Textbook NP Problems

Sijian Tan

1 Problem 2

A store trying to analyze the behavior of its customers will often maintain a two-dimensional array A, where the rows correspond to its customers and the columns correspond to the products it sells. The entry $A[i, j]$ specifies the quantity of product j that has been purchased by customer i.

Here's a tiny example of such an array A.

	liquid detergent	beer	diapers	cat litter
Raj	0	6	0	3
Alanis	2	3	0	0
Chelsea	0	0	0	7

Figure 1: Problem 2

One thing that a store might want to do with this data is the following. Let us say that a subset S of the customers is *diverse* if no two of the customers in S have ever bought the same product (i.e., for each product, at most one of the customers in S has ever bought it). A diverse set of customers can be useful, for example, as a target pool for market research.

We can now define the *Diverse Subset Problem* as follows: Given an $m \times n$ array A as defined above, and a number $k \leq m$, is there a subset of at least k customers that is *diverse*?

Show that Diverse Subset is NP-complete.

1.1 Construction

Independent Set is polynomially reducible to Diverse Subset. Given a graph G and a number k, we construct a customer for each node of G. If any two customers bought the same product, we use an edge to connect them.

We claim that this holds if and only if G has an independent set of size k. If there is a diverse subset of size k, then the corresponding set of nodes has the property that no two are incident to the same edge, so it is an independent set of size k. Conversely, if there is an independent set of size k, then the corresponding set of customers has the property that no two bought the same product, so it is diverse.

2 Problem 3

Suppose you're helping to organize a summer sports camp, and the following problem comes up. The camp is supposed to have at least one counselor who's skilled at each of the n sports covered by the camp (baseball, volleyball, and so on). They have received job applications from m potential counselors. For each of the n sports, there is some subset of the m applicants qualified in that sport. The question is: For a given number $k < m$, is it possible to hire at most k of the counselors and have at least one counselor qualified in each of the n sports? We'll call this the *Efficient Recruiting Problem*.

Show that Efficient Recruiting is NP-complete.

2.1 Construction

Vertex Cover is polynomially reducible to Efficient Recruiting. Given a graph $G = (V, E)$ and an integer k, we would define a sport S_e for each edge e, and a counselor C_v for each vertex v. C_v is qualified in sport S_e if and only if e has an endpoint equal to v.

Now, if there are k counselors that, together, are qualified in all sports, the corresponding vertices in G have the property that each edge has an end in at least one of them; so they define a vertex cover of size k. Conversely, if there is a vertex cover of size k, then this set of counselors has the property that each sport is contained in the list of qualifications of at least one of them.

3 Problem 4

Suppose you're consulting for a group that manages a high-performance real-time system in which asynchronous processes make use of shared resources. Thus the system has a set of n processes and a set of m resources. At any given point in time, each process specifies a set of resources that it requests to use. Each resource might be requested by many processes at once; but it can only be used by a single process at a time. Your job is to allocate resources to processes that request them. If a process is allocated all the resources it requests, then it is *active*; otherwise it is *blocked*. You want to perform the allocation so that as many processes as possible are active. Thus we phrase the *Resource Reservation Problem* as follows: Given a set of processes and resources, the set of requested resources for each process, and a number k, is it possible to allocate resources to processes so that at least k processes will be active? Prove that the problem is NP-complete.

3.1 Construction

Independent Set is polynomially reducible to Diverse Subset. Given a graph G and a number k, we construct a customer for each node of G. If two processors specify the same resource to use, then we add an edge between them.

4 Problem 5

Consider a set $A = \{a_1, \ldots, a_n\}$ and a collection $B_1, B_2, \ldots, B_m$ of subsets of A (i.e., $B_i \subseteq A$ for each i).

We say that a set $H \subseteq A$ is a *hitting set* for the collection $B_1, B_2, \ldots, B_m$ if H contains at least one element from each B_i—that is, if $H \cap B_i$ is not empty for each i (so H "hits" all the sets B_i).

