect to each other, then (u, v) forms a Nash-equilibrium for the game on G. □

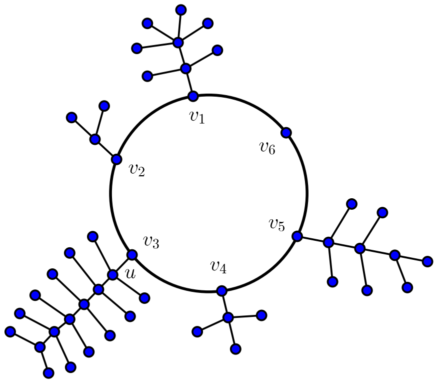


Figure 5.3: (u, v_3) is a Nash-equilibrium for the game on G.

In reverse, if there is a Nash-equilibrium for the game on a unicyclic graph G with cycle C, such that both strategies are in cycle C, then it is also a Nash-equilibrium for the game on C_W, since no player can increase her pay-off. However, if there is a Nash-equilibrium (u_1, v_1) for the game on G in which $u_1, v_1 \in T_{iv}$, for some $v \in C$, then we may not have necessarily a Nash-equilibrium for the game on C_W. For example, Figure 5.3 shows an example of a unicyclic graph with cycle C in which the weight of every node in C_W is less than half of the order of G, and there is a Nash-equilibrium (u, v_3) as shown in the figure. By checking the pay-off matrix of the game on C_W as shown below, we can see that there is no Nash-equilibrium for this game, since there is no entry with stars on both columns (see [9], Chapter 3, Definition 3.1.2).

$$\begin{bmatrix} (0,0) & (17,27) & (8,24) & (11,33^*) & (10,14) & (29,15) \\ (24,8) & (0,0) & (11,33^*) & (10,14) & (29^*,15) & (22,10) \\ (10,14) & (33^*,11) & (0,0) & (29^*,15) & (22,10) & (27,17^*) \\ (33^*,11) & (14,10) & (15,29^*) & (0,0) & (27,17) & (24,8) \\ (14,10) & (15,29^*) & (10,22) & (17,27) & (0,0) & (33^*,11) \\ (15,29) & (10,22) & (17^*,27) & (8,24) & (11,33^*) & (0,0) \end{bmatrix}$$

Assume that G is a unicyclic graph with trees of equal orders attached at the nodes of the cycle C. Then the two-player game on G is like playing on a weighted cycle with equal weights assigned to all nodes. Hence, with the same argument as in proof of Theorem 75, we can see that the set of Nash-equilibriums is determined exactly as for a regular cycle. The only difference is that here the pay-off of the players is a multiple (a constant multiple, which is equal to the weight of the nodes on C) of the pay-off in the regular game on a cycle without weights. Thus, in this case, there is always a Nash-equilibrium for the game on C_W.

From Lemma 80 we conclude that, if we find a Nash-equilibrium for the game on C_W, then we can find a Nash-equilibrium for the game on G. This conclusion shows the importance of the next theorem. We first prove the following lemma that is needed in the proof of Theorem 82.

Lemma 81. *Let $C : v_1, v_2, v_3, v_4, v_5$ be a weighted cycle of order 5 such that the weight of each node $v_i \in C$ is denoted by $wt(v_i)$. Then there are always three nodes say u, v, and w in C such that $d(u,v) = 2 = d(u,w)$, $wt(v) \geq wt(u)$, and $wt(w) \geq wt(u)$.*

Proof. First, note that if the weights of the nodes in C are equal, then $\{v_1, v_3, v_5\}$ forms a triple in C that satisfies the desired property in the statement of the lemma. Hence, we assume that the weight of the nodes in C are not all equal. By contradiction, suppose that there is no such a triple of nodes in C. In particular, for $\{v_1, v_3, v_4\}$, at least one of v_3 and v_4 must have a weight that is at most as large as the weight of v_1. Thus, without loss of generality, by symmetry, we can assume that either $wt(v_3) < wt(v_1)$ and $wt(v_4) \geq wt(v_1)$, or $wt(v_3) < wt(v_1)$ and $wt(v_4) < wt(v_1)$.

If $wt(v_3) < wt(v_1)$ and $wt(v_4) \geq wt(v_1)$, then we must have $wt(v_3) \geq wt(v_5)$. Since otherwise, the triple $\{v_1, v_3, v_5\}$ will be a desired triple, which is a contradiction. Again, by assumption, since there is no desired triple of nodes in C, then at least one of the nodes v_4 and v_5 must have a weight that is at most as large as the weight of v_2. Hence, either $wt(v_5) < wt(v_2)$ and $wt(v_4) \geq wt(v_2)$, or $wt(v_5) < wt(v_2)$ and $wt(v_4) < wt(v_2)$, or $wt(v_5) \geq wt(v_2)$ and $wt(v_4) < wt(v_2)$. If either of the first two cases happens, then we conclude that the triple $\{v_2, v_3, v_5\}$ forms a desired triple, which is a contradiction. If the third case happens, that is, $wt(v_5) \geq wt(v_2)$ and $wt(v_4) < wt(v_2)$, then by the transitivity of the "$\leq$" relation on the set of numbers, we conclude that all the nodes in C must have equal weights, which is a contradiction.

If $wt(v_3) < wt(v_1)$ and $wt(v_4) < wt(v_1)$, then we conclude that $wt(v_3) \geq wt(v_5)$ and $wt(v_4) \geq wt(v_2)$. Otherwise, the triple $\{v_1, v_3, v_5\}$ or $\{v_1, v_2, v_4\}$ forms a desired triple, which is a contradiction. We ave two possibilities: either $wt(v_2) \geq wt(v_5)$, or $wt(v_2) < wt(v_5)$. If the former happens, then the triple $\{v_2, v_3, v_5\}$ forms a desired triple, which is a contradiction. If the latter happens, then the triple $\{v_2, v_4, v_5\}$ forms a desired triple of nodes, which is a contradiction again. Therefore, the lemma is true. □

Now, we can prove the following theorem as the last result of this section.

Theorem 82. *In a two-player game on a unicyclic graph G with cycle C (or weighted cycle C_W) of length $3, 4,$ or 5 there exists a Nash-equilibrium.*

Proof. By Theorem 80, we know that if there is a Nash-equilibrium for the game on C_W, then there must be a Nash-equilibrium for the game on G too. Hence, it suffices to show that in such a case always there is a Nash-equilibrium for the game on C_W. We consider the following possibilities for C_W.

If $|C_W| = 3$, then assume that C consists of three nodes with weights a, b, and c. We can see that the two nodes with the largest weight among a, b, and c always form a Nash-equilibrium, since no one can increase her pay-off unilaterally, and the third node becomes gray.

If $|C_W| = 4$, then assume that $C : v_1, v_2, v_3, v_4, v_1$ consists of four nodes with weights $wt(v_1)$, $wt(v_2)$, $wt(v_3)$, and $wt(v_4)$. We claim that in a possible Nash-equilibrium for this game, the strategies of the players must be adjacent. Namely, if the players choose two non-adjacent nodes in C_W, then the other two nodes in C_W become gray. Therefore, every player by changing her strategy to one of her neighbours can improve her pay-off. Hence, such a case can not be a Nash-equilibrium.

