VORONOI GAME ON GRAPHS

by

Sayan Bandyapadhyay

under the supervision of

Prof. Sandip Das

submitted in fulfillment of the requirements
of the degree of M.Tech in Computer Science

to the

Indian Statistical Institute, Kolkata

July 2013

Dedicated to

My Parents

Certificate

This is to certify that the thesis entitled **"Voronoi Game On Graphs"** submitted by **"Mr. Sayan Bandyapadhyay"** to the Indian Statistical Institute Kolkata, for the award of the Degree of M.Tech in Computer Science, is a record of the original bona fide research work carried out by him under my supervision and guidance. The thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

Kolkata

July 2013

Prof. Sandip Das

Thesis supervisor

ACMU

ISI Kolkata

Acknowledgements

Lifelong learning is a challenge. As a society, we value knowledge and recognize intellectual development as a key to sustainability. This thesis embodies the quest for knowledge made possible by the support of an academic community of dedicated educators. Reaching major milestones in one's life does not happen without the assistance of great many people, and for each one of them I am truly blessed and grateful.

I would like to express my sincere gratitude to my supervisor, Prof. Sandip Das, for his expert supervision and research insight. Above all and the most needed, he provided me unflinching encouragement and support in various ways. I am privileged to have the opportunity to work under him. With his valuable suggestions, understanding, patience, enthusiasm, vivid discussions and above all with his friendly nature, he helped me in successfully completing my M.Tech thesis work.

I am thankful to ISI Kolkata authorities for providing me the necessary facilities for smooth completion of my research work. Special thanks to my senior colleague Mr. Aritra Banik and my classmate Mr. Hirak Sarkar for generously sharing their time and knowledge. I'm grateful to all faculty members of ACMU, ISI Kolkata, for their co-operation and support. I also want to acknowledge the contribution of Mandar sir, Smarajit sir and Murthy sir for teaching us with so much patience.

Words cannot completely express my love and gratitude to my family who have supported and encouraged me through this journey. I would like to thank my parents for their life-long support, and sacrifices, which sustained my interest in research and motivated me towards the successful completion of this study. My heartiest thanks to my sisters and brothers-in-law who have supported me through *my difficult times with their inspiration and continued encouragement. I also acknowledge all the members of my extended family for their love and support.*

I will always remember the true friends who have walked with me on my path. There are too many to mention by name, yet, too few to ever lose room for in my heart. Many thanks to my dear friends, especially of Hirak, Ritankar, Ravi, Sunny, Bipul, Bipin, Sanchita and Abhishek for times of laughter when I needed a reprieve from the intensity of research. I also enjoyed the company of my senior colleagues; Aritra, Ayan, Soumen, Prabhu, Dhruba and Subhabrata.

Finally, I thank the almighty God for the passion, strength, perseverance, and the resources to complete this study.

Kolkata *Sayan Bandyapadhyay*

July 2013

Abstract

Voronoi game is a geometric model of competitive facility location problem, where each market player comes up with a set of possible locations for placing their facilities. The objective of each player is to maximize the region occupied on the underlying space. In this thesis we consider one round *Voronoi game* with two players. Here the underlying space is a road network, which is modeled by a graph embedded in $\mathbb{R}^2$. In this game each of the players places a set of facilities and the underlying graph is subdivided according to the *nearest neighbour rule*. The player which dominates the maximum region of the graph wins the game. This thesis mainly deals with the problem of determining optimal strategies of the players. We characterize the optimal facility locations of second player given a placement by first player. Using this result we design a polytime algorithm for determining the optimal strategy of second player on trees. We also show that the same problem is $\mathcal{NP}$-hard when the underlying space is a general graph. Moreover we present a 1.58 factor approximation algorithm for the above mentioned problem. Then we concentrate on optimal strategy of first player. We give a lower bound on the optimal payoff of first player. We discuss optimal strategy of first player for $(1,1)$ and $(2,1)$ game on tree network. Then we characterize optimal facility locations of first player for $(1,1)$ game on graph network.

Contents

1

Introduction

1.1 Problem Definition

A situation often arises in market where the competitive service providers (Hotel Chains, Supermarkets etc.) want to occupy a big area in a locality for placing their facilities (Hotel, Shopping Mall etc.) so that they could attract as much customers as possible. The game-theoretic analogue of competitive facility location problem is *Voronoi Game* which was proposed by Ahn et al. [1]. In this game the main objective of a player is to cover maximum area by placing its facilities on the underlying space. A point on the underlying space is always served by its nearest facility. Different versions of this game can be modeled by changing the underlying space like line segment, circular arc, graph and 2D-plane.

In this thesis we consider a game where the underlying space is a road network, represented by a graph $G(V, E)$. With each edge $(u, v) \in E$, a positive weight $w(u, v)$ is associated which can be considered as the length of the edge (u, v). Let us assume that an embedding of G on $\mathbb{R}^2$ is given. As any edge (u, v) is having a positive weight $w(u, v)$, it can be mapped to a closed interval $[0, w(u, v)]$ of length $w(u, v)$. For any point p on this interval consider the distance between p and u as $|p|$ and between p and v as $w(u, v) - |p|$. Throughout the thesis, by *a point* p *on* G we mean either $p \in V$ or p belongs to any edge (u, v). For any two points p and q in G, the *distance* between p and q is considered as the weighted shortest path distance between them and is denoted by $d(p, q)$. A weight w_v is associated with each vertex v. For any graph G', let $W(G')$ be sum of weights of vertices and edges of G'. Then weight of G,

$$W(G) = \sum_{(u,v) \in E} w(u, v) + \sum_{v \in V} w_v$$

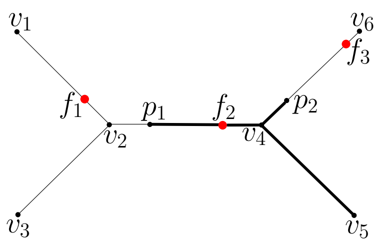

FIGURE 1.1: Service zone of f_2

Refer a portion of an edge as *sub-edge*. Recall that an edge e is modelled as an interval of length $w(e)$. Hence a sub-edge can be represented as a sub-interval of the interval with length defined accordingly. Define a *sub-graph* $G' = (V', E')$ of a Graph G, such that E' is a finite subset of edges and sub-edges of G. Hence V' can contain some vertices of V or some points belong to edges of G. Weight of a sub-edge is same as the length of the sub-interval correspond to it. For any sub-graph G' of G define the weight of G', $W(G')$ as the sum of the weights of the edges, sub-edges and vertices present in G'.

Like any other versions of voronoi game, here facilities are modeled as points in the underlying space. Given any set of facilities F on G, service zone $G_F(f)$ of any facility $f \in F$ is defined as the set of points in G that are closer to f than any other facility $f' \in F$. In case if a point is equidistant to its nearest facilities of P1 and P2, then it is included in the service zone of P2. Observe that for any facility f, $G_F(f)$ is a connected sub-graph of G. It would be more appropriate to refer $G_F(f)$ as a subset of G, because $G_F(f)$ may contain portions of some edges. For example in Figure 1.1, service zone of f_2 (shown in bold) contains the portions of the edges (v_2, v_4) and (v_4, v_6) where p_1 be the point such that $d(f_1, p_1) = d(p_1, f_2)$ and for p_2, $d(f_2, p_2) = d(p_2, f_3)$. For a set of facilities $F' \subseteq F$ define the *service zone* of F', $G_F(F') = \cup_{f \in F'} G_F(f)$. With all these definitions in our hand, next we define the model that we consider throughout this thesis.

1.2 The Model

In this thesis we will consider the *One-Round* (m, k) *Voronoi Game on Graphs*. The game consists of a weighted graph $G(V, E)$ and two players P1 and P2 respectively. Initially P1 places m facilities, followed by which P2 places k facilities in G. For any set of facilities F and S by P1 and P2 respectively, the payoff of P1, $\mathcal{Q}_1(F, S)$ is defined as $W(G_{F \cup S}(F))$ and the payoff of P2, $\mathcal{Q}_2(F, S)$ is defined as $W(G) - \mathcal{Q}_1(F, S)$. Let $\nu(F)$=$\max_S \mathcal{Q}_2(F, S)$, where maximum is taken over any placement of k facilities S by P2. The *One-Round* (m, k) *Voronoi Game on Graphs* can be formally stated as follows.

One-Round (m, k) *Voronoi Game on Graphs:* Given a graph $G = (V, E)$ and two players P1 and P2 having m and k facilities respectively, P1 chooses a set F^* of m facility locations following which P2 chooses a set S^* of k facility locations disjoint from F^* in G, such that:

(i) $\max_S \mathcal{Q}_2(F^*, S)$ is attained at $S{=}S^*$

(ii) $\min_F \nu(F)$ is attained at $F = F^*$, where the minimum is taken over all possible set of facility locations F of P1

1.3 Previous Works

There are quite a lot of works in literature on Voronoi Game. Ahn et al. [1] consider the case where the game is restricted to 1-dimensional continuous domain. Cheong et al. [6] and Fekete et al. [7] deal with 2-dimensional case but for one round. Banik et al. [3] [4] discuss the discrete versions of this game on lines. Demaine et al. [13] consider the discrete version of the game on graphs where the users and facilities are constrained to be located on vertices. A special case of this game when the underlying space is a path is considered by Kiyomi et al. [11]. Recently Banik et al. [5] have studied the discrete version of this game on polygon.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows. In Chapter 2 optimal strategy of P2 is discussed. There a finite set of points is characterized which always contain an optimal placement of P2. Next this result is used to present a polynomial time algorithm for finding optimal strategy of P2 on trees. We also discuss about the complexity of finding optimal strategy of P2 on general graphs. In fact we show that the above mentioned problem is $\mathcal{NP}$-hard. Then we propose a 1.58 factor approximation algorithm for the same problem.

In Chapter 3 we talk about the optimal strategy of P1. There we give a lower bound on optimal payoff of P1 where the underlying space is tree. Next we discuss strategy of P1 for $(1, 1)$ and $(2, 1)$ game on tree. We end the chapter after discussing about optimal strategy of P1 for $(1, 1)$ game on general graphs.

Chapter 4 summarizes the work done in this thesis and presents a number of open problems as future research.

2

Optimal Strategy of Second Player

2.1 Introduction

This chapter is devoted for discussions on optimal strategy of P2 for *One-Round* (m, k) *Voronoi Game on Graphs*. Here we assume that P1 has already placed it's m facilities and now P2 is interested to place it's k facilities so that it's payoff could be optimized.