We now define the *Hitting Set Problem* as follows. We are given a set $A = \{a_1, \ldots, a_n\}$, a collection $B_1, B_2, \ldots, B_m$ of subsets of A, and a number k. We are asked: Is there a hitting set $H \subseteq A$ for $B_1, B_2, \ldots, B_m$ so that the size of H is at most k?

Prove that Hitting Set is NP-complete.

4.1 Construction

Vertex Cover is polynomially reducible to Hitting Set. *Hitting Set* looks like a covering problem, since we are trying to choose at most k objects subject to some constraints. We show that Vertex Cover $\leq_P$ Hitting Set. Thus, we begin with an instance of Vertex Cover, specified by a graph $G = (V, E)$ and a number k. We must construct an equivalent instance of *Hitting Set*. In Vertex Cover, we are trying to choose at most k nodes to form a vertex cover. In *Hitting Set*, we are trying to choose at most k elements to form a hitting set. This suggests that we define the set A in the *Hitting Set* instance to be the V of nodes in the Vertex Cover instance. For each edge $e_i = (u_i, v_i) \in E$, we define a set $S_i = \{u_i, v_i\}$ in the *Hitting Set* instance.

Now we claim that there is a hitting set of size at most k for this instance, if and only if the original graph had a vertex cover of size at most k. For if we consider a hitting set H of size at most k as a subset of the nodes of G, we see that every set is "hit," and hence every edge has at least one end in H: H is a vertex cover of G. Conversely, if we consider a vertex cover C of G, and consider C as a subset of A, we see that each of the sets S_i is "hit" by C.

5 Problem 8

Your friends' preschool-age daughter Madison has recently learned to spell some simple words. To help encourage this, her parents got her a colorful set of refrigerator magnets featuring the letters of the alphabet (some number of copies of the letter A, some number of copies of the letter B, and so on), and the last time you saw her the two of you spent a while arranging the magnets to spell out words that she knows.

Somehow with you and Madison, things always end up getting more elaborate than originally planned, and soon the two of you were trying to spell out words so as to use up all the magnets in the full set—that is, picking words that she knows how to spell, so that once they were all spelled out, each magnet was participating in the spelling of exactly one of the words. (Multiple copies of words are okay here; so for example, if the set of refrigerator magnets includes two copies each of C, A, and T, it would be okay to spell out CAT twice.)

This turned out to be pretty difficult, and it was only later that you realized a plausible reason for this. Suppose we consider a general version of the problem of *Using Up All the Refrigerator Magnets*, where we replace the English alphabet by an arbitrary collection of symbols, and we model Madison's vocabulary as an arbitrary set of strings over this collection of symbols. The goal is the same as in the previous paragraph.

Prove that the problem of *Using Up All the Refrigerator Magnets* is NP-complete.

5.1 Construction

Set Cover is polynomially reducible to this problem. Given an instance of the Set Cover problem, where we have a universe U and a collection of subsets $S_1, S_2, \ldots, S_m$, we can construct an equivalent instance of *Using Up All the Refrigerator Magnets* by treating each element in U as a magnet and each subset S_i as a word that can be spelled using those magnets.

6 Problem 9

Consider the following problem. You are managing a communication network, modeled by a directed graph $G = (V, E)$. There are c users who are interested in making use of this network. User i (for each $i = 1, 2, \ldots, c$) issues a *request* to reserve a specific path P_i in G on which to transmit data.

You are interested in accepting as many of these path requests as possible, subject to the following restriction: if you accept both P_i and P_j, then P_i and P_j cannot share any nodes.

Thus, the *Path Selection Problem* asks: Given a directed graph $G = (V, E)$, a set of requests $P_1, P_2, \ldots, P_c$—each of which must be a path in G—and a number k, is it possible to select at least k of the paths so that no two of the selected paths share any nodes?

Prove that Path Selection is NP-complete.