Now, without loss of generality, assume that the players select two adjacent nodes such as v_1 and v_2. Also, suppose that g_1 and g_2 denotes the pay-off of the first player and the second player, respectively. Then (v_1, v_2) form a Nash-equilibrium if and only if,

$$\begin{aligned} g_1 &= wt(v_1) + wt(v_4) \geq wt(v_3) + wt(v_4), \\ g_2 &= wt(v_2) + wt(v_3) \geq wt(v_4) + wt(v_3). \end{aligned}$$

By simplification, this is equivalent to say that (v_1, v_2) form a Nash-equilibrium if and only if,

$$\begin{aligned} wt(v_1) &\geq wt(v_3), \\ wt(v_2) &\geq wt(v_4). \end{aligned}$$

With a simple discussion, by considering different possibilities, we conclude the following statements:

If $wt(v_1) \geq wt(v_3)$ and $wt(v_2) \geq wt(v_4)$, then (v_1, v_2) form a Nash-equilibrium.

If $wt(v_1) < wt(v_3)$ and $wt(v_2) \geq wt(v_4)$, then (v_2, v_3) form a Nash-equilibrium.

If $wt(v_1) \geq wt(v_3)$ and $wt(v_2) < wt(v_4)$, then (v_1, v_4) form a Nash-equilibrium.

If $wt(v_1) < wt(v_3)$ and $wt(v_2) < wt(v_4)$, then (v_3, v_4) form a Nash-equilibrium.

Thus, for the game on a weighted 4-cycle always we have a Nash-equilibrium.

If $|C_W| = 5$, then assume that cycle $C : v_1, v_2, v_3, v_4, v_5, v_1$ consists of five nodes with weights $wt(v_1)$, $wt(v_2)$, $wt(v_3)$, $wt(v_4)$, and $wt(v_5)$. First, note that if the weight of all the nodes in C are equal, then by the same argument as in proof of Theorem 75, we conclude that any two nodes in C from a Nash-equilibrium. Hence, we assume here that the weight of the nodes in C are not all equal.

Suppose that g_1 and g_2 denote the pay-off of the first and the second player in this game, respectively. We have two possibilities for a Nash-equilibrium: either it consists of two adjacent nodes, or two non-adjacent nodes.

Case 1. If the players select two non-adjacent nodes such as v_1 and v_3 as their strategies, then v_2 becomes gray. By definition, (v_1, v_3) form a Nash-equilibrium if and only if

$$g_1 = wt(v_1) + wt(v_5) \geq wt(v_1) + wt(v_2),$$
$$g_1 = wt(v_1) + wt(v_5) \geq wt(v_5) + wt(v_4),$$

for the first player; and

$$g_2 = wt(v_3) + wt(v_4) \geq wt(v_3) + wt(v_2),$$
$$g_2 = wt(v_3) + wt(v_4) \geq wt(v_5) + wt(v_4),$$

for the second player. After simplifying, we say (v_1, v_3) form a Nash-equilibrium if and only if,

$$wt(v_5) \geq wt(v_2), \tag{5.2}$$
$$wt(v_1) \geq wt(v_4), \tag{5.3}$$
$$wt(v_3) \geq wt(v_5), \tag{5.4}$$
$$wt(v_4) \geq wt(v_2). \tag{5.5}$$

Case 2. If the players select two adjacent nodes such as v_4 and v_5 as their strategies, then v_2 becomes gray. By definition, (v_4, v_5) form a Nash-equilibrium if and only if

$$g_1 = wt(v_4) + wt(v_3) \geq wt(v_1) + wt(v_2),$$
$$g_1 = wt(v_4) + wt(v_3) \geq wt(v_3) + wt(v_2),$$

for the first player; and

$$g_2 = wt(v_5) + wt(v_1) \geq wt(v_3) + wt(v_2),$$
$$g_2 = wt(v_5) + wt(v_1) \geq wt(v_1) + wt(v_2),$$

for the second player. After simplification, we say (v_4, v_5) form a Nash-equilibrium if and only if,

$$wt(v_4) \geq wt(v_2), \tag{5.6}$$
$$wt(v_5) \geq wt(v_2), \tag{5.7}$$
$$wt(v_3) + wt(v_4) \geq wt(v_1) + wt(v_2), \tag{5.8}$$
$$wt(v_1) + wt(v_5) \geq wt(v_2) + wt(v_3). \tag{5.9}$$

Note that parts (2) and (5) from Case 1 imply that, if (v_1, v_3) form a Nash-equilibrium, then the triple $\{v_2, v_4, v_5\}$ is a triple that satisfies the conditions in Lemma 81. Also, parts (6), (7) from Case 2 imply that, if (v_4, v_5) form a Nash-equilibrium, then the triple $\{v_2, v_4, v_5\}$ is a triple that satisfies the conditions in Lemma 81. By Lemma 81, we know that such a triple always exists in any weighted cycle of order 5. Hence, without loss of generality, we assume that in C_W, $wt(v_4) \geq wt(v_2)$ and $wt(v_5) \geq wt(v_2)$. Therefore, by parts (3) and (5) from Case 1, and by parts (6), (7) from Case 2, to prove the existence of a Nash-equilibrium for the game on C_W, it suffices to consider the occurrence of conditions (4) and (5) in Case 1, or conditions (8) and (9) in Case 2, as follows.

Note that there are two possibilities for $wt(v_1)$ and $wt(v_3)$: either $wt(v_1) \leq wt(v_3)$ or $wt(v_1) > wt(v_3)$. Assume that, the conditions in Case 1, that is, $wt(v_3) \geq wt(v_5)$ and $wt(v_1) \geq wt(v_4)$ do not hold. Thus, without loss of generality and by symmetry, we can assume that we have two possibilities for conditions in Case : either $wt(v_3) <$

$wt(v_5)$ and $wt(v_1) \geq wt(v_4)$, (or similarly, by symmetry, $wt(v_3) \geq wt(v_5)$ and $wt(v_1) < wt(v_4)$), or $wt(v_3) < wt(v_5)$ and $wt(v_1) < wt(v_4)$. Hence, we need to consider four possibilities as follows.

First, assume that $wt(v_1) \leq wt(v_3)$. Then this together with $wt(v_4) \geq wt(v_2)$ imply that $wt(v_4) + wt(v_3) \geq wt(v_1) + wt(v_2)$, that is, the condition (8) in Case 2 holds. If $wt(v_3) < wt(v_5)$ and $wt(v_1) \geq wt(v_4)$, then by transitivity of the "$\leq$" relation on the numbers, we conclude that $wt(v_1) \geq wt(v_2)$. This implies that $wt(v_1)+wt(v_5) \geq wt(v_2) + wt(v_3)$. Thus, in this case, (v_4, v_5) forms a Nash-equilibrium, as we have all the conditions in Case 2.

If $wt(v_3) < wt(v_5)$ and $wt(v_1) < wt(v_4)$, then we we have to consider inequality (9). Note that in this case we have another triple $\{v_1, v_3, v_4\}$ with the same conditions as in Lemma 81. By replacing $\{v_1, v_3, v_4\}$ in Case 1, regarding to this new triple we conclude that (v_2, v_5) is a Nash-equilibrium if and only if $wt(v_5) \geq wt(v_3)$ and $wt(v_2) \geq wt(v_4)$. Also, by replacing $\{v_1, v_3, v_4\}$ in Case 2, (v_3, v_4) forms a Nash-equilibrium if and only if $wt(v_2) + wt(v_3) \geq wt(v_1) + wt(v_5)$ and $wt(v_4) + wt(v_5) \geq wt(v_1) + wt(v_2)$. But, we already know that $wt(v_4) \geq wt(v_2)$. Moreover, $wt(v_4) + wt(v_5) \geq wt(v_1) + wt(v_2)$, since $wt(v_4) \geq wt(v_1)$ and $wt(v_5) \geq wt(v_2)$. Therefore, we only need to check

$$wt(v_2) + wt(v_3) \geq wt(v_1) + wt(v_5). \tag{5.10}$$

Since always one of (9) or (10) is true, then in this case, we also have a Nash-equilibrium.

