BSc Thesis Applied Mathematics

Submodular functions and M-convex sets

Alex van Tilburg

Supervisor: Georg Loho and Pieter-Tjerk de Boer

August, 2022

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

This paper is written as Bachelor's Assignment for my Applied Mathematics and Technical Computer Science bachelor. This is also the first time I have heard of submodular functions. This paper is aimed to show some beauty of submodular functions to people who also did not have heard of submodular functions before. Furthermore, it includes the somewhat less known M-convex sets, since there are beautiful alternative interpretations from them, which we will investigate.

Submodular functions and M-convex sets in Graphs

Alex van Tilburg*

August, 2022

Abstract

This paper covers a few simple examples of submodular functions with the proof of their submodularity. All examples relate to graphs, and we will examine the similarities between the different submodular functions. Furthermore, the paper covers M-convex sets. It shows for some submodular functions that the lattice points can be interpreted differently. The paper gives proofs and algorithms to obtain the alternative definition, of the lattice points from the M-convex sets of the cut and coverage functions.

Keywords: Submodular function, Graph, M-convex set

1 Introduction

This paper is about submodular functions and their corresponding M-convex sets. The goal is to list a few examples and show similarities between different classes of submodular functions. Moreover, the alternative definition of a few M-convex sets will be proven. In Section 2, I will define submodular functions, and Section 3 is about different classes of submodular functions. Finally, proof of the alternative definition of the lattice points of the M-convex sets will be presented in Section 4. These M-convex sets are derived from the submodular functions in Section 3.

1.1 Submodular functions

In the first part, we will look at different classes of submodular functions. These are set functions that satisfy a particular system of inequalities. These inequalities are defined in Section 2. This property is general and applicable in many fields. In this paper, we will look at a few of those applications.

1.1.1 Applications

The concept of submodular functions was proposed in 1970 when Edmonds generalized the properties of matroid polyhedra. The submodular functions apply to many areas, such as theoretical computer science. The submodular functions pop up in research concerning different applications such as selecting camera points [13], epidemics such as COVID-19 [2], game theory [10], and machine learning [1]. In this paper, we will look at a number of basic examples from the book *Connections in Combinatorial Optimization* [5], the influence function from social networks [7, 11], and the minimum cost connect function, as well as their proof of submodularity.

*Email: alexvantilburg@outlook.com

1.1.2 Optimization

Submodular functions are easy to optimize. At first, Cunningham developed an algorithm that only worked for a special case of submodular functions. [3] This algorithm was later extended to find the minimum value of a general submodular function in polynomial time. Later, better algorithms were developed that were strongly polynomial, [4] making it easy to minimize any submodular function.

1.2 M-convex sets

M-convex sets are a notion introduced by Kazuo Murota [9]. An M-convex set is a set with lattice points that satisfy a system of inequalities. These sets are applicable in the same areas as submodular functions, such as game theory [10]. Their importance lies in their relation to submodular functions. Moreover, the corresponding submodular function can be constructed from any M-convex set. We will talk about those sets in the second half of the paper. We will mainly investigate the alternative definition of the lattice points for some M-convex sets.

1.2.1 Interpretations

The main goal of this paper is to find another way of defining the lattice points, in particular M-convex sets, rather than the geometric way. First, the geometric way of defining the lattice points of all M-convex sets will be investigated. Then, the lattice points will be defined differently for the cut, coverage, and largest degree functions. This new way of defining the lattice points makes finding all lattice points more intuitive. The lattice points of the cut function can be defined with orientations, and the lattice points of the coverage function can be defined with distributions. This paper will prove that these ways of defining the lattice points are the same as the geometric definition. An algorithm will be obtained to construct a distribution corresponding to a given lattice point for the coverage function.

2 Submodular function

In this section, we will look at the definition of submodular functions. These are set functions with a special property formulated in Definition 2.1.

Definition 2.1 (submodular functions)**.** For a finite set E let $f : 2^E \to \mathbb{R}$ then f is a submodular function if and only if the following holds for every $X, Y \subseteq E$

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \tag{1}$$

[8]

A submodular function is a function with domain 2^E, where E is the finite set, on which the submodular function is defined, for example, $\{0, 1, 2\}$. Then 2^E is an alternative notation for the power set of E. In this example, 2^E would be $\{\emptyset, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}$. An example of a submodular function on 2^E would be the cardinality function. So this means that $f(A) = |A|$ for any $A \subseteq E$. For example, take $X = \{1, 2\}$ and $Y = \{3\}$. Then (1) holds.

$$\begin{aligned} f(\{1,2\}) + f(\{3\}) &\geq f(\{1,2\} \cup \{3\}) + f(\{1,2\} \cap \{3\}) \\ |\{1,2\}| + |\{3\}| &\geq |\{1,2,3\}| + |\emptyset| \\ 2 + 1 &\geq 3 + 0 \\ 3 &\geq 3 \end{aligned}$$

This is not a proof, since it does not show whether works for every $X, Y \subseteq E$. Since the inequality is always equal, this function is modular. In this paper we will only look at submodular functions.

In the next section we will go over the different submodular functions with their proofs, such as the cut-function (Example 1.3 in [8]), coverage function (Example 1.4 in [8]) and influence function [7, 11]. Before that, we will show that the following definition is equivalent to Definition 2.1.

Corollary 2.2 (alternative definition submodular functions)**.** *Let E be a finite set and let $f : 2^E \to \mathbb{R}$ then f is submodular if and only if for every $A \subseteq B \subseteq E$ the following holds for all $x \notin B$.*

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B) \tag{2}$$

Proof. First, assume $f : 2^E \to \mathbb{R}$ is submodular and let $A, B \in 2^E$ such that $A \subseteq B$. Now we can use (1) by substituting X with $A \cup \{x\}$ and Y with B. We obtain the following.

$$f(A \cup \{x\}) + f(B) \geq f(A \cup \{x\} \cup B) + f((A \cup \{x\}) \cap B) \tag{3}$$

Since $A \subseteq B$, $A \cup \{x\} \cup B = B \cup \{x\}$. Furthermore, because $A \subseteq B$ and $x \notin B$, $(A \cup \{x\}) \cap B) = A$ is true. We obtain the following if we substitute this in (3).

$$f(A \cup \{x\}) + f(B) \geq f(B \cup \{x\}) + f(A)$$

Then we reorder the terms to get (2).
Now assume (2) for all $A, B \in 2^E$ and $A \subseteq B \subseteq E$. We will use telescoping to prove that (1) holds. Let $X, Y \in 2^E$ be two arbitrary subsets. Since X is finite, therefore is $X \setminus Y$. Therefore, we can have an ordered sequence $\{p_1, \ldots, p_n\} = X \setminus Y$. Let us construct the following sequences of subsets.

$$P_k = \begin{cases} X \cap Y & k = 0 \\ P_{k-1} \cup \{p_k\} & k > 0 \end{cases}$$

We will do the same for the finite set Y.

$$Q_k = \begin{cases} Y & k = 0 \\ Q_{k-1} \cup \{p_k\} & k > 0 \end{cases}$$

It can be easily shown by induction that $P_n = X$ and $Q_n = X \cup Y$, since it adds all elements of $X \setminus Y$. Furthermore, $P_k \subseteq Q_k$ for all $0 \leq k \leq n$. So from (2) we know $f(P_i) - f(P_{i+1}) \leq f(Q_i) - f(Q_{i+1})$ for all $i \in 0, 1, \ldots, n-1$. So the following holds term-wise.

$$\sum_{i=0}^{n-1} f(P_i) - f(P_{i+1}) \leq \sum_{i=0}^{n-1} f(Q_i) - f(Q_{i+1})$$

By telescoping, we have the following,

$$f(P_0) - f(P_n) \leq f(Q_0) - f(Q_n)$$

By definition, the sets are equal to the following.

$$f(X) - f(X \cup Y) \leq f(X \cap Y) - f(Y)$$

We can reorder the terms to get (1). □

In this paper, we will use both Definition 2.1 and Collary 2.2 to prove certain functions are submodular.