6.1 Construction

Independent Set is polynomially reducible to this problem. Create a graph H where each vertex in H represents one of the paths P_1, P_2, ..., P_c. Add an edge between two vertices in H if the corresponding paths in G share at least one node.

7 Problem 10

Your friends at WebExodus have recently been doing some consulting work for companies that maintain large, publicly accessible Web sites—contractual issues prevent them from saying which ones—and they've come across the following *Strategic Advertising Problem.*

A company comes to them with the map of a Web site, which we'll model as a directed graph $G = (V, E)$. The company also provides a set of t trails typically

followed by users of the site; we'll model these trails as directed paths $P_1, P_2, \ldots, P_t$ in the graph G (i.e., each P_i is a path in G).

The company wants WebExodus to answer the following question for them: Given G, the paths $\{P_i\}$, and a number k, is it possible to place advertisements on at most k of the nodes in G, so that each path P_i includes at least one node containing an advertisement? We'll call this the *Strategic Advertising Problem*, with input G, $\{P_i : i = 1, \ldots, t\}$, and k. Prove this is NP-Complete.

7.1 Construction

Vertex Cover is polynomially reducible to this problem. Construct a new graph H where each node in H represents a node in the original graph G, add an edge between two nodes in H if they are both included in the same path P_i.

A vertex cover in H is a set of vertices such that each edge in H has at least one endpoint in the cover. This corresponds to selecting nodes in G to place advertisements so that every path P_i is "covered," meaning it has at least one node with an advertisement.

8 Problem 11

Given a directed graph G, with start node s and end node t, and thematic elements represented by sets $T_1, T_2, \ldots, T_k$, the Plot Fulfillment Problem asks: Is there a path from s to t that contains at least one node from each of the sets T_i?

Prove that Plot Fulfillment is NP-complete.

8.1 Construction

Hampath is polynomially reducible to this problem. The procedures include:

1. Let $G' = (V', E')$ be the graph instance of the Hampath problem
2. Construct a directed graph $G = (V, E)$ as follows:
   - Set $V = V' \cup \{s, t\}$, where s, t are new vertices
   - The nodes s, t are added to G, with s corresponding to the start vertex s' in G', and t corresponding to the end vertex t' in G'
   - Create thematic sets $T_i = \{v_i\}$ for each $v_i \in V'$, ensuring that each vertex in the original graph G' is included in exactly one thematic set
3. For every directed edge (u, v) in E', include the edge (u, v) in E.
4. Add a directed edge from s to s' and from t' to t

If there is a Hamiltonian path in G' from s' to t', then the corresponding path in G from s to t will visit all vertices in V' exactly once, thereby satisfying the Plot Fulfillment Problem since it will visit all thematic sets.

Conversely, if there is a path in G from s to t that intersects all thematic sets T_i, then this path corresponds to a Hamiltonian path in G' from s' to t'.

8.2 Problem 12

More precisely, we define the **Evasive Path Problem** as follows: Given G, $Z_1, \ldots, Z_k$, $s \in V$, and a destination node $t \in V$, is there an $s - t$ path in G that includes at most one node from each zone Z_i? Prove that Evasive Path is NP-complete.

9 Construction

Same as last question.

10 Problem 13

Here's a simple type of combinatorial auction. There are n items for sale, labeled $I_1, \ldots, I_n$. Each item is indivisible and can only be sold to one person. Now, m different people place bids: The ith bid specifies a subset S_i of the items, and an **offering price** x_i that the bidder is willing to pay for the items in the set S_i, as a single unit. (We'll represent this bid as the pair (S_i, x_i).)

An auctioneer now looks at the set of all m bids; she chooses to **accept** some of these bids and to **reject** the others. Each person whose bid i is accepted gets to take all the items in the corresponding set S_i. Thus the rule is that no two accepted bids can specify sets that contain a common item, since this would involve giving the same item to two different people.

The auctioneer collects the sum of the offering prices of all accepted bids. (Note that this is a "one-shot" auction; there is no opportunity to place further bids.) The auctioneer's goal is to collect as much money as possible.