Now, assume that $wt(v_1) > wt(v_3)$. Then this together with assumption $wt(v_5) \geq wt(v_2)$, imply that inequality (9) holds. Hence, we have to consider inequality (8). By the above argument, we have to consider two possibilities: either $wt(v_3) < wt(v_5)$ and $wt(v_1) \geq wt(v_4)$, or $wt(v_3) < wt(v_5)$ and $wt(v_1) < wt(v_4)$. Note that in both cases, we assume $wt(v_3) < wt(v_5)$. Hence, in both cases triple $\{v_1, v_3, v_5\}$ satisfies the conditions in Lemma 81. By replacing $\{v_1, v_3, v_5\}$ in Case 1, regarding to this new triple, we can say that (v_2, v_4) forms a Nash-equilibrium if and only if $wt(v_2) \geq wt(v_5)$ and $wt(v_4) \geq wt(v_1)$. Also, by Case 2, (v_1, v_5) forms a Nash-equilibrium if and only if

$$wt(v_4) + wt(v_5) \geq wt(v_2) + wt(v_3), \tag{5.11}$$

and

$$wt(v_1) + wt(v_2) \geq wt(v_3) + wt(v_4). \tag{5.12}$$

Since by assumption, $wt(v_5) \geq wt(v_2)$, then Case 1 for triple $\{v_1, v_3, v_5\}$ may not happen. On the other hand, since by assumption $wt(v_4) \geq wt(v_2)$, and in both possibilities $wt(v_3) < wt(v_5)$, then we conclude that inequality (11) holds. Thus, in both cases, we have to consider inequalities (8) and (12). Since always one of (8) or (12) is true, then in this case, we have a Nash-equilibrium (similarly, for the case $wt(v_3) \geq wt(v_5)$ and $wt(v_1) < wt(v_4)$). That is, if (8) holds, then (v_4, v_5) is a Nash-equilibrium. If (12) holds, then (v_1, v_5) forms a Nash-equilibrium. □

Note that as we discussed earlier, the game on the weighted cycle shown in Figure 5.2 does not admit a Nash-equilibrium, and in this example the cycle is of order 6. Hence, Theorem 82 is the best possible.

Note that for a complete bipartite graph $G = G[X, Y]$, any pair of nodes such as (u, v), where $u \in X$ and $v \in Y$ forms a Nash-equilibrium for the two-player game on G. Namely, by playing (u, v), each player wins all the nodes in the opposite part except for the strategy of her opponent. On the other hand, if the first player changes her strategy to a node in Y, then all the nodes in X become gray at the beginning. Thus, her new pay-off will be zero. If she changes her strategy to a different node in X, then by symmetry, her pay-off will not change. Similarly, by symmetry, the second player can not improve her pay-off by changing her strategy unilaterally in such a case. Therefore, (u, v) forms a Nash-equilibrium. However, finding a Nash-equilibrium for an arbitrary bipartite graph in general seems challenging.

5.2 Cartesian Grids

In this section we investigate the existence of Nash-equilibriums for the game on Cartesian grids. In a grid G, we call a subgraph of G which is also a grid by itself, a *subgrid* of G. Assume that A and B are two nodes of a grid G. We denote the maximal subgrid of G which contains A and B as the corner points and consists of all the shortest paths between A and B in G by G_{AB}.

We have the following fact about the centre of a grid, which follows directly using definition.

Theorem 83. *Assume that G is a $m \times n$ grid with centre C, where m and n are positive integers. Then we have the following statements for C.*

(i) *If m and n are odd, then C consists of a single point in the middle.*

(ii) *If one of m and n is odd and the other one is even, then C consists of two adjacent nodes in the middle.*

(iii) *If m and n are even, then C is the 4-cycle in the middle of G.*

Using the centre of a grid, we can always find a Nash-equilibrium for the two-player competitive diffusion game on grids.

Theorem 84. *Suppose that we have a two-player game on a $m \times n$ grid G with centre C, where m and n are positive integers, and $m \leq n$. Then there is always a Nash-equilibrium (u, v) for the game on G. Moreover, we have the following statements about $\{u, v\}$.*

(i) *If m and n are odd, and $C = \{c\}$, then $\{u, v\} = \{c, x\}$ where x is one of the neighbours of c which is placed in the same row as c.*

(ii) *If one of m and n is odd and the other one is even, and $C = \{c_1, c_2\}$, then $\{u, v\} = \{c_1, c_2\}$.*

(iii) *If m and n are even, then $\{u, v\}$ is a pair of adjacent nodes in C.*

Proof. Assume that A and B are the strategies of the players, with g_1 and g_2 as their pay-offs, respectively.

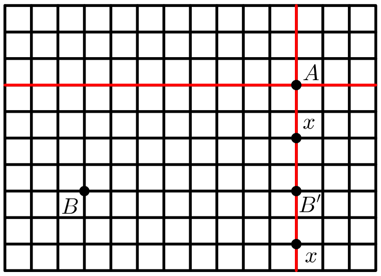


Figure 5.4: The rectangles containing point A on their perimeter.

Then there is a vertical as well as a horizontal line which passes through point A in the grid plane, and forms part of the perimeter of some rectangles created by A (in total, there are at most four possible such rectangles as we see in Figure 5.4, depending on the position of A). We observe that B is always inside of one of those rectangular regions created by A. Now, if G_{AB} is a square, then the distance between A and B is even. Thus, by Lemma 71, there must be some gray nodes appearing on the diagonal points of G_{AB} throughout the game. Hence, each of the players by changing her strategy to one of the neighbours of her previous strategy on the boarder lines of G_{AB} will decrease the distance between A and B by exactly one unit. Hence, the gray nodes vanish and she will gain more. Therefore, this can not be a Nash-equilibrium.

If G_{AB} is not a square, then B is further with respect to one of these rectangles like R_{Ai} than the others. Thus, assuming that B' is the closest point of R_{Ai} with respect to B, for any point like x on the perimeter of R_{Ai}, we have that

$$d(x, A) \leq d(x, B') + d(B', B) = d(x, B).$$

Therefore, through the rounds of the game, the first player (choosing A) gets x before the second player. Thus, the first player wins at least all the nodes in R_{Ai}. Hence, in a possible Nash-equilibrium we have that

$$g_2 \leq mn - |R_{Ai}| \leq mn - \min\{|R_{Aj}| : R_{Aj} \text{ is a rectangle created by A}\}. \tag{5.13}$$

Note that this bound can be achieved only when B is the neighbour of A opposite to the smallest rectangle created by A. Otherwise, the first player wins all the nodes in the smallest rectangle created by A, plus at least the neighbour of A opposite to this rectangle. Thus, the best strategy for the second player is to achieve this bound as discussed. Similarly, we can consider the rectangles created by B, and again we have that

$$g_1 \leq mn - |R_{Bi}| \leq mn - \min\{|R_{Bj}| : R_{Bj} \text{ is a rectangle created by B}\}, \tag{5.14}$$

which can be achieved only when A is the neighbour of B opposite to the smallest rectangle created by B. Hence, in a possible Nash-equilibrium, the strategies of the players should be adjacent.