2.2 Characterization of Optimal Facility Locations

In this section we show that it suffices to search a finite number of points for determining optimal strategy of P2. Let $G(V, E)$ be any graph and $F=\{f_1, f_2, \dots f_m\}$ be a set of facilities placed by P1. First we consider the problem where P2 places only one facility on G. Goal is to find an optimal placement of P2. Let $V=\{v_1, v_2, \dots, v_n\}$ and for any two vertices v_i and v_j denote the edge joining them by e_{ij}. Define an *arc* to be an edge or a sub-edge. An arc between two points u and v are denoted by $\langle u, v\rangle$. For any vertex $v_i \in V$ denote the facility closest from v_i among the facilities in F by $f(v_i)$ and the distance between v_i and $f(v_i)$ by d_i. Let $\Gamma(v_i)$ be the set of points in G which are at a distance d_i from v_i. Now observe that any edge can contain at most two points from $\Gamma(v_i)$. Hence for any vertex v_i, $|\Gamma(v_i)|$ contains $O(|E|)$ many points. Let $\Gamma = \cup_{1\le i\le n}\Gamma(v_i)$. Thus Γ contains $O(|V||E|)$ many points.

Let f_k be any facility of P1 on any edge e_{ij}. Then we assume that, there exists two points $p_1 \in \langle v_i, f_k\rangle$ and $p_2 \in \langle f_k, v_j\rangle$ very close to f_k such that distance between p_1 and f_k and p_2 and f_k are small enough to be considered as zero. For all f_k, $k = 1, \dots, m$ include all those points into Γ and we have the following observation.

Observation 2.2.1. *[2] Number of points in Γ is bounded by $O(|V||E| + m)$.*

Let s be any placement of a facility by P2 located on an arbitrary edge e_{ij}. Consider any path λ between s and any facility $f_l \in F$, such that half of the points of λ are closer to f_l than any other facility in $F \cup \{s\}$ and the rest of the points are closer to s than any other facility in F. Denote all such paths by $\pi(s)$. For example in Figure 2.1 the path between s and f_2 is in $\pi(s)$, but the path between s and f_1 is not in $\pi(s)$. Observe that for any path $\lambda \in \pi(s)$, λ contains at least one of v_i or v_j. Let $\pi_i(s)$ be the set containing all those paths of $\pi(s)$ which contain v_i, but not v_j and $\pi_j(s)$ be the set containing all those paths of $\pi(s)$ which contain v_j, but not v_i. For the paths $\lambda \in \pi(s)$, such that λ contain both of v_i and v_j, include λ in $\pi_i(s)$ if v_i is preceded by v_j in λ, otherwise include it in $\pi_j(s)$. Note that a path is a sequence of vertices and hence the predecessor relationship holds between two vertices. Observe that $\pi_i(s) \cap \pi_j(s) = \emptyset$. Define the set $B_i(s)$ and $B_j(s)$ such that they contain the midpoints of the paths in $\pi_i(s)$ and $\pi_j(s)$ respectively. We refer to those midpoints as *bisectors*.

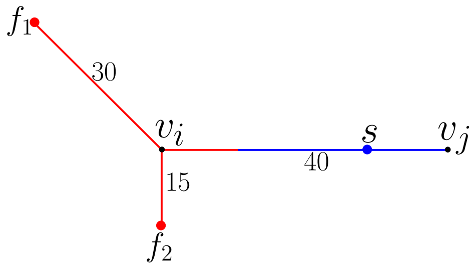

Figure 2.1: Example of facilities placed by P1 and P2

Observation 2.2.2. *[2] Each edge contains at most one point of $B_i(s) \cup B_j(s)$.*

Proof. Suppose there exists two paths λ_1 and λ_2 in $\pi(s)$ such that the bisectors of λ_1 and λ_2 belong to the same edge e_{ab}. Let b_1 and b_2 be the bisectors of λ_1 and λ_2 respectively. Without loss of generality assume the paths λ_1 and λ_2 start at s and end at f_k and f_l respectively. Suppose along the path λ_1, the vertex v_a precedes the vertex v_b. Now there will be two cases.

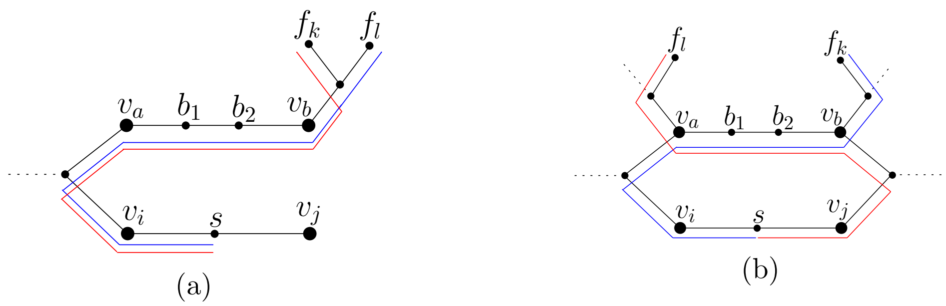

Figure 2.2: Figure showing two cases for the proof of Observation 2.2.2

Case 1: Along the path λ_2, v_a precedes v_b (see Figure 2.2(a)). Suppose distance of b_1 and f_k is δ_1 along λ_1 and distance of b_2 and f_l is δ_2 along λ_2. Now there are two possibilities, $\delta_2 < \delta_1$ or $\delta_2 > \delta_1$. The first possibility contradicts the fact that the arc $\langle b_2, b_1\rangle$ is served by P1 considering the path λ_2. The second possibility contradicts the fact that the arc $\langle b_1, b_2\rangle$ is served by P1 considering the path λ_1. Hence this case can't occur.

Case 2: *Along the path λ_2, v_b precedes v_a* (see Figure 2.2(b)). Without loss of generality assume that b_1 precedes b_2 along λ_1. we consider the path λ_1, then by definition of bisector the arc $\langle b_2, v_b\rangle$ will be served by $P1$. But if we consider the path λ_2, then by definition of bisector the arc $\langle b_2, v_b\rangle$ will be served by P2. Hence contradiction and the observation follows.

□

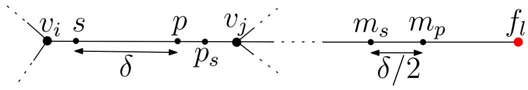

FIGURE 2.3: Positions of s, p_s and f_l

Suppose a facility s_1 of P2 is placed at $s \in e_{ij}$. Let $p_s \in \langle s, v_j\rangle$ be the point closest to s, such that $p_s \in \Gamma \cup V$ (see Figure 2.3). Let $\lambda \in \pi_j(s)$ be a path between s and f_l, where $f_l \in F$ and $m_s \in e_{\alpha\beta}$ be the midpoint of λ. Let p be any point on the arc $\langle s, p_s\rangle$. Suppose the distance between s and p along e_{ij} is δ. If s_1 is now shifted to p, then the path between s_1 and f_l is reduced and hence the mid point is shifted from m_s to a new point m_p. Note that selection of p_s and hence of p ensures that $m_p \in e_{\alpha\beta}$. Now we have the following observation.

Observation 2.2.3. *[2] Distance between m_s and m_p is equal to $\delta/2$ along $e_{\alpha\beta}$.*

Observation 2.2.3 holds for any path $\lambda \in \pi_j(s)$. Similarly for any path $\lambda' \in \pi_i(s)$ consider the point p'_s closest to s, such that $p'_s \in \Gamma \cup V$. Observe that if the facility of P2 is shifted from s to any point $p \in \langle p'_s, s\rangle$ the midpoint of the path λ' is moved to a distance $\delta'/2$, where δ' is the distance between p and s along e_{ij}.

We note that there might be more than one optimal placements by P2, even there might be infinitely many optimal placements by P2. In Figure 2.4 P1 has

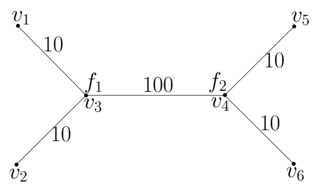

Figure 2.4: Example of facilities placed by P1 and P2

placed two facilities at v_3 and v_4 and any point on the edge joining v_3 and v_4 is an optimal placement by P2. Now we have the following theorem.

Theorem 2.2.1. *[2] There exists an optimal strategy of P2 which belongs to* $\Gamma \cup V$*.*

Proof. Let $\mathring{s}$ be any optimal placement by P2 such that $\mathring{s} \notin \Gamma \cup V$. Suppose $\mathring{s}$ belongs to the edge e_{ij}. Let $p_l \in \langle v_i, \mathring{s} \rangle$ be the point closest to $\mathring{s}$, such that $p_l \in \Gamma \cup V$. Similarly define $p_r \in \langle\ \mathring{s}, v_j \rangle$ be the point closest to $\mathring{s}$, such that $p_r \in \Gamma \cup V$ (see Figure 2.5). Now observe that it is enough to show that either $\mathcal{P}(F, \mathring{s}) \leq \mathcal{P}(F, p_l)$ or $\mathcal{P}(F, \mathring{s}) \leq \mathcal{P}(F, p_r)$.

Now suppose $\mathcal{P}(F, \mathring{s}) > \mathcal{P}(F, p_l)$ and $\mathcal{P}(F, \mathring{s}) > \mathcal{P}(F, p_r)$. Recall that for any placement of facility s by P2, $B_i(s)$ and $B_j(s)$ are the sets of bisectors correspond to the set of paths $\pi_i(s)$ and $\pi_j(s)$. Now based on the emptiness of $B_i(\mathring{s})$ and $B_j(\mathring{s})$ two cases can arise.

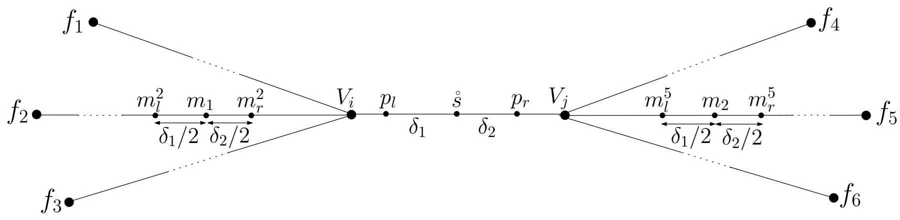

Figure 2.5: Positions of $\mathring{s}$, p_l and p_r

Case 1: $B_i(\mathring{s})=\emptyset$ or $B_j(\mathring{s})=\emptyset$. Without loss of generality assume $B_j(\mathring{s})=\emptyset$. Observe that as $F \neq \emptyset$, $B_i(\mathring{s}) \neq \emptyset$. Now there is no path between $\mathring{s}$ and any facility of P1 via v_j. Thus $\mathcal{P}(F, \mathring{s}) \leq \mathcal{P}(F, p_l)$, which contradicts our basic assumption and hence the result follows.