Thus, the problem of **Winner Determination for Combinatorial Auctions** asks: Given items $I_1, \ldots, I_n$, bids $(S_1, x_1), \ldots, (S_m, x_m)$, and a bound B, is there a collection of bids that the auctioneer can accept so as to collect an amount of money that is at least B?

Example. Suppose an auctioneer decides to use this method to sell some excess computer equipment. There are four items labeled "PC", "monitor", "printer", and "scanner"; and three people place bids. Define

$$S_1 = \{\text{PC, monitor}\}, \quad S_2 = \{\text{PC, printer}\}, \quad S_3 = \{\text{monitor, printer, scanner}\}$$

and

$$x_1 = x_2 = x_3 = 1.$$

The bids are (S_1, x_1), (S_2, x_2), (S_3, x_3), and the bound B is equal to 2.

Then the answer to this instance is no: The auctioneer can accept at most one of the bids (since any two bids have a desired item in common), and this results in a total monetary value of only 1.

Prove that the problem of **Winner Determination in Combinatorial Auctions** is NP-complete.

10.1 Construction

Independent Set is polynomially reducible to this problem. Create a graph H where each vertex in H represents one of the bidders. Add an edge between two vertices in H if two bidders place on same item. Set the value of each bid to 1. Therefore, we ask whether the auctioneer can accept a set of bids of total value $B = k$.

We claim that this holds if and only if G has an independent set of size k. If there is a set of acceptable bids of total value k, then the corresponding set of nodes has the property that no two are incident to the same edge — so it is an independent set of size k. Conversely, if there is an independent set of size k, then the corresponding set of bidders has the property that their bids are disjoint, and their total value is k.

11 Problem 14

Now you're given a set of n jobs, each specified by a set of time intervals, and you want to answer the following question: For a given number k, is it possible to accept at least k of the jobs so that no two of the accepted jobs have any overlap in time?

Show that Multiple Interval Scheduling is NP-complete.

11.1 Construction

Independent Set is polynomially reducible to this problem. Create a graph H where each vertex in H represents each of the job. Add an edge between two vertices in H if two jobs are overlapped.

12 Problem 16

Consider the problem of reasoning about the identity of a set from the size of its intersections with other sets. You are given a finite set U of size n, and a collection $A_1, \ldots, A_m$ of subsets of U. You are also given numbers $c_1, \ldots, c_m$. The question is: Does there exist a set $X \subseteq U$ so that for each $i = 1, 2, \ldots, m$, the cardinality of $X \cap A_i$ is equal to c_i? We will call this an instance of the **Intersection Inference Problem**, with input U, $\{A_i\}$, and $\{c_i\}$.

Prove that Intersection Inference is NP-complete.

12.1 Construction

Hitting Set is polynomially reducible to this problem. Given an instance of the Hitting Set Problem with:

- A finite set $U = \{u_1, u_2, ..., u_n\}$
- A collection $S = \{S_1, S_2, ..., S_m\}$ where each $S_i \subseteq U$
- An integer k

Now construct an instance of the intersection inference problem:

1. Use the same set U from the hitting set instance
2. Set $A_i = S_i$, this ensures that the subsets in the Intersection Inference Problem correspond directly to the subsets in the Hitting Set Problem
3. Set $c_i = 1$, this means we are looking for a set $X \subseteq U$ that intersects each A_i in exactly one element, which aligns with the requirement in the Hitting Set Problem for X to hit each subset.

13 Problem 17

You are given a directed graph $G = (V, E)$ with weights w_e on its edges $e \in E$. The weights can be negative or positive. The **Zero-Weight-Cycle Problem** is to decide if there is a simple cycle in G so that the sum of the edge weights on this cycle is exactly 0. Prove that this problem is NP-complete.

13.1 Construction

Subset Sum is polynomially reducible to this problem. We are given numbers $w_1, \ldots, w_n$, and we want to know if there is a subset that adds up to exactly W. We construct an instance of *Zero-weight-cycle* in which the graph has nodes $0, 1, 2, \ldots, n$, and an edge (i, j) for all pairs $i < j$. The weight of edge (i, j) is equal to w_j. Finally, there is an edge $(n, 0)$ of weight $-W$.