Now, assume that A and B are two adjacent nodes. Thus, each players wins all the nodes in a rectangle. If the players do not choose their strategies as in (i), (ii), or (iii), then one of the players can gain a larger rectangle by changing her strategy to one of the neighbours of her opponent's strategy. Thus, such a case is not a Nash-equilibrium. On the other hand, suppose that players choose their strategies as in (i), (ii), or (iii). Then no one can increase her pay-off, since no one can enlarge the smallest rectangle created by her strategy. Therefore, (i), (ii), and (iii) form the Nash-equilibriums of this game. □

5.3 The Game with at Least Three Players

In this section, we investigate the competitive diffusion game with three players or more for paths, cycles, grids, and trees.

First, we start considering the three-player game on trees. The following lemma gives us a necessary condition for having a Nash-equilibrium for the three player game on trees.

Lemma 85. *Assume that we have a three-player game on a tree T. If (u, v, w) form a Nash-equilibrium for this game, then u, v, and w must be three consecutive nodes in the same path. Moreover, if $\operatorname{diam}(T) \geq 5$, then we must have $d(v) \geq 3$.*

Proof. First, we claim that in any possible Nash-equilibrium (u, v, w) for this game, u, v, and w must be in the same path. Note that in a tree there is a unique path between any pair of nodes. Without loss of generality, suppose that u and w are of the furthest distance within this triple. By contradiction, suppose that node v is not in the path that connects u to w. Thus, the path between u and v must intersect the path between u and w. Otherwise, the distance between v and w must be larger than the distance between u and w, which is a contradiction. Assume that, the first node that appears in both of these paths is called x. We can see that, each player by changing her strategy to x can increase her pay-off, which is a contradiction. Hence, in a possible Nash-equilibrium the strategies of the players must be in the same path.

Now, if u, v, and w are in the same path, but they are not three consecutive nodes, then every player whose strategy is the furthest with respect to the others' strategy can increase her pay-off at least one unit by moving toward the middle strategy. Thus, in a possible Nash-equilibrium the strategies of the players must be three consecutive nodes.

Finally, assume that $\operatorname{diam}(T) \geq 5$, and there are three consecutive nodes u, v, and w in T that form a Nash-equilibrium for this game, where $d(v) = 2$. Then the gain of v is only one node. Since $\operatorname{diam}(T) \geq 5$ and $d(v) = 2$, then there must be a neighbour of u or w, called x which has more than one neighbour. Thus, the player with strategy v can increase her pay-off by moving to x, which is a contradiction. Therefore, in a possible Nash-equilibrium we have $d(v) \geq 3$. □

We need the following immediate corollary of Lemma 78 for proving the last result of this section.

Corollary 86. *Let u and v be two adjacent nodes in a tree T with centroid C. If the maximum branches attached at u and v have empty intersection, then $\{u, v\} \cap C \neq \varnothing$.*

Proof. By contradiction, suppose that $\{u, v\} \cap C = \varnothing$. Then by Lemma 78, we conclude that C is a subset of the maximum branch attached at u and the maximum branch attached at v, which is a contradiction. □

The following theorem is helpful for finding the possible Nash-equilibriums of the three player game on a tree.

Theorem 87. *Suppose that we have a three-player game on a tree T. Assume that u, v, and w are three consecutive node in T, and u' denotes the node in $T \cdot uv$ which is obtained by contraction of edge uv, and w' represents the node obtained by contraction of edge vw in $T \cdot vw$. If (u, v, w) forms a Nash-equilibrium, then we have the following statement.*

(i) In $T \cdot vw$ one of the nodes w' or u is a centroidal node.

(ii) In $T \cdot uv$ one of the nodes u' or w is a centroidal node.

(iii) $C(T) \cap \{u, v, w\} \neq \varnothing$.

Proof. By definition, we know that a set of three consecutive nodes such as $\{u, v, w\}$ forms a Nash-equilibrium if and only if no one can increase her pay-off unilaterally.

Assume that g_1, g_2, and g_3 are the pay-offs of the players with strategies u, v, and w, respectively. Let P_3 be the path u, v, w. We denote the set of branches attached at u, v, and w which do not have a common edge with path P_3, by $\{A_i\}_{i \in I}$, $\{B_j\}_{j \in J}$, and $\{C_k\}_{k \in K}$. By definition of a Nash-equilibrium, the player with strategy u can not increase her pay-off by changing her strategy to a node x in a A_i, where $i \in I$. Namely, her gain when playing with x is a subset of her gain by playing on u. Thus, the only way that she might be able to increase her pay-off is to change her strategy to a node in B_j or in C_k, where $j \in J$ and $k \in K$. We can have a similar argument for the players with strategies v and w. Hence, we can say that the three consecutive nodes u , v, and w form a Nash-equilibrium if and only if we have the following conditions.

$$g_2 = \sum_j |B_j| \geq |A_i|, \text{for each } i \in I. \tag{5.15}$$

$$g_2 = \sum_j |B_j| \geq |C_k|, \text{for each } k \in K. \tag{5.16}$$

$$g_1 = \sum_i |A_i| \geq |B_j|, \text{for each } j \in J. \tag{5.17}$$

$$g_1 = \sum_i |A_i| \geq |C_k|, \text{for each } k \in K. \tag{5.18}$$

$$g_3 = \sum_k |C_k| \geq |A_i|, \text{for each } i \in I. \tag{5.19}$$

$$g_3 = \sum_k |C_k| \geq |B_j|, \text{for each } j \in J. \tag{5.20}$$

Thus, by (5.15) and (5.19), we conclude that for $i \in I$, A_i can not be a maximum branch attached at u in T or $T \cdot vw$. Therefore,

$$wt(u) = \sum_j |B_j| + \sum_k |C_k| + 2, \quad \text{in } T \cdot vw, \text{ and in } T, \tag{5.21}$$

Also, by (5.17) and (5.18), we conclude that for $k \in K$, or $j \in J$, C_k or B_j can not be a maximum branch attached at w in $T \cdot vw$. Therefore, we have that

$$wt(w') = \sum_i |A_i| + 1, \text{ in } T \cdot vw.$$

Hence, by Corollary 86, we conclude that $\{w', u\} \cap C(T \cdot vw) \neq \varnothing$.

Similarly, by (5.16), (5.19), and (5.20), we conclude that

$$wt(w) = \sum_j |B_j| + \sum_i |A_i| + 2, \quad \text{in } T \cdot uv, \text{ and in } T, \tag{5.22}$$

and

$$wt(u') = \sum_k |C_k| + 1, \quad \text{in } T \cdot uv.$$

Therefore, by Corollary 86, we conclude that $\{u', w\} \cap C(T \cdot uv) \neq \varnothing$.

Finally, by (5.21) and (5.22), we know that the maximum branch attached at u is the one that contains v and w, and the maximum branch at w is the one that contains u and v. Hence, the intersection of the maximum branch at u and the maximum branch at w in T consists of v. Therefore, if u or w are not in $C(T)$, then by corollary 86, we conclude that, $v \in C(T)$. Thus, at least one of the nodes u, v, and w must be in $C(T)$. □

Stars are a well-known family of trees that always admit a Nash-equilibrium. More precisely, for every star $K_{1,n}$, with $n \geq 3$, the node in the centre together with any pair of the leaves make a Nash-equilibrium for a three-player game on $K_{1,n}$. Namely, the central node is the unique centroidal node of $K_{1,n}$. Let c be the centre node of $K_{1,n}$, and u, v be two leaves in $K_{1,n}$. Then we can see that the triple (u, c, v) satisfies all the conditions (5.15), (5.16), (5.17), (5.18), (5.19), and (5.20) in proof of Theorem 87. Hence, it forms a Nash-equilibrium for the three player game on $K_{1,n}$. Moreover, by Lemma 85 and Theorem 87, we conclude that any Nash-equilibrium of $K_{1,n}$ is of this form.