3 Classes of submodular functions

First, we will go through different examples of submodular functions. For each function, we will look at the proof of their submodularity. For each class, we will also construct an example from which we will see if it holds for certain values to understand the behavior better. At the end of each subsection, we will show a few interesting practical applications of the corresponding class of submodular functions.

3.1 Cut function

In this section, we will look at a function inspired by d_z from chapter 1.2 of the book Connections in Combinatorial Optimization [5]. We will examine a simpler version, called the cut function. This function is known and used in many areas, such as the Max-Flow Min-Cut Theorem, which tells that in a digraph with capacities, the maximal flow can be calculated by finding the minimal cut [6]. In this section, we will look at an example of the cut function, then prove that the cut function is submodular in an intuitive sense. We will use this cut function to find the corresponding M-convex set later in the paper.

This function is a function defined on a graph. A graph consists of vertices and edges connecting those vertices. Figure 1 is an example of a graph. We will denote a graph

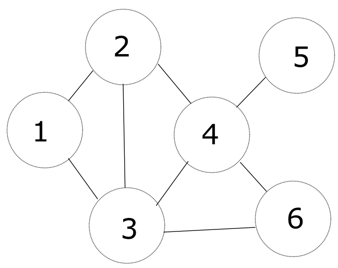


FIGURE 1: example graph

as follows: $G = (V, E)$. This means that the graph has a set of vertices V and a set of undirected edges E. An edge is a tuple of vertices connected by this edge and denoted as (v_1, v_2). Now we can define the cut function.

Definition 3.1 (cut function)**.** Given a graph $G = (V, E)$ and $f : 2^V \to \mathbb{N}$ then f is a cut function if and only if the following is true.

$$f(A) = |\{e \in E | (v_1, v_2) = e, v_1 \in A \wedge v_2 \in \overline{A}\}|$$

This is the cut function. We can interpret this by the number of edges that connect A with its complement. We have to cut this number of edges to disconnect the subset from the rest of the graph.

Example 3.2. Take a graph $G = (V, E)$ as in Figure 2. If you take the following input $\{1, 2, 4\}$, which means the vertices 1, 2 and 4. As you can see in Figure 2, all the edges that connect vertices of $\{1, 2, 4\}$ with vertices of $\{3, 5, 6\}$ are $(1, 3)$, $(2, 3)$, $(3, 4)$, $(4, 5)$ and $(4, 6)$. Therefore, $f(\{1, 2, 4\})$ will equal 5.

The proof that the cut function is submodular can be found in the same book [5] as Proposition 1.2.2. We will also look into a visual proof.

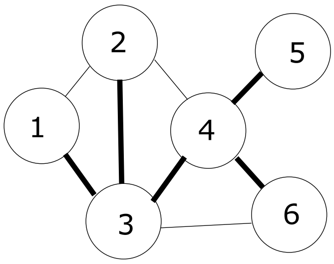


FIGURE 2: example graph with a cut (bold lines)

Example 3.3. Take the graph $G = (V, E)$ as in Figure 1. Then we can take any arbitrary $X, Y \subseteq V$. For this example, we will take $X = \{2, 4\}$ and $Y = \{3, 4\}$. We could draw a Venn diagram as in Figure 3. In this Venn diagram, we have four different sets, namely

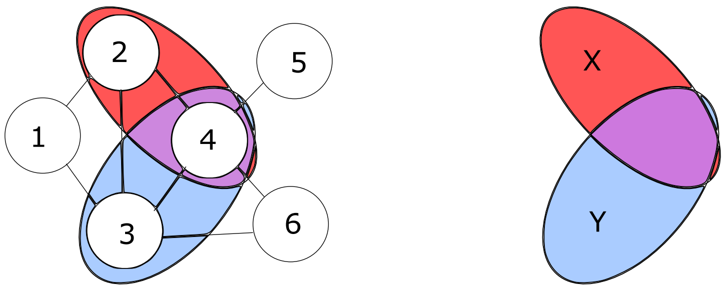


FIGURE 3: example diagram with Venn diagram

$X \setminus Y$, $Y \setminus X$, $X \cap Y$ and $\overline{X \cup Y}$. We can characterize edges as one of the following categories.

1. $X \setminus Y \leftrightarrow Y \setminus X$
2. $X \setminus Y \leftrightarrow X \cap Y$
3. $X \setminus Y \leftrightarrow \overline{X \cup Y}$
4. $Y \setminus X \leftrightarrow X \cap Y$
5. $Y \setminus X \leftrightarrow \overline{X \cup Y}$
6. $X \cap Y \leftrightarrow \overline{X \cup Y}$

All those edges are depicted in Figure 4 with the corresponding number. Now we can look

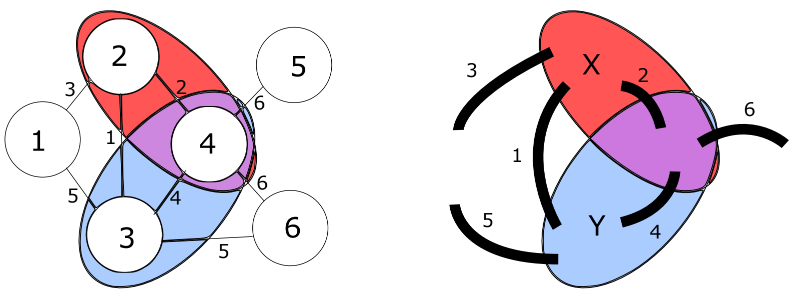


FIGURE 4: example diagram with Venn diagram and type edge

at what type of edges the following function is counting. For example, $f(X)$ will count the edges of type 1, 3, 4 and 6. $f(Y)$ will count the edges of type 1, 2, 5 and 6. So for this example $f(X) = f(\{2,4\}) = 5$ and $f(Y) = f(\{3,4\}) = 6$. Note that 1 and 6 are counted at both terms, whereas all the other types are counted once. We can do the same with $f(X \cup Y)$ and $f(X \cap Y)$. The first term will count the edges of types 3, 5, and 6. The second will count the edges of types 2, 4, and 6. Note that in both terms all edges are counted once, except for edges of type 6, which are counted twice, and one, which are not counted at all. So we have $f(X \cup Y) = 5$ and $f(X \cap Y) = 4$. If you evaluate $f(X) + f(Y)$ and $f(X \cup Y) + f(X \cap Y)$, the values are 11 and 9 respectively. The difference is caused by not counting the edges of type 1 in the second term while counting them twice in the first. This is why the first one is always larger than or equal to the second; therefore, it is submodular. It is equal when no edges of type 1 exist.

From this example, we can see why the cut function is always submodular. The submodularity is useful since, according to [8] submodular functions are easy to minimize, and the minimum cut-function is related to the maximum flow of a graph.[1]

3.2 Coverage function

Now we will look at the coverage function as a submodular function. This function is inspired by γ_m from proposition 1.2.7 in the book by András Frank [5]. This function is used in many applications such as [13] and there is a lot of known about the coverage functions. In this sections, we will start with defining the coverage function. After that, we will look at an example of the coverage function. Finally we will look at the proof of why the coverage function is submodular.

For this purpose, we will take a bipartite graph $G = (V_0, V_1, E)$. Then we will define the neighborhood function.

Definition 3.4 (neighborhood function)**.** Given a graph, $G = (V, E)$ the neighborhood function $N : 2^V \to 2^V$ is defined as follows:

$$N(A) = \{v \in \overline{A} | \exists (v_0, v_1) \in E[v_0 = v \wedge v_1 \in A]\}$$

So the neighborhood function returns the vertices, that have an edge that connects them with a vertex from the given set. Furthermore, this definition can be extended to a bipartite graph, where $V = V_0 \cup V_1$. Next, we will use the neighborhood function to define the coverage function.

Definition 3.5 (coverage function)**.** Given a bipartite graph $G = (V_0, V_1, E)$, the coverage function $f : 2^{V_0} \to \mathbb{Z}$ is defined as follows:

$$f(A) = N\left(\bigcup_{a \in A} a\right)$$

We will look at the proof from the same book [5] of why the coverage function is submodular.