Case 2: $B_i(\mathring{s}) \neq \emptyset$ and $B_j(\mathring{s}) \neq \emptyset$. Suppose distance between $\mathring{s}$ and p_l be δ_1 and distance between $\mathring{s}$ and p_r be δ_2(see Figure 2.5). Further let $|B_i(\mathring{s})| = k_1$ and $|B_j(\mathring{s})| = k_2$. Consider any path λ in $\pi_i(\mathring{s})$. Let $m_1 \in B_i(\mathring{s})$ be the midpoint of λ. Observe that instead of placing the facility at $\mathring{s}$, if P2 would have placed it at p_l, the length of the path λ between two facilities is reduced by δ_1. From Observation 2.2.3 we know the new midpoint is at a distance $\delta_1/2$ from m_1 along λ(see Figure 2.5). Now as $|B_1(\mathring{s})| = k_1$, k_1 many such paths are there. Hence along all such paths the payoff of $P2$ will be increased by $\frac{k_1 * \delta_1}{2}$. Similarly along the paths in $\pi_j(s)$ the payoff will be decreased by $\frac{k_2 * \delta_1}{2}$. Hence,

$$\mathcal{P}(F, p_l) = \mathcal{P}(F, \mathring{s}) + (k_1 - k_2) * \delta_1/2 \tag{2.1}$$

Similarly if P2 would have placed the facility at p_r, the payoff of P2,

$$\mathcal{P}(F, p_r) = \mathcal{P}(F, \mathring{s}) + (k_2 - k_1) * \delta_2/2 \tag{2.2}$$

Now as, $\mathcal{P}(F, \mathring{s}) > \mathcal{P}(F, p_l)$ and $\mathcal{P}(F, \mathring{s}) > \mathcal{P}(F, p_r)$, from Equation 2.1 and 2.2 we get, $(k_1 - k_2) * \delta_1/2 < 0$ and $(k_2 - k_1) * \delta_2/2 < 0$. As $\delta_1, \delta_2 > 0$, we get $(k_1 - k_2) < 0$ and $(k_2 - k_1) < 0$, hence contradiction and the result follows.

□

Now consider the general problem where P2 is interested in placing $k(\geq 1)$ facilities. Again the goal is to find the optimal placement by P2 on G. Consider any set of placements S by P2. Let $s \in S$ be any arbitrary facility location. Without loss of generality we assume s is on the edge e_{ij}. We refine the definition of $\pi(s)$ by saying that $\pi(s)$ is the set of paths between s and any facility of P1 such that for each path $\lambda \in \pi(s)$, half of the points of λ are closer to some $f_i \in F$ than any other facility point in $F \cup S$ and the rest of the points are closer to s than any other facility point in $F \cup S$. Similarly define $\pi_i(s)$ and $\pi_j(s)$ as the disjoint subset of $\pi(s)$, such that the paths in $\pi_i(s)$ and $\pi_j(s)$ contains v_i and v_j respectively. Accordingly let $B_i(s)$ and $B_j(s)$ are the sets of midpoints of the paths in $\pi_i(s)$ and $\pi_j(s)$ respectively. Next we present a theorem whose proof is somewhat similar to the proof of Theorem 2.2.1.

Theorem 2.2.2. *[2] For One-Round (m, k) Voronoi Game on Graphs there exists an optimal strategy of P2 which belongs to $\Gamma \cup V$.*

Proof. Let $\mathring{S}$ be any optimal placement by P2 such that it contains a point $\mathring{s}$ such that $\mathring{s} \notin \Gamma \cup V$. We show that there exists a placement $S' \subseteq \Gamma \cup V$ by P2 such that $\mathcal{P}(F, \mathring{S}) \leq \mathcal{P}(F, S')$. Without loss of generality assume $\mathring{s}$ belongs to the edge e_{ij}. Let $p_l \in \langle v_i, \mathring{s} \rangle$ be the point closest to $\mathring{s}$, such that $p_l \in \Gamma \cup V$. Similarly define $p_r \in \langle \mathring{s}, v_j \rangle$ be the point closest to $\mathring{s}$, such that $p_r \in \Gamma \cup V$.

Instead of placing a facility at $\mathring{s}$ if P2 would have placed it at p_l or p_r, then using a similar argument like in proof of Theorem 2.2.1, we can prove that either $\mathcal{P}(F, \mathring{S}) \leq \mathcal{P}(F, \mathring{S} \setminus \{\mathring{s}\} \cup p_l)$ or $\mathcal{P}(F, \mathring{S}) \leq \mathcal{P}(F, \mathring{S} \setminus \{\mathring{s}\} \cup p_r)$.

By using this construction repeatedly we substitute each of such $\mathring{s} \in \mathring{S}$, such that $\mathring{s} \notin \Gamma \cup V$, by a point in $\Gamma \cup V$. We end up getting a set $S' \subseteq \Gamma \cup V$, such that $\mathcal{P}(F, \mathring{S}) \leq \mathcal{P}(F, S')$, which completes the proof of this theorem.

□

Note that it is possible to design a simple algorithm, which by checking all subsets of size k of the set $\Gamma \cup V$, finds out the optimal strategy of P2 in exponential time. Nevertheless, in the next section we present a polynomial time algorithm for finding optimal strategy of P2 on trees.

2.3 Optimal Facility Locations on Trees

Consider a tree T=(V, E) such that weight of each vertex and edge is non-negative. Let F=$\{f_1, \ldots, f_m\}$ be the m facilities placed by P1 on T. Also let

$|V|$=n. We are interested in the problem of finding a set of k optimal placement of facilities by P2. Now consider the facilities placed by P1 on T. Observe that these facilities divide T into a finite number of subtrees. Here we deviate a bit from the original definition of subtree and assume the bisectors could also be vertices of a subtree. Refer to these subtrees as *partitions* of T with respect to F. Figure 2.6 shows an example tree and it's partitions with respect to facilities of P1.

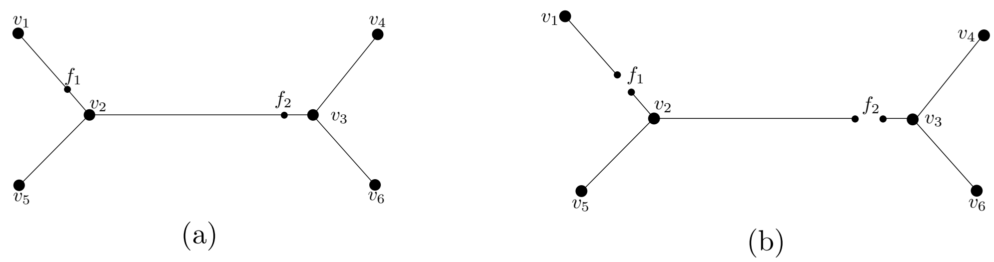

FIGURE 2.6: Example tree and partitions with respect to $\{f_1, f_2\}$.

Let d be the maximum degree of the vertices of T. Note that there could be $O(md)$ many partitions of T. It should be noted that facilities placed by P2 in one partition will not affect the facilities placed in other partitions as the partitions are bounded by either facilities of P1 or leaves. Consider a partition which contain exactly one facility of P1. By placing only one facility in that partition, P2 can serve it totally (see Figure 2.6(b)). Now consider the partitions which contains p facilities of P1, where $p \geq 2$. Let T_i be one such partition. Considering any path as a subtree, let π be the union of the paths of T_i between the facilities of P1. Observe that $T_i \setminus \pi$ is a collection of trees, say $\{\lambda_i : 1 \leq i \leq c\}$. Each tree λ_i shares exactly one vertex, say α_i with π. For example in Figure 2.6(b) the partition containing vertex v_2 is a shared vertex. Note that it is always advantageous for P2 to place a facility on the point v_2, instead of placing it on the edge (v_2, v_5). We have the following observation.

Observation 2.3.1. *For any optimal placement by P2, if a facility of P2 is placed on some λ_k then it is placed on the point α_k.*

Thus for all such λ_i we add the weight of λ_i to the weight of α_i and remove it from T_i. Note that now all the leaves of T_i are facilities of P1. Thus without loss of generality we can assume that if a partition contains more than one facilities of P1, then each of it's leaves contains a facility of P1. We refer to this kind of partitions as *bounded partitions*. Now we define the following routine.

$ALLOC(g_1, \ldots, g_l; p)$**:** Here $g_1, \ldots, g_l$ are monotone concave functions and p is a non-negative integer. Then $ALLOC$ solves the following:

$$\max \sum_{i=1}^{l} g_i(p_i)$$

subject to,

$$\sum_{i=1}^{l} p_i = p$$

Each p_i is a non-negative integer. Frederickson et al.[8] have shown that $ALLOC$ can be solved in $O(l \log p)$ time.

Note that the optimal payoff function on any partition is monotone and concave with respect to number of facilities. Suppose we know the optimal payoff of P2 for placing $\kappa(\geq 1)$ facilities on any partition. Then we can use $ALLOC$ to find the maximal payoff of P2 on T for placing k facilities. Here $g_i(p_i)$ will be the optimal payoff of P2 on i^{th} partition for placing p_i facilities and $p = k$. So if we can compute the optimal payoff of P2 on any partition in polynomial time, we can solve the problem on T in polynomial time. Calculation of optimal payoff of P2 on a partition having exactly one facility of P1 is trivial as discussed before. Remaining is to show that for any bounded partition, the optimal payoff of P2 can be calculated in polynomial time. The idea of our solution is similar to [12].