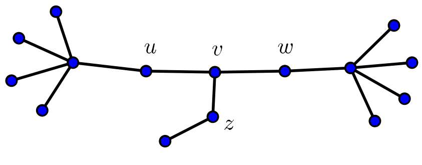


Figure 5.5: A tree of diameter 6 that does not admit a Nash-equilibrium.

Let T be the tree in Figure 5.5. Then we can see that the centroid of T consists of the single node v shown in the figure. Thus, by Lemma 85 and Theorem 87, the only possible Nash-equilibrium for this game is the triple (u, v, w) or (z, v, w). By direct checking, we can see that none of these triples satisfy the conditions in Theorem 87. Therefore, there is no Nash-equilibrium for the three player game on T.

We have the following result for the three-player case of the competitive diffusion game on trees.

Theorem 88. *If T is a tree of diameter at most 4, then there is always a Nash-equilibrium for the three-player game on T.*

Proof. Let $d = diam(T) \leq 4$. We have the following possibilities for d.

First, assume that $d = 1$. Then we can see that the only tree of diameter 1 is the path P_2. If u and v are the nodes of P_2, then we can see that (u, v, v) form a Nash-equilibrium for this game. Namely, in this case, node v becomes gray, and the player with strategy u gains one node. If any of the players with strategy v changes her strategy to u, then u becomes gray, and she will gain nothing again. If the player with strategy u changes her strategy to v, then she gains nothing. Thus, no one can increase her pay-off by changing her strategy.

If $d = 2$, then T must be a star, because, we know that a tree is of diameter two if and only if it is a star. Hence, as we mentioned earlier, there is always a Nash-equilibrium for the three player game on $K_{1,n}$.

Now, assume that $d = 3$. Thus, the length of every longest path in T is three. Suppose that u, v, w are three consecutive nodes in a longest path P in T. Then we claim that these three nodes form a Nash-equilibrium for the three player game on T. Note that since $d = 3$, then exactly one of u or w is a leaf. Without loss of generality, assume that u is a leaf. Again, since $d = 3$, we conclude that $T \smallsetminus \{v, w\}$ is a set of isolated nodes; That is, every node in T is a neighbour of either v or w. Hence, if any player changes her strategy to any other node in $T \smallsetminus \{u, v, w\}$, then she only gains one node. Thus, she can not improve her pay-off. Therefore, (u, v, w) is a Nash-equilibrium.

Finally, suppose that $d = 4$. By assumption, any longest path in T is of length 4 or equivalently, of order 5. By direct checking, we can see that, T has a unique central node called c (that is, node c has the minimum eccentricity). More precisely, c is the middle node of a longest path in T.

Assume that three consecutive nodes u, v, and w form a Nash-equilibrium for the three player game on T. Then we claim that none of u, v, and w is a leaf. By contradiction, suppose that w is a leaf. Hence, the player with strategy w gains only one node that is w. Since $d = 4$, then it implies that she can improve her pay-off by changing her strategy to a neighbour of u that is different from v. Namely, this way she gains at least two nodes, which is a contradiction. Hence, the claim is true. Consequently, in a possible Nash-equilibrium for this game, we have that $v = c$, and u and w must be two neighbours of c.

Note that since $d = 4$, then all the neighbours of u and w, except c, are leaves of T. Suppose that u and w are two neighbours of c in two different branches, called B_1 and B_2, at c such that the weight of B_1 and B_2 is not smaller than the weight of any other branch attached at c. Then we claim that $\{u, v, w\}$ is a Nash-equilibrium for this game. Note that the pay-off of the player with strategy c is at least one. By proof of Theorem 87, we know that the only possible way for her to increase her strategy is moving to a node that is a neighbour of u or w. However, if she changes her strategy to a node $x \in N(u) \cup N(w)$, then she gains only one node, because x is a leaf. Therefore, c is the best strategy for her with respect to u and w.

Note that the gain of the player with strategy u is at least two. By proof of Theorem 87, we also can conclude that, the only possible way for the player with strategy u to increase her pay-off is moving to a node $x \in T \setminus (N[u] \cup \{w\})$. If $x \in N(w)$, then she gains only a leaf. If x is in a branch attached at c, then she gains only a subset of that branch. Hence, by assumption, her gain can not be increased. Thus, u is the best strategy for her. By symmetry, we conclude that, the player with strategy w also can not improve her pay-off. Thus, (u, v, w) as defined above is a Nash-equilibrium. □

Theorem 88 helps us to prove the following result for the paths, as paths are the simplest trees. The following results have been proved independently in paper [20].

Theorem 89. *In a k-player game on a path of length n, where $k \geq 3$, we have the following statements.*

(i) If $k = 3$ and $n \leq 5$, then there is a Nash-equilibrium for the game. If $k = 3$ and $n > 5$, then there is no Nash-equilibrium.

(ii) If $k > 3$, then there is at least one Nash-equilibrium.

Proof. First, note that if $n \leq k$, then by Lemma 72, there is always a Nash-equilibrium for this game. Hence, in the following argument we assume that $n > k$.

If $k = 3$ and $n \leq 5$, then note that any path of order at most 5 is a tree of diameter at most 5. Hence, by Theorem 88, we conclude that there is always a Nash-equilibrium for the game in such a case.

If $k = 3$ and $n \geq 6$, then the diameter of P_n is at least 6. Therefore, by Lemma 85, we conclude that in a possible Nash-equilibrium (u, v, w), $d(v) \geq 3$. However there is no node of degree at least 3 in a path. Thus, there is no Nash-equilibrium for the game in this case.

If $k > 3$, then we find a Nash-equilibrium as follows. Assume that each player chooses a strategy. We call the player who has selected the i-th strategy regarding its order of appearance in the path by the i-th player, where $1 \leq i \leq k$. By Lemma 1, each player wins half of the nodes in any blocked path started from her strategy. Since every path geometrically has two sides and $k \geq 4$, then there must be at least two strategies between the strategy of the first and the k-th player. If there is a blocked path of length more than one between the first and the second strategy (or, similarly between the k-th and $(k-1)$-th strategy), then the corresponding players can increase their pay-off by moving one node toward the middle of the blocked path. Thus, such a strategy is not a Nash-equilibrium. Therefore, in any possible Nash-equilibrium, the strategy of the first and the second players must be adjacent. Similarly, the strategy of the $(k-1)$-th and k-th strategies must be adjacent.

Now, assume that in a strategy profile the strategies of i-th and $(i+1)$-th player are adjacent, for $i \in \{1, k-1\}$. That is, there is no node between the first and the second strategy as well as the $(k-1)$-th and the k-th strategy. Suppose that m_1 is the number of nodes on the left side of the first strategy, and m_{k-1} is the number of the nodes on the right hand side of the k-th strategy. We can see that, $m_1 + 1$ and $m_{k-1} + 1$ are the pay-offs of the first and the k-th player, respectively. For $2 \leq i \leq k-2$, let m_i denote the number of the middle nodes (that is, not including the end points) in the blocked paths induced by the strategies of the $(i-1)$-th player and the i-th player. Thus, the pay-off of the second and the $(k-1)$-th players are equal to $\lfloor \frac{m_2}{2} \rfloor + 1$ and $\lfloor \frac{m_{k-2}}{2} \rfloor + 1$, respectively. Moreover, for $3 \leq i \leq k-3$, the pay-off of the i-th player is equal to $\lfloor m_i/2 \rfloor + \lfloor m_{i+1}/2 \rfloor + 1$.