Theorem 3.6. *Given a bipartite graph $G = (V_0, V_1, E)$, the coverage function f is submodular.*

Proof. Since $N(X) \cap N(Y) \supseteq N(X \cap Y)$ and $N(X) \cup N(Y) = N(X \cup Y)$, the non-negativity of the cardinality function implies at once the submodular inequality. $\square$

[1] If you are interested in this, consult [6]

Now, we will look at an example, as in Section 3.1.

Example 3.7. Take the bipartite graph $G = (V_0, V_1, E)$ as in Figure 5, where $V_0 = \{A, B, C\}$ and $V_1 = \{1, 2, 3, 4, 5, 6, 7\}$. Then let f be the coverage function. In V_1, we have

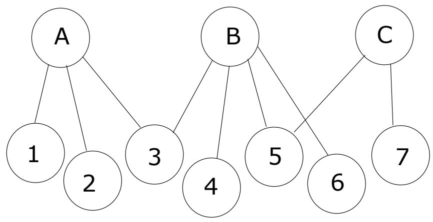


FIGURE 5: example bipartite graph

seven possible types.

1. $A \cap B \cap C$
2. $\overline{A} \cap B \cap C$
3. $A \cap \overline{B} \cap C$
4. $\overline{A} \cap \overline{B} \cap C$
5. $A \cap B \cap \overline{C}$
6. $\overline{A} \cap B \cap \overline{C}$
7. $A \cap \overline{B} \cap \overline{C}$

In Figure 5 we do not have vertices of the types 1 and 3. We will consider those when looking at the proof of why the coverage function is submodular. Let $X = \{A, B\}$ and $Y = \{B, C\}$. If we take $N(X)$, it includes vertices of the types 1, 2, 3, 5, 6, and 7. Essentially all types include A or B. The same can be observed for $N(Y)$. Then we would get the types 1, 2, 3, 4, 5, and 6. Then, if we take the union of both, we will get all types. Which is also the same as $N(X) \cup N(Y)$. Therefore, the following must be true: $N(X) \cup N(Y) = N(X \cup Y)$. If we look at $N(X) \cap N(Y)$, we will get the types 1, 2, 3, 4, and 5, which are significantly less than taking the union. However, it must at least include the types of edges that are in $X \cap Y$ and which has a neighbor to an element in X and Y. So $N(X) \cap N(Y)$ can be larger than $N(X \cap Y)$. All elements in the second term are included in the first. Therefore, the following is true $N(X) \supseteq N(X \cap Y)$. This means that because of cardinality, the following two statements are true:

$$|N(X) \cup N(Y)| = |N(X \cup Y)| = f(X \cup Y) \tag{4}$$

$$|N(X) \cap N(Y)| \geq |N(X \cap Y)| = f(X \cap Y) \tag{5}$$

Since $|N(X)| + |N(Y)|$ counts all elements of $N(X)$ and $N(Y)$ and counts the elements of the intersection twice, the following is true:

$$f(X)+f(Y) = |N(X)|+|N(Y)| = |N(X)\cup N(Y)|+|N(X)\cap N(Y)| \geq f(X\cup Y)+f(X\cap Y)$$

Therefore, the coverage function is submodular.

There are many applications of optimization problems with submodular functions. One such example would be effectively placing a camera that covers as much area as possible. [13]

3.3 Largest degree function

In this section, we will look at the largest value function. This is a special case of the maximum function. However, it is interesting to look at this function since it shows how a function can be converted to a coverage problem. We will use this to show how the

S	$f(S)$
$\emptyset$	0
$\{5\}$	1
$\{1,2,3\}$	3
$\{1,5,6\}$	2

S	$f(S)$
$\{4\}$	4
$\{3\}$	3
$\{3,4\}$	4
V	4

TABLE 1: Sample values of the maximum degree function of the graph in Figure 1

interpretation of the M-convex set of the largest degree function can be found if we know how to convert it to the coverage function. First, we will define the largest degree function. After, we will look at how for each largest degree function, a graph exists on which the coverage function gives the same values. Let us define the degree first.

Definition 3.8 (degree function)**.** Given a graph $G = (V, E)$. The degree function $d : V \to \mathbb{Z}$ is defined as follows:

$$d(v) = |N(\{v\})|$$

Now we do not have a set function. To obtain one, we need the following definition.

Definition 3.9 (the largest degree function)**.** Given a graph $G = (V, E)$ and the degree function d as defined in Definition 3.8. Then the largest degree function $f : 2^V \to \mathbb{Z}$ is defined as follows:

$$f(S) = \max_{v \in S} d(v)$$

Furthermore define $f(\emptyset) = 0$

Now we can take an example and look at it.

Example 3.10. Take the graph $G = (V, E)$ as in Figure 1 and let f be the largest degree function. If we look at a few values, we obtain Table 1. We will not prove this is submodular directly. Instead, we will show that this graph with the largest degree function can be converted to a bipartite graph with a coverage function that gives the same values. We will do it with this example. We construct the bipartite graph as in Figure 6. This figure is clustered with edges. However, we can see that A is connected to every vertex in $\{1, \ldots, 6\}$. Moreover, D is only connected to 4. Furthermore, 4 is connected to all vertices in $\{A, B, C, D\}$. This means that it will output 4 vertices, equal to the degree of 4 if we have the neighbors function on a set that includes 4. This works for any combination of vertices in $\{1,\ldots,6\}$.

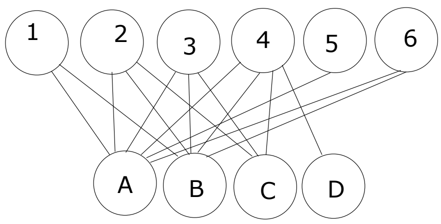


FIGURE 6: example bipartite graph for largest degree function

We will prove that for every graph, there exists a bipartite graph where the coverage function will output the same. For this, we will first prove a lemma.

Lemma 3.11. *Given the graph $G = (V_0, V_1, E)$ as defined in (7), $x, y \in V_0$ and $d(x) \leq d(y)$. The following holds:*

$$N(\{x\}) \subseteq N(\{y\}) \tag{6}$$

Where $N : 2^{V_0} \to 2^{V_1}$ is defined as in Definition 3.4.

Proof. Let $G = (V_0, V_1, E)$ be a bipartite graph. Fix $x, y \in V_0$ with $d(x) \leq d(y)$. Take $n \in \mathbb{N}^+$ such that $d_n \in N(x)$. Then we know the following:

$(x, d_n) \in E$ (By of N)
$d(x) \geq n$ (By definition of G and therefore E.)
$d(y) \geq n$ (Since $d(x) \leq d(y)$)
$(y, d_n) \in E$ (By definition of E)
$d_n \in N(\{y\})$ (By of N)

Therefore, (6) is true. □

We will use the lemma in the following theorem.

Theorem 3.12. *For every largest degree function $f : V \to \mathbb{Z}$ as defined in Definition 3.9. Then there exists a bipartite graph $G = (V_0, V_1, E)$ such that the neighbor function as defined in Definition 3.4 is equal to f. Furthermore, G can be defined as follows:*

$$\begin{aligned} &V_0 = V \text{ (These are vertices are from the domain of } f) \\ &V_1 = \{d_1, \ldots, d_{f(V)}\} \text{ (Those vertices are added to construct } G) \\ &E = \{(v_0, d_n) | v_0 \in V_0, n \in \mathbb{N}^+, n \leq d(v_0)\} \end{aligned} \tag{7}$$

Proof. Let $G = (V_0, V_1, E)$ be a bipartite graph defined as in (7). Then we have to prove that $f(S) = |N(S)|$ for all $S \subseteq V$. Take $S \subseteq V$. Then we have the following from the definition of the largest degree function.

$$f(S) = \max_{v \in S} d(v)$$

Take $s \in S$ such that $f(S) = d(s)$. Then we obtain.

$$\max_{v \in S} d(v) = d(s)$$

Now we want to prove that $d(s) = |N(S)|$ First, we take an arbitrary $v \in S$. Since $d(v) \leq d(s)$, by Lemma 3.11 the following is true:

$$N(\{v\}) \subseteq N(\{s\})$$

Since this is true for all $v \in S$ and by definition of d and N, we know the following is true:

$$d(s) = |N(\{s\})| = |N(S)|$$

So now we have proven that $f(S) = |N(S)|$ □

For every largest degree function, a bipartite graph exists such that the neighborhood function gives the same output. Moreover, we know that the neighborhood function on a bipartite graph is the same as a coverage function, so the largest degree function must be submodular.