Now we want to solve the following problem. For any bounded partition we want to find a set of κ placement for a given integer κ such that payoff of P2 is maximized. In the rest of this section instead of solving this problem we will solve a more general

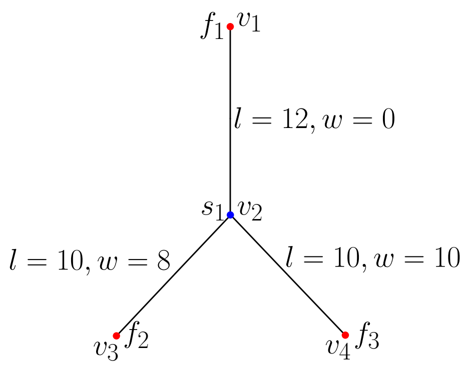

FIGURE 2.7: Service zones of P1 and P2

problem. Let $T(V, E)$ be any tree where each leaf of T is occupied by a facility of P1 and with each edge e_{ij} of T two real values are associated l_{ij}, denotes the length of the edge and w_{ij}, denotes the weight of that edge. Note that lengths of edges are used for calculation of distance and weights are used for computing payoff. For example in Figure 2.7 the payoff of P2 from edges $(v_1, v_2), (v_2, v_3)$ and (v_2, v_4) are $0, 4$ and 5 respectively. Let m' be the number of leaves of T. Also let $|V|$=n'. P2 wants to place κ facilities on the vertices of T such that its payoff is maximized. By Theorem 2.2.2 it suffices to consider only points of $\Gamma \cup V$ as the possible optimal facility locations of P2. For the sake of simplicity add the points of Γ into the set of vertices. Then observe that the new set of vertices contains $O(n'm')$ many points. Recall that for any placement of facility s by P1, bisector is the midpoint of the path λ where λ be any path between s and any facility f_i of P1 such that half of the path is served by s and the other half by f_i. Include the set of bisectors B into the set of vertices and define a new tree $T'(V', E')$ where $V' = V \cup B$ and edges are defined accordingly. Assign the length and the weights of new edges proportionally. Our aim is to choose κ points from $V \subset V'$ such that the payoff is maximized. Choose an arbitrary vertex $v_r \in V$ to be the root of T'. For any v_i let T_i be the subtree rooted at v_i. Now we define a subroutine OPT which recursively places the facilities of P2.

$OPT(T_i, \delta, V_r, E_r)$**:** Here T_i is a tree rooted at v_i, $1 \leq \delta \leq \kappa$ is an integer, $V_r \subseteq V'$

is set of restricted vertices where facilities can not be placed and $E_r \subseteq E'$ is a set of edges with zero weight. $OPT(T_i, \delta, V_r, E_r)$ places δ facilities of P2 on T_i recursively. Let the set of vertices of T_i which are not in V_r be $\{v_{i1}, v_{i2}, \ldots, v_{i\iota}\}$. Also let these vertices are sorted in increasing order of their distances from v_i. For vertex v_{ij} consider the set $\Delta_{ij} = T_i \setminus \lambda_{ij}$, where λ_{ij} be the path between v_i and v_{ij}. Observe that Δ_{ij} is a forest and let Δ_{ij} contains α trees. Let E_{ij} be the set of edges contained in the service zone of the facility of P2 placed at v_{ij}. Let T^l_{ij}, $1 \leq l \leq \alpha$ be the trees of Δ_{ij} and V^l_{ij} be the set of vertices of T^l_{ij}. Also let $g_{ij}(p_l) = OPT(T^l_{ij}, p_l, (V_r \cup \{v_{i1}, v_{i2} \ldots v_{i(j-1)}\}) \cap V^l_{ij}, E_r \cup E_{ij})$, where $\sum_{l=1}^{\alpha} p_l {=} \delta - 1$.

Let $Q_{ij}{=}ALLOC(g^1_{ij}(p_1), g^2_{ij}(p_2), \ldots, g^{\alpha}_{ij}(p_\alpha)){+}W(E_{ij})$, where $W(E_{ij})$ is sum of the weights of the edges E_{ij}. Then $OPT(T_i, \delta, V_r, E_r)$ returns the following, $\max_{1 \leq j \leq \iota} Q_{ij}$.

Suppose in an iteration of OPT a facility is placed at v_{ij}, say at a distance d_{ij} from v_r. Then note that in the following iterations no facility could be placed within a distance d_{ij} from v_r. Consider any subtree T_i rooted at v_i, which is passed as an argument of OPT. Note that T_i is the maximal subtree rooted at v_i and thus all of it's leaves contain a facility of P1. Moreover some of it's vertices are flagged so that no facility can further be placed on them and some of it's edges could have zero weight. First of all we show the independence of the placement of facilities by P2 on the subtrees of Δ_{ij}.

Observation 2.3.2. *For any $\mu \neq \nu$ placement of facilities of P2 in T^μ_{ij} and T^ν_{ij} are independent of each other.*

Proof. Let v^l_{ij} be the root of T^l_{ij}, where $1 \leq l \leq \alpha$. Consider any subtree T^μ_{ij}, such that $v^\mu_{ij} = v_{ij}$, then the service zone of any facility in T^μ_{ij} will be limited within T^μ_{ij}. Moreover as T^μ_{ij} is connected to other subtrees through v^μ_{ij}, facilities of P2 in other subtrees will not get any payoff from T^μ_{ij}. Now consider two subtrees T^μ_{ij} and T^ν_{ij}

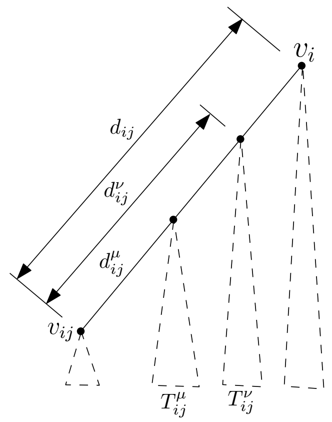

FIGURE 2.8: Independence of T^{μ}_{ij} and T^{ν}_{ij}

such that $v^{\mu}_{ij} \neq v_{ij}$ and $v^{\nu}_{ij} \neq v_{ij}$. Let d^{l}_{ij} be the distance between v_{ij} and v^{l}_{ij}. Also let d_{ij} be the distance between v_i and v_{ij}. Without loss of generality we assume $d^{\mu}_{ij} < d^{\nu}_{ij}$(see Figure 2.8). Note that $d_{ij} \geq d^{\nu}_{ij}$. Now the facility of P2 on T^{ν}_{ij} closest to v^{μ}_{ij} could lie at a distance $d^{\nu}_{ij} + d^{\nu}_{ij} - d^{\mu}_{ij}$ from v^{μ}_{ij}, which is greater than d^{μ}_{ij}. Hence the facility at v_{ij} is closest to v^{μ}_{ij} than any other facilities in T^{ν}_{ij}. Hence any facility placed at T^{ν}_{ij} doesn't get any payoff from T^{μ}_{ij}. Similarly any facility placed at T^{μ}_{ij} doesn't get any payoff from T^{ν}_{ij}, which completes the proof of this observation. $\square$

Note that to get the optimal payoff of P2 from bounded partition T', $OPT(T', \kappa, V' \setminus V, \emptyset)$ should be invoked. Now we will argue about the optimality of the solution return by this subroutine.

Lemma 2.3.1. *$OPT(T', \delta, V' \setminus V, \emptyset)$ calculates the optimal payoff of P2 from T' for placing δ facilities.*

Proof. Note that it is sufficient to prove that for any subtree T_i, the call $OPT(T_i, \delta, V_r, E_r)$ returns the optimal solution. We'll prove this claim using induction on subtree containment. In base case the subtrees containing some constant number of vertices and the payoff of P2 can be calculated in a trivial manner using OPT. Now suppose OPT can calculate the optimal payoff of P2 on all the subtrees of T_i correctly. Now for $1 \leq j \leq \iota$, $OPT(T_i, \delta, V_r, E_r)$ chooses a vertex v_{ij} and places a facility at v_{ij}.

Then it calls the functions $g_{ij}(p_l) = OPT(T_{ij}^l, p_l, V_r \cup \{v_{i1}, v_{i2} \dots v_{i(j-1)}\}, E_r \cup E_{ij})$ recursively. Now by induction hypothesis the functions $g_{ij}(p_l)$, for $1 \leq p_l \leq \delta$ returns the correct solutions. Moreover by Observation 2.3.2 the placement of facilities of P2 in different T_{ij}^l are independent of each other. Hence the further call to $ALLOC$ will return the optimal payoff of P2 Q_{ij} for placing δ facilities, such that the first facility of P2 is placed at v_{ij}. Thus $\max_{1 \leq j \leq \iota} Q_{ij}$ will give the optimal payoff of P2 from T_i for placing δ facilities, which completes the proof of this lemma. $\square$

Though OPT calculates the maximal payoff, if we make the recursive call as described, the time complexity could be exponential. This is because the recursive calls could be made repetitively with same arguments and everytime the function will be computed separately. We'll refine this algorithm by applying dynamic programming. We'll store the values of recursive calls so that no redundant calculations are done. Before that we have to argue that the number of subtrees which are used as the argument of OPT over all calls are bounded polynomially. All the subtrees which have considered are maximal and specified by some sets V_r and E_r. We note that number of maximal subtrees is same as the number of vertices, as for any vertex v there can be at most one maximal subtree rooted at v. Now we'll show that the number of distinct V_r considered over all calls of OPT is polynomial.

Observation 2.3.3. *Number of distinct subsets of vertices V_r considered over all calls of OPT is polynomial.*

Proof. Consider the call of $OPT(T_i, \delta, V_r, E_r)$ on the subtree T_i, rooted at v_i. Now for any V_r, V_r trivially contains the non-vertex bisectors of T_i, as no facility is supposed to be placed on them. Let d_f be the distance of the facility of P2 which is farthest from the root of T'. Also let d_i be the distance of v_i from the root of T' (see Figure 2.9). Hence for the call on T_i, if $d_f \geq d_i$, $V_r \setminus (B \setminus V)$ will be the set of vertices of T_i within a distance $d_f - d_i$ from v_i. Thus for any call of $OPT(T_i, \delta, V_r, E_r)$, V_r can be uniquely specified by a tuple $(v_i, d_f - d_i)$. Now for a fixed subtree T_i, there could be

$O(n'm')$ many distinct d_f. As number of subtrees is $O(n'm')$, the number of such tuples is $O(n'^2m'^2)$, which completes the proof of this observation. □

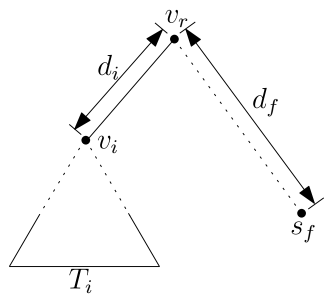

FIGURE 2.9: Farthest facility s_f from v_r

Now we'll argue about the number of distinct E_r over all calls of OPT.