If $2 \leq i \leq k-1$, and the i-th player changes her strategy to a node that is in placed before the strategy of the $(i+1)$-player or after the strategy of the $(i-1)$-th player, then her new pay-off will be

$$\lfloor x/2 \rfloor \geq \lfloor (m_i + m_{i+1} - x)/2 \rfloor + 1.$$

If she changes her strategy to a node between the strategy of the j-th player and the $(j+1)$-th player, then her new pay-off will be

$$\lfloor \frac{m_j - x - 1}{2} \rfloor + \lfloor \frac{x}{2} \rfloor + 1,$$

where $0 \leq x \leq m_j - 1$. Since $\lfloor \frac{m_j - x - 1}{2} \rfloor + \lfloor \frac{x}{2} \rfloor + 1$ is maximized at $x = 0$, then by definition of a Nash-equilibrium, we conclude that such a strategy profile is a Nash-equilibrium, if and only if the set of the following constraints has a non-empty solution for variables $m_1, m_2, \ldots, m_{k-1}$ in the set of nonnegative integers:

$$m_1 + m_2 + \cdots + m_{k-1} = n + 1 - k,$$

$$m_1 \geq \lfloor (m_i - 1)/2 \rfloor, \text{ for } 2 \leq i \leq k-2, \text{ and } m_1 \geq m_{k-1} - 1,$$

$$m_{k-1} \geq \lfloor (m_i - 1)/2 \rfloor, \text{ for } 2 \leq i \leq k-2, \text{ and } m_{k-1} \geq m_1 - 1,$$

$$\lfloor m_2/2 \rfloor \geq \lfloor (m_i - 1)/2 \rfloor, \text{ for } 3 \leq i \leq k-2,$$

$$\text{and } \lfloor m_2/2 \rfloor \geq m_1 - 1, \text{ and } \lfloor m_2/2 \rfloor \geq m_{k-1} - 1,$$

$$\lfloor m_{k-2}/2 \rfloor \geq \lfloor (m_i - 1)/2 \rfloor, \text{ for } 2 \leq i \leq k-3,$$

$$\text{and } \lfloor m_{k-2}/2 \rfloor \geq m_1 - 1, \text{ and } \lfloor m_{k-2}/2 \rfloor \geq m_{k-1} - 1,$$

$$\text{For } 3 \leq i \leq k-3, \ \lfloor m_i/2 \rfloor + \lfloor m_{i+1}/2 \rfloor \geq \lfloor (m_j - 1)/2 \rfloor,$$

$$\text{where } 2 \leq j \leq k-2, \ j \neq i, i+1, \text{and } \lfloor m_i/2 \rfloor + \lfloor m_{i+1}/2 \rfloor \geq m_1 - 1,$$

$$\text{and } \lfloor m_i/2 \rfloor + \lfloor m_{i+1}/2 \rfloor \geq m_{k-1} - 1.$$

We can see that one solution for this set of constraints is when we take m_i's to be all equal or almost equal (that is, their difference is at most 1), for $2 \leq i \leq k-2$, and $m_1 = m_{k-1} = \max\{m_i : 2 \leq i \leq k-2\}/2$. Moreover, whenever there is no $2 \leq j \leq k-1$ where m_j and m_{j+1} are both odd. Hence, for $k > 3$, there is always a Nash-equilibrium. □

Now, we consider the game with three players or more on cycles.

Theorem 90. *In a k-player game on a cycle C_n, where $k \geq 3$, there is always a Nash-equilibrium.*

Proof. Suppose that we have a k-player game on a cycle C_n with strategy profile $(v_1, v_2, \ldots, v_k)$ where the strategies are indexed according to their order of appearance in C_n. For $1 \leq i \leq k-1$, let n_i denote the number of the nodes in the blocked path induced by v_i and v_{i+1}. Also, let n_k be the number of nodes in the blocked path induced by v_1 and v_k. Then the gain of the player with strategy v_i equals $\lfloor \frac{n_{i-1}}{2} \rfloor + \lfloor \frac{n_i}{2} \rfloor$, for $2 \leq i \leq k$. Moreover, the gain of the player with strategy v_1 is $\lfloor \frac{n_k}{2} \rfloor + \lfloor \frac{n_1}{2} \rfloor$. On the other hand, if the player with strategy v_i changes her strategy to a node v'_i which is between v_j and v_{j+1}, for $1 \leq j \leq k-1$, such that v'_i is of distance x from v_j, then her new pay-off will be $\lfloor \frac{n_j - x + 1}{2} \rfloor + \lfloor \frac{x}{2} \rfloor$. If v'_i is between v_k and v_1 such that v'_i is of distance x from v_k, then her new pay-off will be $\lfloor \frac{n_k - x + 1}{2} \rfloor + \lfloor \frac{x}{2} \rfloor$. Thus, by definition, $(v_1, v_2, \ldots, v_k)$ forms a Nash-equilibrium if and only if the following constraint holds:

$$\lfloor \frac{n_i}{2} \rfloor + \lfloor \frac{n_{i+1}}{2} \rfloor \geq \lfloor \frac{n_j - x + 1}{2} \rfloor + \lfloor \frac{x}{2} \rfloor, \text{ for any } 1 \leq i \leq k-1, \text{ and for } 1 \leq j \leq k,$$
$$\text{where } 0 \leq x \leq n_j.$$
$$\lfloor \frac{n_{k-1}}{2} \rfloor + \lfloor \frac{n_k}{2} \rfloor \geq \lfloor \frac{n_j - x + 1}{2} \rfloor + \lfloor \frac{x}{2} \rfloor, \text{ for any } 1 \leq j \leq k, \text{ where } 0 \leq x \leq n_j.$$

Since $\lfloor \frac{n_j - x + 1}{2} \rfloor + \lfloor \frac{x}{2} \rfloor$ is maximized at $x = 0$, then we conclude that $(v_1, v_2, \ldots, v_k)$ forms a Nash-equilibrium if and only if the following constraint holds:

$$\lfloor \frac{n_i}{2} \rfloor + \lfloor \frac{n_{i+1}}{2} \rfloor \geq \lfloor \frac{n_j + 1}{2} \rfloor, \text{for any } 1 \leq i \leq k-1, \text{ and for } 1 \leq j \leq k.$$
$$\lfloor \frac{n_{k-1}}{2} \rfloor + \lfloor \frac{n_k}{2} \rfloor \geq \lfloor \frac{n_j + 1}{2} \rfloor, \text{for any } 1 \leq j \leq k.$$

We can see that one solution for this set of constraints is when we take m_i's in a way that their difference is at most 2. Moreover, there should be no $2 \leq j \leq k-1$ where n_j and n_{j+1} are both odd. Also, n_1 and n_k can not be both oddd. Hence, for $k > 3$, there is always a Nash-equilibrium. □

In [20], Bulteau *et al.* prove a conjecture about the three player game on the Cartesian grids that we suggested in [51]. Below is the statement of the theorem.

Theorem 91 ([20])**.** *If $n \geq 5$ and $m \geq 5$, then there is no Nash-equilibrium for the* *three player game on the Cartesian grid* $P_n \,\square\, P_m$.

As discussed in [20], the existence of Nash-equilibriums for the k-player game on grids, where $k > 3$ is left as a open problem.