We claim that there is a subset that adds up to exactly W if and only if G has a zero- weight cycle. If there is such a subset S, then we define a cycle that starts at 0, goes through the nodes whose indices are in S, and then returns to 0 on the edge $(n, 0)$. The weight of $-W$ on the edge $(n, 0)$ precisely cancels the sum of the other edge weights. Conversely, all cycles in G must use the edge $(n, 0)$, and so if there is a zero-weight cycle, then the other edges must exactly cancel $-W$ — in other words, their indices must form a set that adds up to exactly W.

14 Problem 19

Handling hazardous materials adds additional complexity to what is, for them, an already complicated task. Suppose a convoy of ships arrives in the morning and delivers a total of n cannisters, each containing a different kind of hazardous material. Standing on the dock is a set of m trucks, each of which can hold up to k containers.

Assume any cannister can be placed in any truck; however, there are certain pairs of cannisters that cannot be placed together in the same truck. (The chemicals they contain may react explosively if brought into contact.) Is there a way to load all n cannisters into the m trucks so that no truck is overloaded, and no two cannisters are placed in the same truck when they are not supposed to be? Prove this problem is NP-Complete.

14.1 Construction

Independent Set is polynomially reducible to this problem. Given an instance of the Independent Set Problem with:

- A graph $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices and E is the set of edges.
- A target size k.

We will construct an instance of the **Hazardous Material Loading Problem**.

1. **Vertices to Cannisters:**
   - Each vertex v_i in G corresponds to a cannister c_i in the Hazardous Material Loading Problem.
2. **Edges to Incompatibilities:**
   - For each edge $(v_i, v_j) \in E$, define that the corresponding cannisters c_i and c_j cannot be placed together in the same truck. This models the "incompatibility" based on adjacency in the graph.
3. **Trucks to the Independent Set Size:**
   - Set the number of trucks m equal to 1, and the capacity k of the truck equal to the independent set size k. This means we are trying to load up to k cannisters into a single truck.
4. **Interpretation:**
   - Finding a way to load the cannisters into the trucks without violating the constraints corresponds directly to finding an independent set of size k in the graph G. Specifically, placing k mutually compatible (non-adjacent) cannisters in the truck corresponds to selecting k non-adjacent vertices in the graph.

15 Problem 21

We can now define the **Fully Compatible Configuration (FCC) Problem**. An instance of FCC consists of disjoint sets of options $A_1, A_2, \ldots, A_k$, and a set P of *incompatible pairs* (x, y), where x and y are elements of different sets of options. The problem is to decide whether there exists a fully compatible configuration: a selection of an element from each option set so that no pair of selected elements belongs to the set P.

Example. Suppose $k = 3$, and the sets A_1, A_2, A_3 denote options for the operating system, the text-editing software, and the e-mail program, respectively. We have

$$A_1 = \{\text{Linux, Windows NT}\},$$

$$A_2 = \{\text{emacs, Word}\},$$

$$A_3 = \{\text{Outlook, Eudora, rmail}\},$$

with the set of incompatible pairs equal to

$$P = \{(\text{Linux, Word}), (\text{Linux, Outlook}), (\text{Word, rmail})\}.$$

Then the answer to the decision problem in this instance of FCC is yes—for example, the choices Linux $\in A_1$, emacs $\in A_2$, rmail $\in A_3$ is a fully compatible configuration according to the definitions above.

Prove that Fully Compatible Configuration is NP-complete.

15.1 Construction

Independent Set is polynomially reducible to this problem. Combine all the options from A_i and let them be the nodes. If two options are incompatible, just add an edge.