This function can be generalized to $f : 2^E \to \mathbb{N}$ where, for any arbitrary mapping $g : E \to \mathbb{N}$, f is defined as follows:

$$f(A) = \max_{a \in A} g(a)$$

This is not necessarily the most interesting example of a submodular function because it is a special case of the maximum function which is submodular. Nevertheless, it shows that the coverage function pops up in other submodular functions.

3.4 Influence function

In this section, we will cover the influence function. This model is inspired by real-life applications such as studying social networks [7, 11]. In this section, this paper will be referenced. First, we will describe the independent cascade model from a paper by David Kempe and Jon Kleinberg, and Éva Tardos [7]. Then, we will look at an example of this model. Finally, we will examine the proof of submodularity of the influence function.

Let us define the influence function. We will take the independent cascade model from the work of David Kempe and Jon Kleinberg, and Éva Tardos [7]. For this, we will need a graph $G = (V, E)$ with directed edges. Each vertex is either active or inactive and can become active if a neighbor activates it. In each discrete time step, when a vertex becomes active or is active in the initialization, it will try to activate each member with a probability of $p \in [0, 1]^E$. If the activation is successful, that neighbor vertex will be active in the next step. Thus we will have a sequence of sets A_n, where in each time step i, A_i represents the active vertices at that moment. A_0 represents the set of vertices that are active at the initialization. The independent cascade model will run until a particular time t.

Definition 3.13 (influence function)**.** Given a graph $G = (V, E)$ and the independent cascade model. Then the influence function $\sigma : 2^V \to \mathbb{Z}$ is defined as follows:

$$\sigma(A) = |A_t|$$

Where A_i is the set of all activated vertices at time step i where $A_0 := A$.

To illustrate this function, we will look at an example.

Example 3.14. In Figure 7 we can see the graph. Let us call it G and let the vertices be V and edges E.

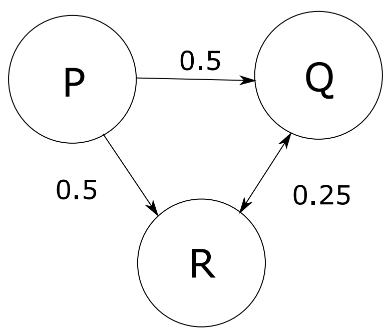


FIGURE 7: example graph for relation function

In Figure 7 we can see the probabilities on the edges. Let us denote them as p_e. Now, we will look at the value of σ. Take $\sigma(\{R\})$, in the first step. There will be a chance of activating Q with probability 1. Since nothing happens after, we can calculate the expectation after step 1. That would be the following:

$$\sigma(\{R\}) = 1 \cdot P(|A_1| = 1) + 2 \cdot P(|A_1| = 2) = 1 \cdot 0.75 + 2 \cdot 0.25 = 1.25$$

It would be the same for $\sigma(\{Q\})$. The interesting term would be $\sigma(\{P\})$. In this case, it will try to activate both Q and R in the first step. If it succeeds, then one of the neighbors will try to activate the other vertex. If both succeed, then no activation happens after time step 1. In this way we obtain the following equation:

$$\begin{aligned}\sigma(\{P\}) &= 1 \cdot P(|A_2| = 1) + 2 \cdot P(|A_2| = 2) + 3 \cdot P(|A_2| = 3)\\ &= 1 \cdot 0.25 + 2 \cdot 0.5 \cdot 0.75 + 3 \cdot (0.25 + 0.5 \cdot 0.25)\\ &= 2.125\end{aligned}$$

Now, we can also calculate $\sigma(\{P, Q, R\})$. However, this term is trivial since they all are activated, so it will be $\sigma(\{P, Q, R\}) = 3$. The same line of reasoning works for $\sigma(\{Q, R\}) = 2$, since P can never be activated. Next, by symmetry $\sigma(\{P, Q\}) = \sigma(\{P, R\})$, we will calculate its value.

$$\sigma(\{P, R\}) = 2 \cdot P(|A_1| = 2) + 3 \cdot P(|A_1| = 3) = 2 \cdot 0.5 \cdot 0.75 + 3 \cdot (1 - 0.5 \cdot 0.75) = 2.625$$

The paper by David Kempe, Jon Kleinberg, and Éva Tardos [7] shows that the maximum coverage function is a special case of the influence maximization problem. This means that the coverage function is connected to the influence function. However, this proves that the influence maximization problem is NP-hard like the maximum coverage problem. To be more specific, approximating to at least a factor of $1 - e$ is easy. However, if a better approximation is required, the problem becomes NP-hard. In the paper, the triggering model is used to prove the influence function's submodularity, defined as follows:

Definition 3.15 (triggering model)**.** Each node v independently chooses a random "triggering set" T_v according to some distribution over subsets of its incoming neighbors. Initially, a set A is activated. Subsequently, an inactive node v becomes active in step t if it has a neighbor in its chosen triggering set T_v that is active in step $t - 1$. [7]

It is easy to prove that the influence function in this model is submodular. By using (2), we can show that the effect of adding a vertex to a smaller subset is larger or equal to adding it to a larger set, with a fixed triggering set T. From this, we can show that the independent cascade model can be converted to a special case of the triggering model. We will do it with the previous example.

Example 3.16. Take model from Example 3.14. We must determine the probability distribution X of possible triggering sets T. We can construct the triggering set as follows: for all directed edges $(v_1, v_2) \in E$ we add v_1 with the probability $p_{(v_1,v_2)}$ to T_{v_2}. So for example we obtain $T = (\emptyset, \{P\}, \{P, Q\})$ of the form (T_P, T_Q, T_R) with a probability of $0.5 \cdot 0.5 \cdot 0.75 \cdot 0.25 = 0.046875$.

We have shown in Example 3.16 that this case of the independent cascade model can be transformed into a triggering model. Since it is easy to prove that the influence function in the triggering model is submodular, the influence function in the independent cascade model must also be submodular. The triggering model shows that the influence function in the independent cascade model contains multiple coverage functions in disguise since if the triggering set is fixed, it becomes a deterministic coverage function. Fixed triggering set is about what vertices are reachable from the initial active set in the digraph.

3.5 Minimum cost connection function

In this subsection, we will briefly go over another submodular function. Like the previous one, it is hard to calculate its value. The function is called the minimum cost connection function. Given a graph $G = (E, V)$ and a weight, $w : E \to \mathbb{Z}$ we have a fixed subset of vertices $S \subset E$. Then the minimum cost function is the minimal sum of the weight of the edges that connect the vertices in A.

Definition 3.17. *P is a path from vertex $a \in V$ to $b \in V$ if and only if P corresponds to a sequence of hops between vertices that starts in a and ends in b. Moreover, no vertex is used twice.* [12]

Now we will define connectivity. This concept is also defined in chapter 6.2 of the same book [12] as follows:

Definition 3.18. *A graph G is connected if there is a path between each pair of nodes.*

Now we can define the minimum cost connection function.

Definition 3.19 (minimum cost connection function)**.** *given a graph $G = (V, E)$ that is connected and a weight function $w : E \to \mathbb{Z}$. $f : 2^V \to \mathbb{Z}$ is a minimum cost connection function if and only if the following is true for any subset $A \subseteq V$.*

Let $G(A)$ denote the subset of all subgraphs that contain A and are connected, and E' is the edges of a graph G'

$$f(A) = \min_{G' \in G(A)} \sum_{e \in E'} w(e)$$

We will not be proving that this function is submodular. Instead, we will look at an example and then see why it is hard to the actual value of the function and it must be submodular.