Chapter 6

Conclusion

In this thesis, we introduced the graph burning process and its corresponding graph parameter the burning number. We defined graph burning as a model for the spread of influence in social networks mainly motivated by two other well-known models: the competitive diffusion game [2] and Firefighter [27]. It turned out that graph burning is related to several other graph theory problems such as graph bootstrap percolation [5], push and pull protocol [22], and graph domination [34], and also, other variations of domination in graphs like the distance domination, broadcasting, facility location, and (k, r)-centre problem [8, 25, 32, 34].

By definition, graph burning is quite different from the Firefighter problem, the competitive diffusion game, and graph bootstrap percolation and broadcasting, as its underlying model is new. Despite the fact that the burning problem is tightly related to distance domination problem as we saw in Theorem 30 in Chapter 2, it has a different nature compared to domination problems. Namely, as we discussed in Chapter 3, it is **NP**-complete even for trees with a simple structure as spider graphs and trees with maximum degree three, and even for path-forests. However, it is known that domination problems are solvable in polynomial time for trees [34].

As we considered the burning problem from different perspectives in this thesis, it seems that graph burning has its unique personality among all these problems. The most significant conjecture that we have on the burning number is Conjecture 28. That is, for any graph G of order n with t components, where n and t are positive integers, we conjecture that

$$b(G) \leq \lceil \sqrt{n} \rceil + t - 1.$$

We have seen the truth of this conjecture for paths, path-forests, perfect binary trees, and some specific spider graphs in Chapter 2 and Chapter 3. However, the problem remains open in general.

Conjecture 24 from Chapter 2 suggests that: If G and $\overline{G}$ are both connected graphs of order n, then we have that $b(G)b(\overline{G}) \leq n+4$. We can see that the bound in this conjecture is tight for C_5. However, it remains open in general for connected graphs.

Another interesting problem from Chapter 2 is presented in Conjecture 41 for the exact value of the burning number of the hypercube graphs. That is, for the n-dimensional hypercube graph Q_n, we conjecture that

$$b(Q_n) = \left\lceil \frac{n}{2} \right\rceil + 1.$$

Moreover, if n is even, then in any optimum burning sequence for Q_n, the first and the second sources of fire must be two nodes of distance n.

In an effort to prove Conjecture 28, we came out to conclude Conjecture 51 in Chapter 3, that suggests a relation between a structural property of a graph and its burning number. Here is the statement of this conjecture:

Assume that $\{Q_i\}_{i=1}^{t}$, where $t \geq 3$, forms a decomposed spider in a tree T, and let v_i bes the non-terminal end point of Q_i, for $1 \leq i \leq t$. If $b(\cup_{i=1}^{t} Q_i) \geq k$, and $d(v_i, v_j) \geq 2k$ for all $1 \leq i, j \leq t$, then $b(T) \geq k+1$.

We know the truth of this conjecture for $t \geq k$ in general, and for some specific spider graphs, as discussed in Chapter 3. Also, we know that it is true for paths by Theorem 19 from Chapter 2. However, we have not proved it is true for many other special families of trees. Note that if the conjecture was true, it can not be useful for finding a lower bound on the burning number of all trees. For example, by Theorem 3.2, we know that for a perfect binary tree T with radius r, $b(T) = r+1$. By direct checking, we can see that there is no decomposed spider in T with r terminal paths that satisfies the condition in the conjecture. Hence, we guess that it might be helpful in finding a lower bound on $b(T)$ for a tree T, where $b(T) < \text{rad}(T) + 1$.

In Chapter 4, we introduced and considered the new non-deterministic Poisson graph burning process. We obtained a general upper bound on the burning time, and an asymptotic order for the burning time of paths. We plan to consider this new parameter for other families of graphs.

In Chapter 5, we considered the competitive diffusion game only in a some families of graphs. Considering the existence of Nash-equilibriums for this game remains open for many other families of graphs such as outer planar graphs, bipartite graphs, and the Cartesian grids for more than three players.

This thesis was just the beginning for the study of the graph burning and still we are left with many open problems and unexplored directions that move toward a deeper understanding of this new graph parameter. Here are some future directions:

Determining the burning number remains open for many classes of graphs, including trees and disconnected graphs. It remains open to consider the burning number in real-world social networks such as Facebook or LinkedIn. As Theorem 36 suggests, the burning number of on-line social networks is likely of constant order as the network grows over time. We remark that burning number generalizes naturally to directed graphs; one interesting direction is to determine the burning number on Kleinberg's small world model [42], which adds random directed edges to the Cartesian grid.

By Lemma 2 from Chapter 2, we know that

$$b(G) = \min_{i \geq 1} b_i(G).$$

A different direction that remains open to explore is considering the $b_i(G)$'s of a graph G.

A simple variation which leads to complex dynamics is to change the rules for nodes to burn. As in graph bootstrap percolation [5], the rules could be varied so nodes burn only if they are adjacent to at least r burned neighbors, where $r > 1$. We plan on studying this variation in future work.

Finally, we can also think of the idea of combining the burning process and the competitive diffusion game on graphs. That is, we can think of having a N-player game on the node set of a graph G with the following rules:

Each player i, where $1 \leq i \leq N$, has a distinct colour i. Initially all the nodes are uncoloured. In each round of the game, each player chooses an uncoloured node and colours the node with its own colour, if such a node is available. If two or more players choose the same node, then that node becomes gray and goes out of the game. At the same time, every uncoloured node that has coloured neighbours, receives the colours from them. If such a node receives more than one colour, then it becomes gray and goes out the game. The game ends when nobody can colour a new uncoloured node. The pay-off of each player i, where $1 \leq i \leq N$, is the number of the nodes in G with colour i. One could study the existence of Nash-equilibrium, or finding the fastest time of ending this game.

Bibliography

[1] J. Adler and U. Lev, Bootstrap Percolation: visualizations and applications, *Braz. J. Phys.* **33** (2003), 641–644.

[2] N. Alon, M. Feldman, A.D. Procaccia, M. Tennenholtz, A note on competitive diffusion through social networks, *Information Processing Letters* **110** (2010), 221–225.

[3] N. Alon, P. Prałat, N. Wormald, Cleaning regular graphs with brushes, *SIAM Journal on Discrete Mathematics* **23** (2008), 233–250.

[4] M. Aouchiche, P. Hansen, A survey of Nordhaus-Gaddum type relations, *Discrete Applied Mathematics* **161** (2013), 466–546.

[5] J. Balogh, B. Bollobás, R. Morris, Graph bootstrap percolation, *Random Structures and Algorithms* **41** (2012), 413–440.

[6] S. Banerjee, A. Das, A. Gopalan, S. Shakkottai, Epidemic spreading with external agents, *IEEE Transactions on Information Theory*, July 2014.

[7] A. Barghi, P. Winkler, Firefighting on a random geometric graph, *Random Structures and Algorithms* **46** (2015), 466–477.

[8] J. Barilan, G. Kortsarz, and D. Peleg, How to allocate network centers, *J. Algorithms*, **15** (1993), 385–415.

[9] E.N. Barron, *Game Theory, An Introduction, 2nd Edition*, John Wiley & Sons, Inc., Hoboken, NJ, 2013.

[10] T. Basar, S.R. Etesami, Complexity of Equilibrium in Diffusion Games on Social Networks, In:*Proceedings of the American Control Conference (ACC '14), pages 2065–2070. Institute of Electrical and Electronics Engineers Inc.*, 2014.

[11] C. Berge, *Theory of Graphs and its Applications*, Translated by Alison Doig, Methuen & Co. Ltd., London; John Wiley & Sons Inc., New York, 1962.