Observation 2.3.4. *Number of distinct subsets of edges E_r considered over all calls of OPT is polynomial.*

Proof. Let's consider the call of $OPT(T_i, \delta, V_r, E_r)$ on the subtree T_i, rooted at v_i. Now E_r is the set of zero weighted edges of T_i. If $E_r = \emptyset$ there is nothing to prove. So consider the case where $E_r \neq \emptyset$. Thus E_r is contained within the service zone of facilities of P2 which are already placed on T'. Let v_c be the vertex of T' which contains the facility of P2 closest from v_i(see Figure 2.10). Then E_r is the exact set of edges of T_i, which are in service zone of the facility placed at v_c. Thus given T_i, E_r could be specified by the vertex v_c. Hence for any call of $OPT(T_i, \delta, V_r, E_r)$, E_r can be uniquely specified by a tuple (v_i, v_c). Note that number of such distinct tuples is bounded by $O(n'^2m'^2)$, as each of v_i and v_c are bounded by $O(n'm')$. Hence the observation follows. □

Observation 2.3.3 and 2.3.4 show that number of subtrees on which the OPT calls are made is bounded by a polynomial. Now we'll discuss the dynamic programming based approach.

We maintain a table M to store the values returned by OPT, correspond to different arguments passed to it. Each row of M corresponds to a subtree as defined

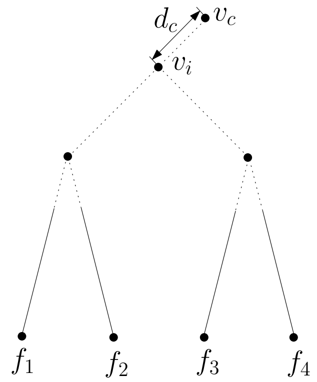

Figure 2.10: Closest facility of P2 from v_i is at v_c

before. Note that we can enumerate all such subtrees. One such enumeration is based on subtree containment relationship. Also for same subtree with different V_r's, say V_r^1 and V_r^2 can be enumerated using subset containment relationship as we know either $V_r^1 \subseteq V_r^2$ or $V_r^2 \subseteq V_r^1$. Similarly for same subtree with different E_r's can be enumerated. M has $\kappa + 1$ columns marked by 0 to κ. Moreover we want the $M(.,.)$ values of all the subtrees of a tree should be calculated before the calculation of $M(.,.)$ values correspond to it. Hence if the calculations are done in bottom-up approach the entries of $M(.,.)$ could be calculated correctly. Once all the $M(.,.)$ values of all the subtrees of a tree T_i are calculated the entries of $M(.,.)$ correspond to T_i can be calculated in polynomial time using $ALLOC$. As M contains polynomial number of entries, all of it's entries can be calculated in polynomial time. Finally the entry of M correspond to the subtree T' with V_r=$V' \setminus V$, $E_r = \emptyset$ and $\delta = \kappa$ will give the anticipated result.

Hence on any bounded partition the optimal payoff of P2 can be calculated in polynomial time. As there are $O(m)$ many such bounded partitions, the total time for calculation of payoffs on bounded partitions is also polynomial. Hence we have the following theorem.

Theorem 2.3.2. *The optimal payoff of P2 correspond to placement of k facilities on any tree can be calculated in polynomial time.*

2.4 Computational Complexity for Graphs

In this section we prove that given a placement of m facilities by P1 determining the optimal placement by P2 for *One-Round* (m, k) *Voronoi Game on Graphs* is $\mathcal{NP}$-hard. Let us call the problem of finding the optimal placement of P2 in *One-Round* (m, k) *Voronoi Game on Graphs* as *Maximum payoff problem*. Now consider the decision version of *Maximum payoff problem*. Given a graph $G = (V, E)$, a set of m facilities F by P1 in G, and a real number δ, we have to decide whether there exists a set of k points S disjoint from F in G such that the payoff of P2, $\mathcal{Q}_2(F, S) \geq \delta$ or not. Clearly the problem is in $\mathcal{NP}$ as given any placement of facility by P1 and P2 it is possible to find out $\mathcal{Q}_2(F, S)$ in polynomial time. To prove that this problem is $\mathcal{NP}$-hard we show a reduction from *Minimum Dominating Set* problem which is known to be $\mathcal{NP}$-hard[9]. At first let us define the *Minimum Dominating Set* problem.

Minimum Dominating Set Problem: Given a graph $G = (V, E)$ a *dominating set* is a set of vertices $S \subseteq V$ such that each vertex in graph G is either in S or is a neighbor of at least one element of S. The problem asks to find such S with minimum cardinality.

Given a graph G and an integer k, the decision version of Minimum Dominating Set Problem asks whether there exist a *dominating set* of size k or not. Now we have the following theorem.

Theorem 2.4.1. *[2] Maximum Payoff Problem is $\mathcal{NP}$-complete.*

Proof. It is already shown that the problem is in NP. Remaining is to prove the NP-hardness. Let $\mathcal{I}$=(G, k) be any valid instance of the *minimum dominating set* where G is an un-weighted graph and k is an integer. We will construct a new weighted graph $G' = (V', E')$ from G by adding a pendant vertex to each of the vertices. Figure 2.11 is showing the construction for an example graph. Let $\tilde{F}$ be

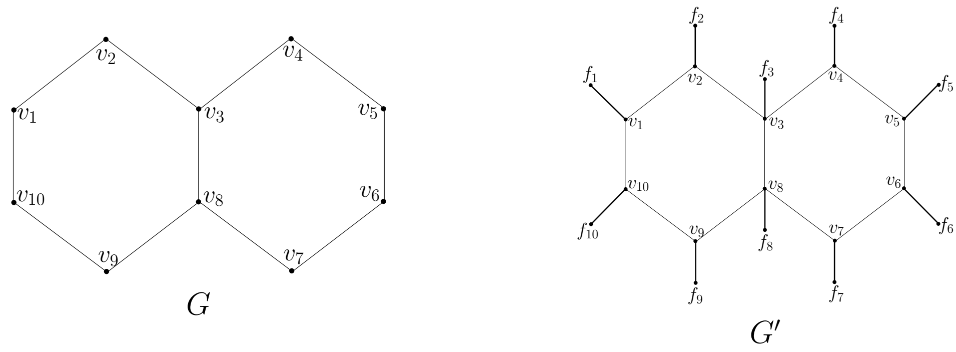

Figure 2.11: Construction of G' from an example graph G.

the set of $|V|$ new vertices. Now $V' = V \cup \tilde{F}$ and $E' = E \cup (v_i, f_i)\ \forall v_i \in V$ and $f_i \in \tilde{F}$. We will assign weight $w_e < \frac{1}{|V|+|E|+k}$ to each edge $e \in E'$ and weight $w_v = 1$ for each vertex $v \in V'$. Now consider the *Maximum Payoff Problem* on G' with $\tilde{F}$ as the placement of P1. Now it is sufficient to prove that there exists a dominating set of size k in G if and only if there exists a set of k points S in G' such that $\mathcal{Q}_2(\tilde{F}, S) \geq |V|$.

Let S be a set of k points in G' such that $\mathcal{Q}_2(\tilde{F}, S) \geq |V|$. Without loss of generality assume $S \subseteq \Gamma \cup V$, if not then using a construction similar to proof of Theorem 2.2.2 S can be modified so that $S \subseteq \Gamma \cup V$ and $\mathcal{Q}_2(\tilde{F}, S) \geq |V|$. Recall that we have assumed for each edge (f_i, v_i) there exists a point p_i very close to f_i such that distance between p_i and f_i is small enough to be considered as zero. Denote the set of all such points as P. Now observe that as weight of each edge is same, $\Gamma \subseteq V \cup P$. Hence $S \subseteq P \cup V$. Now we will construct a new set of placements of facilities S' from S as follows. For all points $s_i \in S \cap V$ add s_i to S'. For all points $s_i \in S \cap P$ let v_j is adjacent to s_i. If $v_j \notin S$ add v_j to S', else add any vertex $v \in V$ such that $v \notin S$ (see Figure 2.12). Observe $S' \subset V$ and $\mathcal{Q}_2(\tilde{F}, S') > \mathcal{Q}_2(\tilde{F}, S) - k * w_e$. Now the payoff $\mathcal{Q}_2(\tilde{F}, S)$ can be written as $\mathcal{Q}_{E'} + \mathcal{Q}_V$ where $\mathcal{Q}_{E'}$ sum of the length of all the arcs those are served by P2 and $\mathcal{Q}_V$ is the number of vertices in the service zone of P2. Observe now $\mathcal{Q}_{E'} \leq (|V|+|E|) * w_e$. Hence $\mathcal{Q}_V \geq \mathcal{Q}_2(\tilde{F}, S) - k * w_e - (|V|+|E|) * w_e$. But $w_e < \frac{1}{|V|+|E|+k}$, that is $w_e * (|V|+|E|+k) < 1$, $|V|$ and $\mathcal{Q}_V$ are integers. Further $\mathcal{Q}_2(\tilde{F}, S) \geq |V|$. Therefore $\mathcal{Q}_V \geq |V|$. Now any vertex $v_i \in V$ will be served by a facility $s_j \in S'$ if either $v_i \in S'$ or s_j is neighbor of v_i. Hence S' is a dominating set of G of size k.

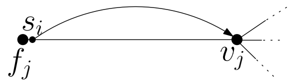

FIGURE 2.12: Formation of S' from S in proof of Theorem 2.4.1

Now consider the case where the graph G has a dominating set D of size k. In graph G', D can be used for placement by $P2$. Every vertex in V is adjacent to one of the vertices of D. So the payoff by $P2$ is at least $|V|$. Hence the result follows. □

2.5 Approximation Bound on Optimal Payoff

In this section we describe a 1.58 factor approximation bound on the optimal payoff of P2. We reduce our problem to *Weighted Maximum Coverage Problem* and use the existing approximation bound for *Weighted Maximum Coverage Problem* to derive an approximation bound for our problem. But before that let us define the *Weighted Maximum Coverage Problem.*

Weighted Maximum Coverage Problem: Given an universe $X=\{x_1, x_2, \ldots, x_n\}$, a family $\mathcal{S}$ of subsets of X, an integer τ and a weight function w_i associated with each $x_i \in X$ the *Weighted Maximum Coverage Problem* is to find τ sets such that total weight of the covered elements is maximized.