Example 3.20. Given the graph $G = (V, E)$ with the weight $w : E \to \mathbb{Z}$ as in Figure 8. Now we can calculate some values of the minimum cost connection function f. You can find a few of them on the table. For example, $f(\{A, D\})$ is just a simple shortest path problem. So we can find $f(\{A, D\}) = 3$. $f(\{A, B, C, D, E\})$ is a minimum spanning tree problem. Since you need to include all nodes, you should be able to connect all nodes with $N - 1$ edges, where N is the number of vertices. So in the case of $f(\{A, B, C, D, E\}) = 5$, we can see that the evaluation of the function will be easy if we put in all vertices. However, it will be harder if you don't have only two vertices or all vertices. This problem is known as the Steiner tree problem, described in chapter 6.5 of a book by Piet Mieghem [12]. This problem is NP-hard. So like the influence function, we should use an approximation function.

With this example we will reason why it should be submodular using (2). Here you will have to take a set P and a proper subset $Q \subset P$, then adding that element to the subset Q will let the function increase less than adding to the entire set P. So let us take $P = \{A, B, C\}$ and $Q = \{A, B\}$, and let's add the vertex E. Then the difference of P and $P \cup \{E\}$ is 1 and for Q and $Q \cup \{E\}$ is 3, thus it is higher for the subset. This will always be the case, which makes intuitive sense since the smallest subgraph that connects P has all edges in the smallest subgraph that connects Q. That means adding edges to find a subgraph that connects with E as well. The subgraph will add edges that may be already in the smallest subgraph of P and never less because if it would, then the smallest subgraph of P is not the smallest possible anymore. This way of choosing the subgraph

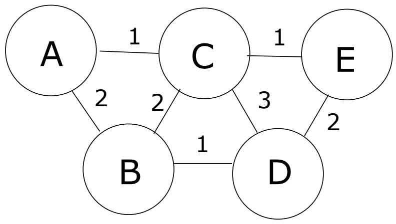


FIGURE 8: example of a graph with weights on the edges

works for every graph, so the function must be submodular. However, writing down the proof is not within scope since we will not use this finding later.

In the example, we have looked at a concrete case where the submodular function is hard to evaluate, like the influence function. This is however a fascinating function since, if you want to know for example how to connect a computer network effectively, then you are interested in how you can connect as many computers as possible at the lowest cost.

4 M-convex sets

Now we have seen examples of submodular functions. We will focus on the corresponding M-convex sets. The aim of this paper to find an alternative interpretation of the lattice points of M-convex sets. For this, we will solely focus on integer-valued submodular functions. We will first examine the geometric definition of all M-convex sets. Afterwards, we will investigate the alternative definition of M-convex sets of the cut, coverage, and largest degree function. Let us define the M-convex set.

Definition 4.1. Given a submodular function $f : 2^E \to \mathbb{Z}$. The M-convex set of f is defined as follows:

$$M = \left\{ x \in \mathbb{Z}^E \middle| \forall A \in 2^E \left[\sum_{a \in A} x_a \leq f(A) \right], \sum_{e \in E} x_e = f(E) \right\} \tag{8}$$

It is not easy to interpret the M-convex set from the definition alone, so we will explain this using an example.

Example 4.2. For this example, we will use the cut function of the following graph.

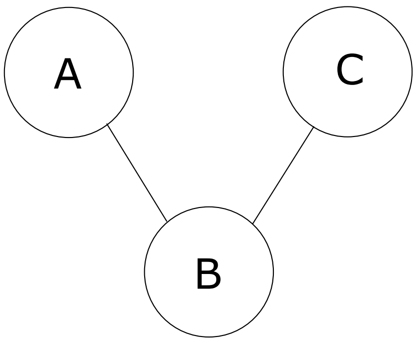


FIGURE 9: example graph

S	$f(S)$	S	$f(S)$
$\{\}$	0	$\{A,B\}$	1
$\{A\}$	1	$\{A,C\}$	2
$\{B\}$	2	$\{B,C\}$	1
$\{C\}$	1	$\{A,B,C\}$	0

TABLE 2: Values of the cut function of the graph in Figure 9

From this, we can construct the M-convex set. For this purpose, it helps to calculate all values of the cut function. In Table 2 you can see the values of all possible sets.

We have a system of equations for which the points in the M-convex sets must satisfy, which can be found below.

$$x_A \leq 1 = f(\{A\})$$

$$x_B \leq 2 = f(\{B\})$$

$$x_C \leq 1 = f(\{C\})$$

$$x_A + x_B \leq 1 = f(\{A,B\})$$

$$x_A + x_C \leq 2 = f(\{A,C\})$$

$$x_B + x_C \leq 1 = f(\{B,C\})$$

$$x_A + x_B + x_C = 0 = f(\{A,B,C\})$$

Some may recognize this as a linear system of equations. One way to interpret this is as a convex set enclosed by hyperplanes. To illustrate this we will fix, $x_C = 0$ then we obtain Figure 10.

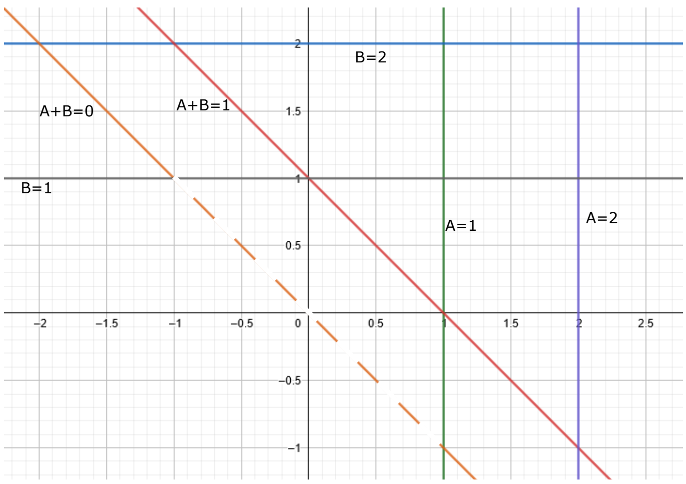


FIGURE 10: visualization of the M-convex inequalities

In Figure 10 we can see that the lines of the other inequalities enclose the line of the equality $x_A + x_B + x_C = 0$. Since $x_C = 0$, we can draw it on a two-dimensional plane. If we want to do this at all, x_C we need to draw it in a three-dimensional plane. However, the idea stays the same. The M-convex sets contain all lattice points on the line $A+B=0$ and are below or left of the other lines. Since then, the point only has integer coordinates and satisfies the associated inequalities. That includes the points $(-1,1,0)$, $(0,0,0)$ and $(1,-1,0)$ where they are of the form (x_A, x_B, x_C). As you can see in Table 3, these are indeed all the points where $x_C = 0$.

Table 3 can be constructed, by trying all possibilities. However, this section will look at different ways to interpret this set. We will call the interpretation as in Figure 10 the

x_A	x_B	x_C
1	0	−1
1	−1	0
1	−2	1
0	1	−1
0	0	0

x_A	x_B	x_C
0	−1	1
−1	2	−1
−1	1	0
−1	0	1

Table 3: All lattice points in M-convex set

geometric interpretation. We will go over the different submodular functions from the previous section to see in which different ways the lattice points in the M-convex set can be interpreted.

4.1 Cut function

For the cut function, we will take the graph in Figure 9. From this, we can construct the M-convex set. It will be as in Table 3. First, we will look at the alternative way to interpret the M-convex set. Then we will see why it works. For this, we need to understand the different ways to orient the graph. By this, we ask the question: In what ways can we direct the edges? In Figure 11 we see an example of this.