[12] S. Bessy, D. Rautenbach, Bounds, Approximation, and Hardness for the Burning Number, arXiv:1511.06023.

[13] S. Bhagat, A. Goyal, L.V.S. Lakshmanan, Maximizing Product Adoption in Social Networks, In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, (2012), 603–612.

[14] B. Bollobás, O. Riordan, *Percolation*, Cambridge University Press, New York, 2006.

[15] A. Bonato, N. Hadi, P. Horn, P. Prałat, C. Wang, Models of on-line social networks, *Internet Mathematics* **6** (2011), 285–313.

[16] A. Bonato, J. Janssen, E. Roshanbin, Burning a Graph as a Model of Social Contagion, Lecture Notes in Computer Science 8882, (2014), 13–22.

[17] A. Bonato, J. Janssen, E. Roshanbin, How to burn a graph, accepted to Internet Mathematics 2015, arXiv:1507.06524.

[18] A. Bonato, J. Janssen, E. Roshanbin, Burning a Graph is Hard, arXiv:1511.06774.

[19] A. Bonato, R.J. Nowakowski, *The Game of Cops and Robbers on Graphs*, American Mathematical Society, Providence, Rhode Island, 2011.

[20] L. Bulteau, V. Froese, N. Talmon, Multi-Player Diffusion Games on Graph Classes, *Lecture Notes in Comput. Sci.* **9076** (2015), 200–211.

[21] J. Chalupa, P.L. Leath and G.R. Reich, Bootstrap percolation on a Bethe lattice, *J. Phys. C.* **12** (1979), L31–L35.

[22] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, Epidemic algorithms for replicated database maintenance, In:*Proceedings of 6th Symp. Principles of Distributed Computing (PODC)*, 1–12, 1987.

[23] J. Díaz, D. Mitsche, G. Perarnau, X. Pérez-Giménez, On the relation between graph distance and Euclidean distance of random geometric graphs, Preprint, arxiv:1404.4757.

[24] C. Dürr, N.K. Thang, Nash equilibria in Voronoi games on graphs, In: Proceedings of the 15th Annual European Symposium on Algorithms, 17–28, 2007.

[25] D.J. Erwin, Dominating broadcasts in graphs, *Bull. Inst. Comb. Appl.*, **42** (2004), 89–105.

[26] S. Finbow, A. King, G. MacGillivray, R. Rizzi, The firefighter problem for graphs of maximum degree three, *Discrete Mathematics* **307** (2007), 2094–2105.

[27] S. Finbow, G. MacGillivray, The Firefighter problem: a survey of results, directions and questions, *Australasian Journal of Combinatorics* **43** (2009), 57–77.

[28] J. Flum, M. Grohe *Parameterized Complexity Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.

[29] X. Gao, Y. Wang, X. Li, W. Wu, Analysis on theoretical bounds for approximating dominating set problems, *Discret. Math. Algorithms Appl.* **1(1)** (2009), 71–84.

[30] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of **NP**-Completeness*, W.H. Freeman, 1979.

[31] A. Goel, S. Rai and B. Krishnamachari, Sharp thresholds for monotone properties in random geometric graphs, *Annals of Applied Probability*, **15** 364–370, 2005.

[32] T.F. Gonzalez, Clustering to minimize the maximum intercluster distance, *Theoretical Computer Science, Elsevier Science B.V.* **38** (1985), 293–306.

[33] B.L. Hartnell, Firefighter! An application of domination, Presentation, 25th Manitoba Conference on Combinatorial Mathematics and Computing, University of Manitoba in Winnipeg, Canada, 1995.

[34] T.W. Haynes, S.T. Hedetniemi, P.J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, New York, 1998.

[35] M.A. Henning, Distance domination in graphs, In T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, editors, *Domination in Graphs: Advanced Topics*, pages 321–349. Marcel Dekker, New York, 1998.

[36] H. Hulett, T.G. Will, G.J. Woeginger, Multigraph realizations of degree sequences: Maximization is easy, minimization is hard, *Operations Research Letters* **36** (2008), 594–596.

[37] W. Imrich, S. Klavžar, *Product Graphs-Structure and Recognition*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley-Interscience, New York, 2000.

[38] S. Janson, T. Łuczak, A. Ruciński. *Random graphs*, Wiley, New York, 2000.

[39] J. Janssen, C. Vautour, Finding safe strategies for competitive diffusion on trees, *Internet Mathematics* **11** (2014), 232–252.

[40] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, Randomized Rumor Spreading, In:*Proceedings of 41st Symp. Foundations of Computer Science (FOCS)*, 565–574, 2000.

[41] D. Kempe, J. Kleinberg, É. Tardos, Influential nodes in a diffusion model for social networks, In: *Proceedings 32nd International Colloquium on Automata, Languages and Programming(ICALP)*, 2005.

[42] J. Kleinberg, The small-world phenomenon: an algorithmic perspective, In: *Proc. 32nd ACM Symp. Theory of Computing*, 2000.

[43] D. Li, L. Liu, H. Yang, Minimum connected r-hop k-dominating set in wireless networks, *Discret. Math. Algorithms Appl.* **1(1)** (2009), 45–57.

[44] M. Mavronicolas, B. Monien, V.G. Papadopoulou, F. Schoppmann, Voronoi games on cycle graphs, In: Proceedings of the 33rd International Symposium on Mathematical Foundations of Computer Science, 503–514, 2008.

[45] A. Meir, J.W. Moon, Relations between packing and covering numbers of a tree, *Pacific Journal of Mathematics* **61** (1975), 225–233.

[46] D. Mitsche, P. Prałat, E. Roshanbin, Burning graphs - a probabilistic perspective, arXiv:1505.03052.

[47] E.A. Nordhaus, J. Gaddum, On complementary graphs, *Amer. Math. Monthly* **63** (1956), 175–177.

[48] O. Ore, *Theory of Graphs*, Number 38 in American Mathematical Society Publications. AMS, Providence, 1962.

[49] M. Penrose, *Random Geometric Graphs*, Oxford Studies in Probability. Oxford U.P., 2003.

[50] M. Penrose, The longest edge of the random minimal spanning tree, *Annals of Applied Probability*, **7**(2), 340–361, 1997.

[51] E. Roshanbin, The Competitive Diffusion Game in Classes of Graphs, Algorithmic Aspects in Information and Management Volume 8546 of the series Lecture Notes in Computer Science, (2014), 275–287.

[52] S.M. Ross, *Stochastic Processes, 2nd edition*, Wiley Series in Probability and Statistics: Probability and Statistics, 1996.

[53] S.M. Ross, *Introduction to Probability Models, 11th edition*, Elsevier/Academic Press, Amsterdam, 2014.

[54] L. Small, O. Mason, Information diffusion on the iterated local transitivity model of online social networks, *Discrete Applied Mathematics* **161** (2013), 1338–1344.

[55] L. Small, O. Mason, Nash Equilibria for Competitive Information Diffusion on Trees, *Information Processing Letters* **113** (2013), 217–219.

[56] P.J. Slater, R-domination in graphs, *J. Assoc. Comput. Mach.* **23** (1976), 446–450.

[57] R. Takehara, M. Hachimori, M. Shigeno, A comment on pure-strategy Nash equilibria in competitive diffusion games, *Information Processing Letters* **112** (2012), 59–60.

[58] D.B. West, *Introduction to Graph Theory, 2nd edition*, Prentice Hall, 2001.

[59] H.S. Wilf, The uniform selection of free trees, *Journal of Algorithms* **2** (1981), 204–207.