The *Weighted Maximum Coverage Problem* is $\mathcal{NP}$-hard, and cannot be approximated within $\frac{e-1}{e} - o(1) \approx 1.58$ factor, under standard assumptions [10]. There is a greedy approximation algorithm for the *Weighted Maximum Coverage Problem*, which at each stage chooses a set, which contains the maximum weighted uncovered elements. Now we are having the following theorem.

Theorem 2.5.1. *[10] The greedy algorithm for Weighted Maximum Coverage Problem achieves an approximation ratio of* $\frac{e-1}{e}$.

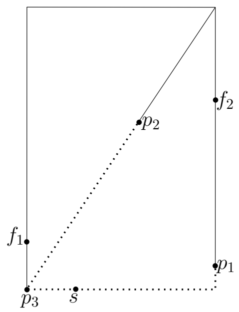

FIGURE 2.13: Service zone of s

Now consider any instance of our problem. Let $G = (V, E)$ be any graph and F be any set of facilities placed by P1 in G. P2 wants to place k new facilities. For the sake of simplicity we assume that the weight of each vertex is zero. But by a simple modification our algorithm can be extended to handle the case when the vertices are having non-zero weights. Now from Theorem 2.2.2 we know that there exists an optimal placement by *P2* which belong to $\Gamma \cup V$. Now consider any placement of facility $s \in \Gamma \cup V$ by P2. Let Ω_s be the set of bisectors correspond to s. For example in Figure 2.13, P1 has placed two facilities f_1 and f_2 and P2 has placed the facility s. The service zone of P2 is shown with dotted lines. Here the set Ω_s will be equal to $\{p_1, p_2, p_3\}$. Define

$$\Omega = \{\cup_{s \in \Gamma \cup V} \Omega_s\} \cup \Gamma$$

From Observation 2.2.2 it is implied that the cardinality of Ω is bounded by $O((\Gamma \cup V)E)$, that is $O((V + E)^2)$. Now from $G = (V, E)$ construct another graph $G' = (V', E')$ in the following way. $V' = V \cup \Omega \cup F$. For each edge $e_{ij} \in E$ which does not contain any point of Ω include that edge to E'. Any edge e_{ij}, which contains one or more points of Ω, $\{\omega_1, \omega_2, \dots \omega_l\}$ sorted along v_i to v_j, add the edges $(v_i, \omega_1), (\omega_1, \omega_2) \dots (\omega_{l-1}, \omega_l), (\omega_l, v_j)$ to E'. Now observe that for any placement s by P2, payoff of P2 will be equal to a set of edges $\tilde{E} \subseteq E'$ of G'.

Now consider the set system where X is equal to E' and for each point $p_i \in \Gamma \cup V$ define the set $S_i \subseteq E'$, such that, S_i is the set of edges that is in service zone of the facility of P2 at p_i. For each edge $e_i \in E'$ the weight of e_i is equal to the length of

e_i. Now run the greedy algorithm for the *Weighted Maximum Coverage Problem* on this set system for $\tau = k$. Then we have this lemma which directly follows from the construction.

Lemma 2.5.2. *[2] An α factor approximation bound for the Weighted Maximum Coverage Problem will produce an α factor approximation bound for Maximum Payoff Problem.*

Now from Theorem 2.5.1 and Lemma 2.5.2 we have the following theorem.

Theorem 2.5.3. *[2] There exist an 1.58 factor approximation algorithm for Maximum Payoff Problem.*

Note that the above mentioned construction can be extended in case if the vertices are having non-zero weights. There the vertices are also member of the ground set. For each point $p_i \in \Gamma \cup V$, the payoff of P2 for the facility placed at p_i would be the total weight of the edges as well as vertices contained in service zone of that facility.

3

Optimal Strategy of First Player

3.1 Introduction

This chapter deals with determination of optimal strategy of P1. To be precise we mainly discuss the results for $(1, 1)$ and $(2, 1)$ game. First we discuss a result on tree showing a lower bound on payoff of P1. Then we discuss about optimal strategy of P1 on trees and we finish this section after discussing about the results on general graphs.

3.2 Bound on Optimal Payoff on Trees

Consider the game where the underlying space is a tree T=(V, E). Let P=$\{p_1, p_2, \ldots, p_\tau\}$ be any set of points on T. Observe that $T \setminus P$ is a set of sub-trees of T. Refer those sub-trees as partitions of T. For example in Figure 3.1, four partitions of an example tree has shown with respect to the set of points $\{p_1, p_2, p_3\}$. Let us denote $T \setminus P$ by $T(P)$. Observe that for any set of m facilities placed by P1 in T, partitions T into at least $m + 1$ partitions. By placing one or more facility in a partition, P2 can get only a portion of that partition. Let us denote the total weight of T by $\mathcal{W}$. Also assume that the tree is rooted at some vertex. Now we are going to show a lower bound on optimal payoff of P1 on trees all of whose vertices are having zero weight. But by a simple extension it can be shown that the bound holds also for the trees with nonzero weights. Now we have the following lemma.

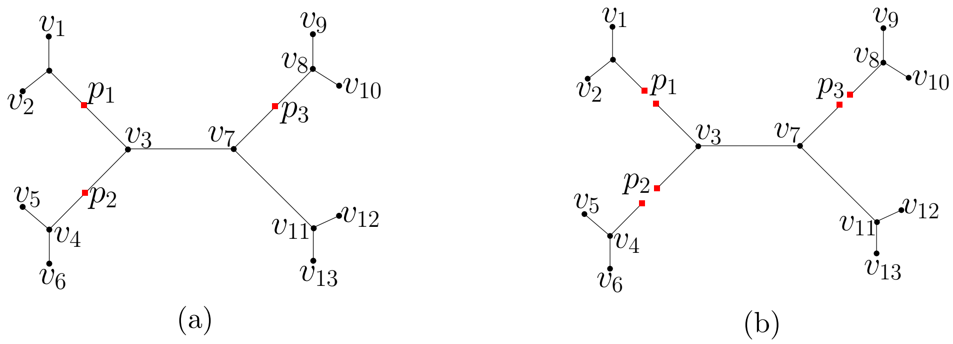

FIGURE 3.1: Example of partition of a tree: (a) Original Tree T and (b) Partitions of T.

Lemma 3.2.1. *[2] For any tree T with all zero weighted vertices, there exists a set of points $P = \{p_1, p_2, \ldots, p_\tau\}$ which partitions T into at least $\tau + 1$ sub-trees, such that weight of each sub-tree $T_i \in T(P)$ is at most $\frac{\mathcal{W}}{\tau+1}$, where τ is any positive integer.*

Proof. Observe that it is enough to prove that given any weighted tree T and a positive integer τ there exist a point $\mathring{p}$, which partitions the tree into two or more parts such that weight of one part is less than or equal to $\frac{\tau*\mathcal{W}}{\tau+1}$ and weight of all other parts are less than or equal to $\frac{\mathcal{W}}{\tau+1}$. Choose an arbitrary vertex of tree as the root of T. Define an extended weight function w_T which maps the vertices of T to $\mathbb{R}$, such that, weight of any leaf vertex v is zero and weight of any internal vertex v_i is equal to $\sum_{v_j}(w_{v_i} + w_T(v_j) + w(v_i, v_j))$, where v_j is child of v_i and $w(v_i, v_j)$ be the weight of the edge (v_i, v_j). Now observe that there will always be a vertex with weight greater than or equal to $\frac{\mathcal{W}}{\tau+1}$ all of whose children are having weight less than $\frac{\mathcal{W}}{\tau+1}$. Denote that vertex by $\breve{v}$. Let the children of $\breve{v}$ be $\{v_1, v_2, \ldots, v_k\}$. Now if for all $1 \leq i \leq k$, $w_T(v_i) + w(\breve{v}, v_i)$ is less than $\frac{\mathcal{W}}{\tau+1}$, then $\mathring{p} = \breve{v}$. Otherwise there exist a child v_j of $\breve{v}$, such that, $w_T(v_j) + w(\breve{v}, v_j) > \frac{\mathcal{W}}{\tau+1}$, but $w_T(v_j) < \frac{\mathcal{W}}{\tau+1}$. But in that case observe that there exists a point p on the edge $(\breve{v}, v_j)$ which partitions the tree into two parts, one having weight $\frac{\mathcal{W}}{\tau+1}$ and other having weight $\frac{\tau * \mathcal{W}}{\tau+1}$. Thus $\mathring{p} = p$ and the result follows. $\square$

Now we have the following corollary.

Corollary 3.2.1. *[2] There exists a placement strategy of P1 such that it always achieves at least* $\frac{m-k+1}{m+1}\mathcal{W}$ *as its payoff for One-Round (m,k) Voronoi Game on Trees, where* $m \geq k$.

Proof. We prove this corollary by proposing an placement strategy of P1. By Lemma 3.2.1 we know that there exists a set F' such that F' partition the tree T in a manner such that each of the partition is having weight at most $\frac{\mathcal{W}}{m+1}$, where $|F'|{=}m$. Suppose P1 places its facilities on the points of F'. By placing k facilities P2 can occupy k partitions. Payoff of P2 in that case would be $\frac{\mathcal{W}}{m+1}k$. Hence the payoff of P1 is at least $\frac{m-k+1}{m+1}\mathcal{W}$, which completes the proof of this corollary. $\square$

Now consider a restricted version of this game where $k = 1$, i.e P2 places only one facility. Also consider the class of *Star* trees with $m + 1$ edges of equal weight. For this case, an optimal strategy of P1 is to place a facility at the *central vertex* and the remaining $m - 1$ to anywhere on the *Star*. On the other hand P2 chooses a point as close as possible to the *central vertex*, preferably on an edge which doesn't contain any facility of P1, as its optimal strategy. Thus service zone of P2 is limited within an edge and payoff of P1 is $\frac{m}{m+1}\mathcal{W}$. Hence the bound of Corollary 3.2.1 is tight.

3.3 Optimal strategy on trees

In this section we discuss about the optimal strategy of P1 for $(1,1)$ and $(2,1)$ games. Here the underlying space is a tree. The next two subsections contain the strategies for $(1,1)$ and $(2,1)$ games respectively.