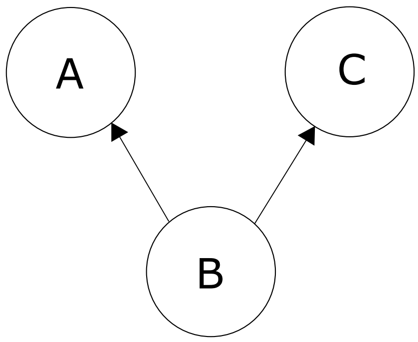


Figure 11: example graph with orientation

Now that all the edges have a direction, we can calculate the in-degree and out-degree of each vertex v, denoted as δ_v and μ_v respectively. If we subtract the out-degree from the in-degree, we obtain 1,−2, and −1 for A, B, and C, respectively. This result also corresponds with a lattice point in the M-convex set. For every orientation, we get a lattice point in the M-convex set. These orientations also include the orientation where some edges are still undirected. The undirected edges mean that the in and out degree does not change. Furthermore, the converse is also true. For every lattice point in the M-convex set, an orientation exists such that the in-degree minus the out-degree is equal to the value of the integer that belongs to the corresponding vector index. In this paper, we will prove this statement. The converse is also true but harder to prove.

For this, we will need to look at the two rules of the M-convex set. The first is the following equation:

$$x_A + x_B + x_C = 0$$

We will prove in a lemma that this property holds

Lemma 4.3. *Given a graph $G = (V, E)$ with an orientation. Then construct a lattice point as follows: For all $v \in V$,*

$$x_v := \delta_v - \mu_v$$

Then the following holds:

$$\sum_{v \in V} x_v = 0$$

Proof. Let x be given such that $x_v = \delta_v - \mu_v$ for all $v \in V$. Then we can rewrite the following:

$$\sum_{v \in V} x_v = \sum_{v \in V} (\delta_v - \mu_v) = \sum_{v \in V} \delta_v - \sum_{v \in V} \mu_v$$

Since we know that in any orientation, the sum of the out-degree must be equal to the sum of the in degree, RHS is equal to zero. So $\sum_{v \in V} x_v = 0$ holds. □

We will now prove the following lemma. This proof is a little more complicated, and we will prove it by contradiction.

Lemma 4.4. *Given a graph $G = (V, E)$ with an orientation and a cut function f. Then construct a lattice point as follows: For all $v \in V$,*

$$x_v := \delta_v - \mu_v$$

Then the following holds for all $S \subset V$:

$$\sum_{v \in S} x_v \leq f(S) \tag{9}$$

Proof. Let an orientation, cut function f of an arbitrary set $S \subset V$ be given. Since we know that $x_v = \delta_v - \mu_v$ holds for all $v \in V$, we have the following.

$$\sum_{v \in S} x_v = \sum_{v \in S} (\delta_v + \mu_v)$$

Since all directed edges between vertices in S cancel each other in $\sum_{v \in S} (\delta_v + \mu_v)$, we only have the edges that go in and out S. Therefore the following equation holds.

$$\sum_{v \in S} (\delta_v + \mu_v) = \delta_S - \mu_S$$

Where δ_S and μ_S are the amount of edges that go in or out S, respectively. We know that the sum of all edges is $f(S)$, so the sum of the in and out edges cannot be larger. Therefore, the following is true:

$$\delta_S + \mu_S \leq f(S)$$

We can rewrite the following since μ_S is non-negative,

$$\delta_S - \mu_S = \delta_S + \mu_S - 2\mu_S \leq f(S) - 2\mu_S \leq f(S)$$

So we obtained (9). □

Now we have proven that for each orientation, a lattice point exists that satisfies the linear inequalities.

Theorem 4.5. *Given an orientation and a cut function f. A lattice point exists in the M-convex set of f such that the following holds for all $v \in V$:*

$$x_v = \delta_v - \mu_v \tag{10}$$

Proof. Let x be given by (10) then we need to prove that it satisfies the two properties of M-convex sets. Which is done by Lemma 4.3 and Lemma 4.4. □

Now that we have proven that for every orientation, we can construct a lattice point that is part of the M-convex set of the cut function. This is insightful since we now can easily construct all lattice points by hand intuitively.

4.2 Coverage function

For the interpretation of the M-convex set of a coverage function, we will first look at an example. Then we will prove that there is a corresponding lattice point for every distribution. Finally, we will prove that we can construct a distribution for every lattice point. We will do this by using an algorithm and proving its correctness.

Example 4.6. We will examine the coverage function, thus we have a graph $G = (V_0, V_1, E)$ as in Figure 12, where $V_0 = \{A, B, C\}$ and $V_1 = \{P, Q, R, S\}$.

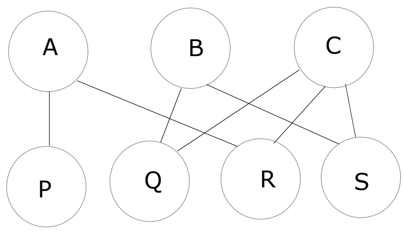


FIGURE 12: example bipartite graph for coverage function

If we take $E = \{A, B, C\}$, we can calculate all lattice points that satisfy the following equations.

$$\begin{aligned}
x_A &\leq 2 = f(\{A\}) \\
x_B &\leq 2 = f(\{B\}) \\
x_C &\leq 3 = f(\{C\}) \\
x_A + x_B &\leq 4 = f(\{A, B\}) \\
x_A + x_C &\leq 4 = f(\{A, C\}) \\
x_B + x_C &\leq 3 = f(\{B, C\}) \\
x_A + x_B + x_C &= 4 = f(\{A, B, C\})
\end{aligned}$$

All lattice points can be found in Table 4. These points can be found by the geometric interpretation. However, if we look at the equations they have to satisfy, we might notice that we distribute the vertices P, Q, R, and S to a neighbor vertex and then count how many vertices each vertex has. For example, if we pair P with A, Q and R with C and S with B then we have $x_A = |\{P\}| = 1$, $x_B = |\{S\}| = 1$ and $x_C = |\{Q, R\}| = 2$, which is indeed a lattice point. Let us formulate it more precisely.

Definition 4.7 (distribution of vertices)**.** Given a bipartite graph $G = (V_0, V_1, E)$. A distribution D of V_1 over V_0 satisfies the following conditions.

1. $D \in (2^{V_1})^{V_0}$ (D is a vector with sets of vertices as values)

x_A	x_B	x_C
2	2	0
2	1	1
2	0	2

x_A	x_B	x_C
1	2	1
1	1	2
1	0	3

TABLE 4: Values of the cut function of the graph in Figure 9

2. $\forall x, y \in V_0[x \neq y \implies D_x \cap D_y = \emptyset]$ (All subsets are disjoint)
3. $\bigcup_{v \in V_0} D_v = V_1$ (All subsets cover everything)
4. $\forall v \in V_0 \ \forall n \in D_v[(v, n) \in E]$ (Vertices are assigned to neighbors)

Note that this is not exactly a partition since it might contain empty sets.

Before we formulate the theorem, we will prove a lemma.

Lemma 4.8. *Given a bipartite graph $G = (V_0, V_1, E)$ and a coverage function $f : 2^{V_0} \to \mathbb{Z}$ as defined in Definition 3.5. Let D be a distribution of V_1 over V_0 and let $N : 2^{V_0} \to 2^{V_1}$ be a function that maps a subset of V_0 to a set of vertices that are neighbors of at least one of the vertices in V_0. Then the following holds:*

$$\forall V \subseteq V_0 \left[\left| \bigcup_{v \in V} D_v \right| \leq |N(V)| \right] \tag{11}$$

Proof. Take $V \subseteq V_0$. Then let $x \in \bigcup_{v \in V} D_v$. There exists a $v \in V$ such that $x \in D_v$. Since $V \subseteq V_0$ so is $v \in V_0$. Because of the fourth condition of distribution D, $(v, x) \in E$ is true. So that means x neighbors at least one vertex, namely $v \in V_0$. Since $x \in N(V)$ the following is true:

$$\bigcup_{v \in V} D_v \subseteq N(V)$$

From the definition of cardinality, (11) is proven. □

Now let us formulate the theorem and prove it.

Theorem 4.9. *Given a bipartite graph $G = (V_0, V_1, E)$ and a coverage function $f : 2^{V_0} \to \mathbb{Z}$ as defined in Definition 3.5 and let S be an M-convex set as defined in Definition 4.1. Then for every distribution D of V_1 over V_0, there is a $x \in S$ such that the following is true:*

$$\forall v \in V_0[|D_v| = x_v] \tag{12}$$

Proof. Let D be a distribution that satisfy the conditions in Definition 4.7.

Define x such that (12) holds.

Now we have to prove that $x \in S$. First, proof that $\sum_{v \in V_0} x_v = f(V_0)$. From the definition, we know the following:

$$\sum_{v \in V_0} x_v = \sum_{v \in V_0} |D_v|$$

Since D is a distribution, all sets D_i are disjoint and cover V_1.

$$\sum_{v \in V_0} |D_v| = |V_1|$$

Since all vertices in V_1 have a neighbor in V_0, the following statement holds:

$$|V_1| = f(V_0)$$

Therefore, the first condition of an M-convex set is proven. Now we have to prove the following:

$$\forall V \subset V_0 \left[\sum_{v \in V} x_v \leq f(V) \right]$$

Take $V \subset V_0$. We obtain the following from the definition:

$$\sum_{v \in V} x_v = \sum_{v \in V} |D_v|$$

Since D_i is disjoint for $i \in V \subset V_0$. We have the following:

$$\sum_{v \in V} |D_v| = \left| \bigcup_{v \in V} D_v \right|$$

By Lemma 4.8 we know the following is true:

$$\left| \bigcup_{v \in V} D_v \right| \leq |N(V)|$$

Where N is the neighborhood function from Definition 3.4. By definition, the following holds:

$$|N(V)| = f(V)$$

We have proven the second condition. Therefore there exists a $x \in S$ such that (12) holds. □

We have proven that given a distribution, we have a lattice point in the M-convex set corresponding to the coverage function. The converse is also true, but it is harder to prove. It helps to look first at why it makes sense that it would be true.

Example 4.10. If we take $G = (V_0, V_1, E)$ such as in Figure 12 then we can choose a lattice point. For example $x = (2, 1, 1)$ of the form (x_A, x_B, x_C). Then we can make a distribution D such that (12) holds. First, note that P must belong to A since it cannot be assigned to another vertex from V_0. Furthermore, since $x_A = 2$ and A has two neighbors, we also have to assign R to A. So we have $D_A = \{P, R\}$. We are left with B, C, Q and S. We can see that P and S are part of this subgraph, and it does not matter how we will assign those to B and C. Thus we found there are two possible distributions. Namely, $(\{P, R\}, \{S\}, \{Q\})$ and $(\{P, R\}, \{Q\}, \{S\})$ of the form (D_A, D_B, D_C).

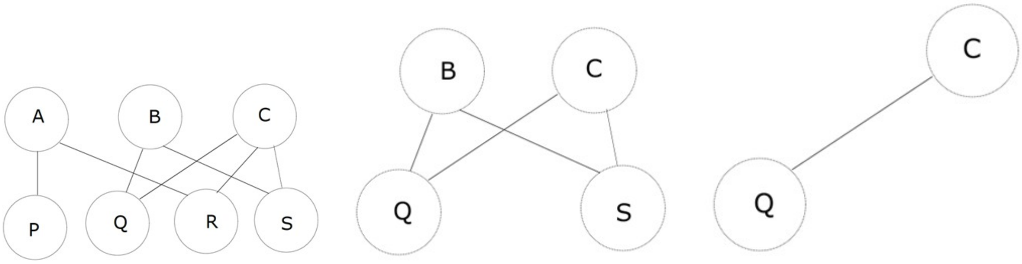
It is harder to prove that, for every $x \in S$ a distribution exists D such that it satisfies (12). We can prove this by proving the Algorithm 1 gives a distribution given a lattice point.

Algorithm 1 An algorithm for constructing a distribution given a lattice point

Require: x is the given lattice point of the M-convex set
G is the given bipartite graph
V_0 and V_1 are the bisets of G

$D \leftarrow (\{\}, \{\}, \ldots, \{\})$ (This is a tuple of empty sets paired with the vertices in V_0)
while $\sum_{v \in V_0} x_v > 0$ **do**
  if $\exists v \in V_0, x_v = |N(\{v\})|$ **then**
    take an $v \in V_0$ such that $x_v = |N(\{v\})|$
    for all $n \in N(\{v\})$ **do**
      $D_v \leftarrow D_v \cup \{n\}$
      remove n and the incident edges from G.
    end for
    $x_v \leftarrow 0$
    remove v and the incident edges from G.
  else
    take an arbitrary $v \in V_0$
    take an arbitrary $n \in N(\{v\})$
    $D_v \leftarrow D_v \cup \{n\}$
    $x_v \leftarrow x_v - 1$
    remove n and the incident edges from G.
    if $x_v = 0$ **then**
      remove v and the incident edges from G.
    end if
  end if
end while
return D

Before proving the correctness of the algorithm we will examine an example of how the algorithm works.



(A) first iteration (B) second iteration (C) third iteration

FIGURE 13: example execution of Algorithm 1

Example 4.11. Take a bipartite graph $G = (V_0, V_1, E)$ as in Figure 13a. Where $V_0 = \{A, B, C\}$ and $V_1 = \{P, Q, R, S\}$. Take $x = (2, 1, 1)$ of the form (x_A, x_B, x_C). First, we initialize $D = (\emptyset, \emptyset, \emptyset)$ of the form (D_A, D_B, D_C). In the first iteration of the for loop the if statement is true since $|N(A)| = x_A$. So take $v = A$. Then according to the nested for loop we iterate through all neighbors. We add the neighbors to the empty D and remove them from the graph. After the nested for loop we set $x_A = 0$ and remove A from the graph. We then have $D = (\{P, R\}, \emptyset, \emptyset)$, $x = (0, 1, 1)$, and G is as in Figure 13b. In the next iteration the if statement is false and therefore we move to the else part. We take $v = B$ arbitrarily. Then we take one arbitrary neighbor $n = S$. We add S to D_B, decrease x_B by one, and remove both B and S from G. We then have $D = (\{P, R\}, \{S\}, \emptyset)$, $x = (0, 0, 1)$, and G is as in Figure 13c. In the final iteration the if statement is true therefore, all neighbors of C are removed, x_C is set to 0, and both C and Q are removed from the graph. Now we have $D = (\{P, R\}, \{S\}, \{Q\})$, $x = (0, 0, 0)$, and Q is empty. Since $x_A + x_B + x_C = 0$ the algorithm has finished and a distribution corresponding to the vertex is $(\{P, R\}, \{S\}, \{Q\})$

To prove total correctness of Algorithm 1 for each graph, we need to prove partial correctness and termination. We will first prove the termination since this is easier than its correctness.

Lemma 4.12. *Algorithm 1 always terminate.*

Proof. Take a bipartite graph $G = (V_0, V_1, E)$. Since the sum $\sum_{v \in V_0} x_v = |V_1|$ and each iteration, the sum decreases by at least one, so the sum must hit 0. $\square$

Now we need to prove the partial correctness. We will achieve this with two lemmas. Both lemmas will use loop invariants.

Lemma 4.13. *Algorithm 1 will return a distribution.*

Proof. First, we prove property 1. This proof is trivial since D starts as a tuple, and only vertices from V_1 are added to D.

To prove properties 2 and 4, we will use two invariants. The first invariant is that D always upholds properties 2 and 4 from the distribution definition. The second invariant is that all vertices in D are not in V_1. In both cases, neighboring vertices from V_1 are added to D. Therefore the first invariant holds. Afterwards, the vertices are removed from V_1. Now it holds the second invariant as well. Since both invariants hold properties, 2 and 4 are proven.

Now we only need to prove property 3. This property holds since every time the sum $\sum_{v \in V_0} x_v$ decreases by some amount n and since it will hit precisely zero (x_n cannot become negative), there will be n different vertices added to D. So it must cover V_1. Therefore, property 3 is actual.