3.3.1 Optimal strategy for $(1,1)$ game

Consider any facility location f_1 of P1. f_1 could be a vertex or some point on an edge. Also consider the branches of f_1. P2 should place it's facility as close as possible to f_1 on the maximum weighted branch of f_1 for maximum gain. As P1 try to minimize the maximum gain of P2, for placing it's facility it should choose the point where the weight of the maximum branch is minimized. For any weighted tree T let $\mathcal{W}_T$ be the weight of T. Call a point p on T as *central point* if weight of all the branches of p is at most $\frac{\mathcal{W}_T}{2}$. Now we have the following observation.

Observation 3.3.1. *For any tree a central point always exists and is unique.*

Proof. We describe a procedure that essentially finds out a central point. This is more than sufficient to prove the existence of central point. Consider any weighted tree T. Without loss of generality we assume T has a non-leaf node. Otherwise the tree contain only one edge and the midpoint of that edge is the unique central point. Start with any non-leaf node v of T. There could be two cases. All the branches of v are having weight at most $\frac{\mathcal{W}_T}{2}$ or their exists a branch B_v of v whose weight is more than $\frac{\mathcal{W}_T}{2}$. In first case v is the anticipated central point. In second case let u be the neighbour of v on B_v. If the branch of u which contain v has weight more than $\frac{\mathcal{W}_T}{2}$, then the edge (u, v) must contain a point which has two branches of weight exactly $\frac{\mathcal{W}_T}{2}$. Hence that is a central point of T. Otherwise u is a non-leaf node. Repeat the procedure for u. Note that this procedure always stops and returns a central point.

For any point q, q must be in some branch of a central point p. By definition weight of all the branches of p are at most $\frac{\mathcal{W}_T}{2}$. Thus the branch of q which contain p must have weight more than $\frac{\mathcal{W}_T}{2}$ and so q can't be a central point, which proves the uniqueness of central point. Hence the observation follows. □

Now we argue that the central point is the optimal strategy of P1 for $(1, 1)$ game.

Lemma 3.3.1. *For One-Round* $(1, 1)$ *Voronoi Game on Tree the optimal facility location of P1 is the central point.*

Proof. Consider the central point p_c of any tree T. If P1 places it's facility on p_c, payoff of P2 is always less than $\frac{\mathcal{W}_T}{2}$. But if it places in some other point and P2 places on p_c, then payoff of P2 is at least $\frac{\mathcal{W}_T}{2}$. Hence p_c is the optimal facility location of P1 which completes the proof of the lemma. □

3.3.2 Optimal strategy for $(2, 1)$ game

Consider any placement of P1 for $(2, 1)$ game on tree T. The optimal placement of P2 must be either on the path between the two facilities of P1 or on a point as close as possible to facilities of P1. Thus their must exist two leaf nodes l_1 and l_2 such that the path between l_1 and l_2 contains all the three facilities. Hence an optimal placement of P1 on T should also be optimal for the scenario where the facilities of P1 are restricted to be placed on some path between two leaves of T. Suppose we are able to calculate optimal strategies of P1 on any such paths between two leaves, then taking the maximum payoff of P1 among all such paths gives the optimal payoff of P1 on T. It is remaining to show what should be the optimal strategy of P1 on such paths which we discuss in the rest of this subsection.

Consider any path π between two leaves l_1 and l_2 of T. We solve the problem of determining the optimal strategy of P1 when the facilities of P1 are restricted to be placed on π only. Say v be a vertex on π. Consider the branches of v which is not in π. As no facilities could be placed on these branches we add the weight of these branches to weight of v. Thus without loss of generality we assume that the underlying space is now a path containing some vertices. Note that after placement, facilities $\{f_1, f_2\}$ of P1 divides the path into three portions. One between l_1 and f_1, one between f_1 and f_2 and the other between f_2 and l_2. If f_1 or f_2 are vertices, those portions don't contain weight of those vertices. Optimal payoffs of P2 from first and the third portion are the weight of those portions. Say the distance between f_1 and f_2 is d. Also let W_d be the weight of a maximum weighted $\frac{d}{2}$ length interval among all the $\frac{d}{2}$ length intervals of the second portion. Then optimal payoff of P2 from the second portion is W_d. The optimal payoff of P2 is the maximum among the payoffs from those three portions.

Now consider a simpler problem than the one, which we are supposed to solve. Suppose P1 has already placed one facility on π. We need to find on which point on the path between f_1 and l_2, it should place the other facility so that it's payoff is maximized. Define two functions f and g, where $f(x)$ and $g(x)$ denotes the payoffs of P2 from second and third portion while f_2 is at a distance x from f_1. It is easy to see that $g(x)$ is a piecewise linear function with jumps on the vertices and is strictly decreasing in nature. On the other hand $f(x)$ is a strictly increasing piecewise linear function. Consider the scenario in Figure 3.2. When x=4, P2 can occupy v_1 and $f(x)$=12. The payoff increases linearly until x just becomes greater than 8. Here P2 can occupy v_2 and hence it's payoff is 24. So the point as close as possible and on right to v_2 is a jump point for f. The payoff increases linearly until x=10. At x=10, P2 can occupy both of the vertices v_1 and v_2, as their distance is 5 and P2 can occupy the maximum weighted interval of length 5. So if d is the distance between two vertices, then x=2d could be a jump point for f. No other jump points are possible for f. Hence f and g are having $O(n^2)$ and $O(n)$ jump points respectively. Note that the optimal position of f_2 is the lowest point correspond to upper envelope of f and g.

Figure 3.2: Optimal payoff of P2 from the second portion.

Suppose the jump points correspond to the function g are sorted in increasing order. As g is strictly decreasing in nature we can use a variant of binary search to find the optimal strategy of P1. Say g has k jump points. Denote the piece of linear segments of g by (s_i, t_i), where $1 \leq i \leq k+1$. Then we perform binary search to find out the segment (s_i, t_i) such that either of the following four cases is satisfied, i. $g(s_i)$=$f(s_i)$, ii. $g(t_i)$=$f(t_i)$, iii. $g(s_i) > f(s_i)$ and $g(t_i) < f(t_i)$ and iv. $g(s_i) > f(s_i)$ and $g(t_i) > f(t_i)$ and $g(s_{i+1}) < f(s_{i+1})$ and $g(t_{i+1}) < f(t_{i+1})$. All the four cases are shown in Figure 3.3. For each of such probed segment the evaluation of functions on the points s_i, t_i, s_{i+1} and t_{i+1} takes $O(n)$ time. As k=$O(n)$, in total $O(n \log n)$ time is needed.

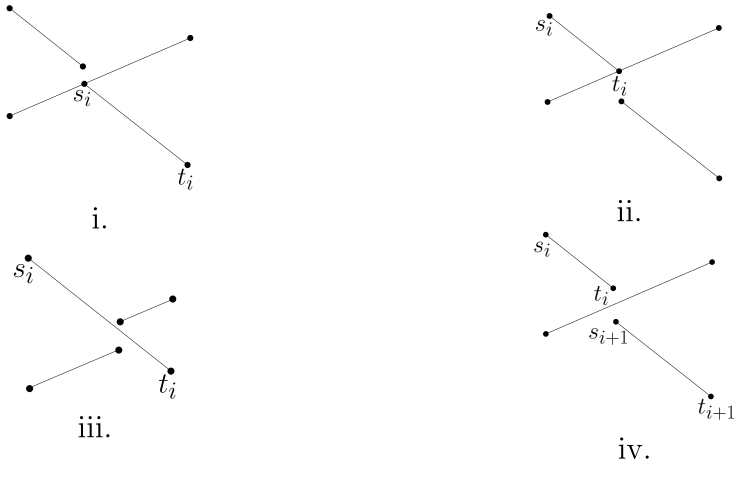

Figure 3.3: Cases correspond to the functions f and g

Now we consider the original problem of finding optimal facility locations of P1 on the path between l_1 and l_2. Let f_1 is placed at a distance y from l_1. Then define the function h such that $h(y)$ denotes the payoff of P2 from the portion between l_1 and f_1. Suppose we move the facility f_1 from l_1 to l_2 and observe $h(y)$ on each non-leaf vertex. For any placement of f_1 we know the optimal placement of f_2. Let x be the variable which denotes the distance between f_1 and optimal location of f_2. Then $h(y)$ increases with increase of y, but $f(x)$ and $g(x)$ monotonically decrease. There will be some vertex v_1 and v_2 such that $h(v_1) < \max(f(x), g(x))$ and $h(v_2) \geq \max(f(x), g(x))$. Consider any other point p on the right of v_2. Maximum payoff of P2 is $h(p)$ if f_1 is at p, which is greater than $h(v_2)$, the maximum payoff of P2, if f_1 is at v_2. Thus optimal location of f_1 can't lie on right of v_2. Similarly it can't be on the left of v_1 either. Hence optimal location of f_1 must be some point on the edge (v_1, v_2). We try to find out the point where exactly the transition occurs. When f_1 is at v_1 two cases may occur, f_2 is on a point of some edge or on a vertex. For the first case if f_1 is shifted a small ϵ unit, then f_2 will be shifted $\frac{\epsilon}{3}$ unit($f(x)$=$g(x)$ for this case). Hence $f(x)$ or $g(x)$ will be changed linearly until f_2 reaches to some vertex. Hence if the distance between f_2 and next vertex is d, a jump will occur when f_1 is at a distance y+$3d$. Hence for each vertex on right of f_2 we get a jump point. For the second case, let f_2 is on a vertex v_f. If $g(x) > f(x)$ with an ϵ unit increase of y, x is also increased by ϵ unit. Here optimal payoff of P2 is decreased linearly until $g(x)$=$f(x)$ or f_1 is on v_2 or f_2 is on some other vertex. Hence if distance between f_2 and some vertex on right of it is d, then y+d is a jump point. So in this case also for each vertex on right of f_2 we get a jump point. For the remaining case where $g(x) \leq f(x)$, with the increase of y, f_2 remains on v_f until the $\frac{x}{2}$ length interval containing v_f as it's right endpoint have lesser or equal weight than a maximum weighted $\frac{x}{2}$ length interval not containing v_f. Now the interval containing v_f becomes lesser than a maximum weighted interval if there exists another vertex $v_{f'}$ such that $d(v_f, v_{f'})$=d and x is just lesser than $2d$. Hence in this case for each such $v_{f'}$ one jump point may occur. Hence there could be $O(n)$ such jump points.