Now we have proven all four properties. □

Lemma 4.14. *Algorithm 1 will return a distribution that satisfies (12)*

Proof. Since by Lemma 4.13 it will return a distribution. To prove it satisfies (12), we will use a graph G', which is the induced subgraph containing the vertices removed in the algorithm. Furthermore, we have x', which contains the values subtracted from x in the algorithm. We will use the following invariant:

The following equation is true for all $V' \subseteq V'_0$:

$$\sum_{v' \in V'} x'_{v'} \leq |N(V')| \tag{13}$$

First, the invariant holds before the loop since G' is empty. Now assume the invariant is true at the start of an iteration. In each case, a vertex from V_0 is removed and added to V'_0. Call the vertex v_0. In the same iteration, a subset of n vertex is removed from V_1 and then added to V'_1. Let n be the number of vertices removed this way. Now prove the invariant will hold after the iteration. Assume to the contrary that a $V' \subseteq V'_0$ exists that contains v_0 such that (13) is not true after the iteration. Then $f(V') > |N(V')|$ is true after the iteration, so the following is true.

$$f(V') - n > |N(V')| - n$$

Which is equal to $f(V') > |N(V')|$ before the iteration, because n neighbors were removed from G and added to G'. This statement is a contradiction since it violates the invariant before the iteration. So the invariant must hold.

The invariant holds after the loop. Therefore, G' must contain the original graph G since all vertices are removed and added to G', and x' is equal to the original x. This means that (12) is satisfied. □

Theorem 4.15. *The converse of Theorem 4.9 is also true.*

Proof. Since Algorithm 1 returns a distribution that satisfies (12) by Lemma 4.14 for every lattice point. Therefore, for every lattice point in the M-convex set of the coverage function, a distribution exists such that (12) holds. □

Now we have a proof that every lattice point has a different interpretation, namely the distribution interpretation, rather than the geometric interpretation. As shown before, it is helpful to see that the largest degree function is a coverage problem in disguise. We will show that the alternative interpretation of an M-convex of a largest degree function is similar to the coverage function.

4.3 Largest degree function

In this section, we will look how knowing the largest degree function can be converted to the coverage function, helps us with finding an alternative definition for the lattice points of the M-convex set. We will look at an example how this can be achieved.

Example 4.16. Take graph $G = (V, E)$ as in Figure 14. We can convert this graph into a bipartite graph $G' = (V'_0, V'_1, E')$. In Figure 15 you can see G'.

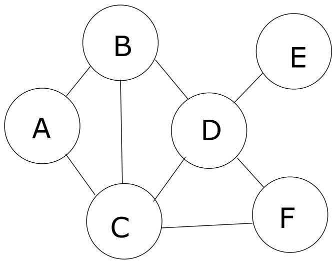


FIGURE 14: example graph

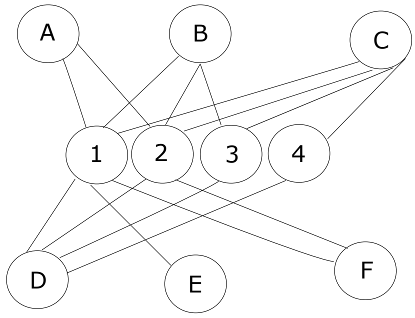


FIGURE 15: bipartite graph from Figure 14

As proven, the alternative interpretation of the m-convex set of a coverage function means that this has the same alternative interpretation of an m-convex set. This means that given a lattice point in the m-convex set, there is a distribution in G'. So take for example the following distribution $D = (\{\}, \{\}, \{2,4\}, \{3\}, \{\}, \{1\})$ in the form $(D_A, D_B, D_C, D_E, D_F)$. Then we have the lattice point $x = (0,0,2,1,0,1)$ of the form, $(x_A, x_B, x_C, x_D, x_E, x_F)$ which is indeed part of the M-convex set.

This example shows that knowing the coverage function is in the largest degree function helps us with finding the alternative definition of the lattice points. This can help us with finding the alternative definition for functions such as the influence function since they have the coverage problem packed in the function.

5 Conclusions

This paper aimed to get examples of submodular functions and investigate the similarities. Moreover, this paper gave alternative interpretations of the M-convex sets corresponding to the cut and coverage function, and it also proved that the alternative interpretation was equivalent to the geometric interpretation.

5.1 submodular function

We have seen many examples of submodular functions. If we look at the proofs of submodularity, we see that coverage is a problem hidden in many of those functions. For example, the maximum function can be converted to a coverage problem. Moreover, for the proof of the influence function, the triggering model is used, and this model is a probability model between different coverage function instances. We also saw the same reachability concept for the minimum cost connection function as in the triggering model.

Moreover, we have seen there are countless applications for submodular functions, such as in economic models, computer science, and mathematical branches, such as machine learning [1], game theory [10], et cetera.

5.2 M-convex sets

In this paper, we have shown the proof that each orientation and distribution can be associated with a lattice point of the M-convex set of the cut function and coverage function, respectively. For the coverage function, we have proven the converse as well. With the algorithm, we can construct a distribution for every lattice point, and the correctness of this algorithm is proven. This means there are alternative definitions for the lattice points of the M-convex sets rather than the geometric definition. These alternative definitions help with understanding the meaning of the lattice points. Moreover, since it can be done for the coverage function, similar methods can be used for different functions, such as the maximum function.

6 Recommendations

This paper covers many exciting ideas. However, some subjects deserve further elaboration, which is listed in this section.

6.1 Submodular functions

In this paper, we saw how different functions are similar to the coverage and cut functions. For example, the influence function contains the coverage problem, but it would be interesting to investigate how this coverage problem relates to the coverage function and how it relates to the different models of social networks described in the book by David Kempe, Jon Kleinberg, and Éva Tardos [7].

6.2 M-convex set

We have investigated that orientation and distribution are different definitions of the lattice points of the M-convex sets of the cut and coverage function, respectively. Different submodular functions, such as the influence function, might have a different definition for the lattice points. Moreover, there might be a way to obtain this corresponding interpretation of the lattice point. We have not obtained one for the cut function. However, there might be an algorithm to obtain the lattice points.

References

[1] Jeff Bilmes. Submodularity In Machine Learning and Artificial Intelligence. jan 2022.

[2] Chunsheng Cui, · Baiqiu Li, and · Liu Wang. The selection of COVID-19 epidemic prevention and control programs based on group decision-making. *Complex Intelligent Systems 2021 8:2*, 8(2):1653–1662, jan 2022.

[3] William H Cunningham. Testing membership in matroid polyhedra. *Journal of Combinatorial Theory, Series B*, 36(2):161–188, 1984.

[4] William H Cunningham. On submodular function minimization. *Combinatorica*, 5(3):185–192, 1985.

[5] András Frank. *Connections in Combinatorial Optimization.* Oxford university press, 2011.

[6] Ralph P. Grimaldi. *Discrete and combinatorial mathematics*, chapter 13.3. Pearson Education, 2003.

[7] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015.

[8] L. Lovász. Submodular functions and convexity. *Mathematical programming: the state of the art: Bonn 1982*, pages 235–257, 1983.

[9] K. Murota. Discrete convex analysis. *MATHEMATICAL PROGRAMMING*, 83:313–372, 1998.

[10] Kazuo Murota et al. Discrete convex analysis: A tool for economics and game theory. *Journal of Mechanism and Institution Design*, 1(1):151–273, 2016.

[11] Guoyao Rao, Yongcai Wang, Wenping Chen, Deying Li, and Weili Wu. Maximize the probability of union-influenced in social networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13135 LNCS:288–301, 2021.

[12] Piet van Mieghem. *Data Communications Networking.* Techne Press, Amsterdam, The Netherlands, 206.

[13] Pei Yao, Longkun Guo, Shuangjuan Li, and Huihong Peng. Target coverage with minimum number of camera sensors. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13135 LNCS:12–24, 2021.