Thus (v_1, v_2) may contain $O(n)$ jump points where the slope of optimal payoff of P2 gets changed. If all of these points are sorted we can apply binary search to find the points where the transition occurs. Here for each of the $O(\log n)$ pass we need to find the optimal position of f_2. Thus the searching time is $O(n \log^2 n)$ and hence optimal strategy of P1 on a path can be computed in $O(n \log^2 n)$ time. Hence the optimal strategy of P1 on the tree T can be computed in $O(n^3 \log^2 n)$ time.

It's not hard to see that the same strategy on the path can be extended to $(m, 1)$ game. Thus the problem on the path could be solved in $O(n \log^m n)$ time for $(m, 1)$ game.

3.4 Optimal strategy on graphs

This section is dedicated to finding optimal facility locations of P1. Here we talk about the strategy for $(1, 1)$ game on general graphs.

3.4.1 Characterization of Optimal Facility Locations

In this subsection we discuss about the strategy of P1 for *One-Round* $(1, 1)$ *Voronoi Game on Graphs*. Here P1 is interested in choosing a facility location so that it could maximize it's payoff. Define a function f, such that for any point p on the graph G, $f(p)$ denote the optimal payoff of P2. From Theorem 2.2.1 we know that there exists an optimal facility location of P2 which belong to the set $\Gamma \cup V$ with respect to a placement of P1. Note that the set $\Gamma \cup V$ may get changed with respect to placement of P1. We refer to this set of points as event points. For any placement of P1 calculate the payoff of P2 on these event points. Then consider the upper envelope of all these payoff functions. Then note that this upper envelope is exactly equal to the function f. The placement of P1 which minimizes the maximum payoff of P2 is the optimal placement of P1. Hence a point correspond to minimum value of the upper envelope is an optimal facility location of P1.

For any vertex $v \in V$, let $g^v(p)$ denotes the payoff of P2 with respect to the placement of P1 on p and placement of P1 on v. Note that $g^v(p)$ is piecewise linear. And the point where slopes of the linear functions get changed, either it is a vertex itself or some bisector must touch a vertex. Hence if the slope gets changed on a point $p \notin V$, then $\exists v_1$ such that $d(p, v_1)$=$d(v_1, v)$. For any two vertices v_i and v_j, let E_1^{ij}=$\{q : d(v_i, q) = d(v_i, v_j)\}$. Also let E_1^i=$\bigcup_{v_j \in V} E_1^{ij}$ and E_1=$\bigcup_{v_i \in V} E_1^i$. Then $E_1^i \cup V$ denotes the set of points where the slopes of the function $g^{v_i}(p)$ could get changed.

Observation 3.4.1. *For any vertex v_i, $|E_1^i|$=$O(|V||E|)$.*

Consider the back up points with respect to a placement of P1. If the placement is on an edge, then there will be exactly two such points, else if it is on a vertex, then the number of back up points is same as degree of that vertex. If the placement is on a vertex then the optimal payoff of P2 on back up points can be calculated by calculating payoff of P2 on all the back up points with respect to that vertex. Consider a placement of P1 on a non-vertex point $p \in (v_i, v_j)$, where $i < j$. Let b_l^p and b_g^p be the back up points closer to v_i and v_j respectively. Note that here if the facility of P1 is moved along (v_i, v_j), then the payoff on b_l^p and b_g^p remain same until some bisector touches a vertex or the facility is on a vertex. We characterize the points where these payoff can get changed. For any two points q and r, let $d^P(q, r)$ be the distance between q and r along the path P. Then consider the set of points E_2=$\{q : \exists v, d^{S_1}(p, v) = d^{S_2}(p, v)\}$, where if $q \in (v_i, v_j)$, then S_1 contains v_i, but not v_j and S_2 contains v_j, but not v_i. Then note that if the placement of P1 is shifted along an edge the bisetors could touch each vertex at most once. Hence we have the following observation.

Observation 3.4.2. $|E_2|$=$O(|V||E|)$

We get two payoff functions correspond to b_l^p and b_g^p with respect to placement of P1 on p. Hence we can calculate all the payoff functions correspond to back up points.

Let $\Gamma_1 \subseteq \Gamma$ be the set of points as close as possible to the facility of P1. Define Γ_2=$\Gamma \setminus \Gamma_1$. Now consider the payoff functions of P2 with respect to points on Γ_2. Consider any placement $p \in (v_i, v_j)$. Also consider the payoff of P2 on a point $q \in (v_k, v_l)$ of Γ. Then if the facility of P1 is shifted along (v_i, v_j), q is also shifted and payoff may get changed. Note that the payoff function with respect to q is piecewise linear. Also note that the slope of this function gets changed either when p or q reaches to a vertex or some bisector touches a vertex. Now the bisectors may touch any vertex at most once. Hence the payoff could be changed in at most $O(n)$ points. Note that with the movement of p, after when q touches a vertex new event points could be generated, while the old one may get vanished. So the payoff function on q is defined for those points p for which q remains on (v_k, v_l). So for all the edges the slope of the payoff function with respect to a point of Γ_2 could be changed at most $O(n^2)$ times. Hence we can calculate all the payoff functions correspond to points in Γ_2.

Thus we get $O(n^2)$ linear piecewise functions with $O(n^2)$ jump points in each function. As these functions are like monotone chains they could intersect in at most $O(n^4)$ points. Thus the upper envelope of these functions can also contain at most $O(n^4)$ jump points. Hence we have the following lemma.

Lemma 3.4.1. *Their exists $O(n^4)$ points which must contain an optimal facility location of P1.*

4

Conclusion and Future Research

In this chapter we summarize the work in this thesis and propose some unsolved problems as possible future works.

4.1 Contributions

Voronoi Game is having a rich literature since it was introduced. Many researchers have worked on many variants of this game. We select transportation network as the underlying space of the game and hence modelled using planar graph. Now we summarize the works of this thesis.

In Chapter 2 we discussed about the optimal strategy of second player. At first optimal facility locations of P2 are characterized. Then using that result we propose a polytime algorithm to determine optimal strategy of P2 on tree network. Thereafter we prove that the problem of determining optimal strategy of P2 on general graph is $\mathcal{NP}$-Hard. We concluded the chapter with a constant factor approximation algorithm for the above mentioned problem.

Chapter 3 is fully devoted for describing optimal strategy of P1. At first we propose a lower bound on optimal payoff of P1. Then we describe optimal strategy of P1 for $(1, 1)$ and $(2, 1)$ game on tree network. Lastly we describe an optimal strategy of P1 for $(1, 1)$ game on graph.

4.2 Open Problems

Our study leaves several open problems and directions of future research. Some of the immediate open problems resulted from the study in this thesis are indicated below.

- For general graph it is proved that to determine optimal strategy of P2 is $\mathcal{NP}$-Hard. But what can be said about the problem when we consider simpler structures like Cactus Graph, Chordal Graph etc.?
- Is it possible to give a better algorithm for the above mentioned problem when the underlying space is tree?
- Is it possible to give a tighter approximation bound for the same problem on graph?
- What should be the optimal strategy of P1 for (m, n) game? Is it possible to give a polynomial time algorithm for this problem or prove it as $\mathcal{NP}$-Hard.
- What can be said about the said problem when the underlying space is tree?
- We considered one round game. What if there are more rounds involved?

- What if the facilities of P1 and P2 are placed alternatively?
- What can be said about existence of pure Nash equilibrium on this version of voronoi game?

Bibliography

[1] Hee-Kap Ahn, Siu-Wing Cheng, Otfried Cheong, Mordecai J. Golin, and René van Oostrum. Competitive facility location: the voronoi game. *Theor. Comput. Sci.*, 310(1-3):457–467, 2004.

[2] Sayan Bandyapadhyay, Aritra Banik, Sandip Das, and Hirak Sarkar. Voronoi game on graphs. *WALCOM*, pages 77–88, 2013.

[3] Aritra Banik, Bhaswar B. Bhattacharya, and Sandip Das. Optimal strategies for the one-round discrete voronoi game on a line. *Journal of Combinatorial Optimization*, pages 1–15, 2012.

[4] Aritra Banik, Bhaswar B. Bhattacharya, Sandip Das, and Sreeja Das. Two-round discrete voronoi game along a line. *FAW-AAIM*, pages 210–220, 2013.

[5] Aritra Banik, Sandip Das, Anil Maheshwari, and Michiel H. M. Smid. The discrete voronoi game in a simple polygon. *COCOON*, pages 197–207, 2013.

[6] Otfried Cheong, Sariel Har-Peled, Nathan Linial, and Jirí Matousek. The one-round voronoi game. *Discrete & Computational Geometry*, 31(1):125–138, 2004.

[7] Sándor P. Fekete and Henk Meijer. The one-round voronoi game replayed. *Comput. Geom.*, 30(2):81–94, 2005.

[8] Greg Frederickson and Donald Johnson. Optimal algorithms for generating quantile information in x+y and matrices with sorted columns. *13th Ann. Conference on Information Science and Systems*, pages 47–52, 1979.

[9] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[10] Dorit S. Hochbaum. *Approximation algorithms for $\mathcal{NP}$-Hard problems.* PWS publishing company, 1996.

[11] Masashi Kiyomi, Toshiki Saitoh, and Ryuhei Uehara. Voronoi game on a path. *IEICE Transactions*, 94-D(6):1185–1189, 2011.

[12] Nimrod Megiddo, Eitan Zemel, and S Louis Hakimi. The maximum coverage location problem. *SIAM Journal on Algebraic Discrete Methods*, 4(2):253–261, 1983.

[13] Sachio Teramoto, Erik D. Demaine, and Ryuhei Uehara. The voronoi game on graphs and its complexity. *J. Graph Algorithms Appl.*, 15(4):485–501, 2011.

Bio-Data

Full Name	:	Sayan Bandyapadhyay

Education

July, 2011 – Till Date	:	M.Tech in Computer Science Indian Statistical Institute, Kolkata.
August, 2007 – May, 2011	:	B.Tech in Computer Science Govt. College of Engg. & Leather Technology, Kolkata.

Research Experience

Mar, 2013 – Till Date	:	Voronoi Game on Trees
Oct, 2012 – Feb, 2013	:	New Variations of facility location problems
Dec, 2011 – Sep, 2012	:	Voronoi Game on Graphs
June, 2010 – May, 2011	:	Job Modelling Using Clone Detection Technique

Publications

1. Voronoi Game on Graphs, with A. Banik, S. Das and H. Sarkar, WALCOM-2013

2. Voronoi Game on Graphs, with A. Banik, S. Das and H. Sarkar, submitted to Theoretical Computer Science