16 Question 26

This suggests solving the following **Number Partitioning Problem**. You are given positive integers $x_1, \ldots, x_n$; you want to decide whether the numbers can be partitioned into two sets S_1 and S_2 with the same sum:

$$\sum_{x_i \in S_1} x_i = \sum_{x_j \in S_2} x_j.$$

Show that Number Partitioning is NP-complete.

16.1 Construction

Subset Sum is polynomially reducible to this problem. Given an instance of a subset sum:

- A set of integers $H = \{a_1, a_2, ..., a_n\}$
- A target integer T

For the subset sum problem, there exists:

$$\sum_{a_i \in H'} a_i = T \tag{1}$$

Now we construct the Number Partitioning instance:

1. Assuming the total sum of $H = TS$
2. Construct the universal set as $S = \{x_1, x_2, ..., x_n, x_{n+1}\}$. Here, $x_k = a_k$ for $k = 1, 2, ..., n$ and $x_{n+1} = TS - 2T$
3. Now the total sum of $S = (TS + TS - 2T) = 2TS - 2T$

4. Construct the subset S_1 as $S_1 = \{H', x_{n+1}\}$, so that we have:

$$\sum_{x_i \in S_1} x_i = T + TS - 2T = TS - T \tag{2}$$

5. Therefore the remaining part of S (denoted as S_2) will satisfy:

$$\sum_{x_j \in S_2} x_j = (2TS - 2T) - (TS - T) = TS - T \tag{3}$$

6. Finally we have:

$$\sum_{x_i \in S_1} x_i = \sum_{x_j \in S_2} x_j \tag{4}$$

17 Problem 27

Consider the following problem. You are given positive integers $x_1, \ldots, x_n$, and numbers k and B. You want to know whether it is possible to *partition* the numbers $\{x_i\}$ into k sets $S_1, \ldots, S_k$ so that the squared sums of the sets add up to at most B:

$$\sum_{i=1}^{k} \left(\sum_{x_j \in S_i} x_j \right)^2 \leq B.$$

Show that this problem is NP-complete.

17.1 Construction

Subset Sum is polynomially reducible to this problem. Given an instance of a subset sum:

- A set of integers $H = \{a_1, a_2, ..., a_n\}$
- A target integer T

For the subset sum problem, there exists:

$$\sum_{a_i \in H'} a_i = T \tag{5}$$

Now we construct the new instance:

1. Set the integers $S = \{x_1, x_2, ..., x_n\}$ to be equal to the elements $\{a_1, a_2, ..., a_n\}$ from the Subset Sum problem.

2. Set $k = 2$, this means we want to partition the set into two subsets.

3. Set $B = T^2$. This is because we want to find a partition where one subset sums to T, which gives a squared sum of T^2, and the other subset should sum to 0, resulting in a squared sum of 0. Thus, the total sum of squares would be T^2.

4. Therefore, the following equation is satisfied:

$$\left(\sum_{x_j \in S_1} x_j\right)^2 + \left(\sum_{x_j \in S_2} x_j\right)^2 \leq T^2 \tag{6}$$

18 Problem 28

The following is a version of the Independent Set Problem. You are given a graph $G = (V, E)$ and an integer k. For this problem, we will call a set $I \subseteq V$ *strongly independent* if, for any two nodes $v, u \in I$, the edge (v, u) does not belong to E, and there is also no path of two edges from u to v, that is, there is no node w such that both $(u, w) \in E$ and $(w, v) \in E$. The **Strongly Independent Set Problem** is to decide whether G has a strongly independent set of size at least k.

Prove that the Strongly Independent Set Problem is NP-complete.

18.1 Construction

Independence Set is polynomially reducible to this problem. Given a graph G and a number k, we construct a new graph G' in which we replace each edge $e = (u, v)$ by a path of length two: we add a new node w_e, and we add edges (u, w_e), (w_e, v). We also include edges between every pair of new nodes.

Now suppose that G has an independent set of size k. Then in this new graph G', all these k nodes are distance at least three from each other, so this is a strongly independent set of size k. Conversely, suppose G' has a strongly independent set of size k. Now, this set can't contain any of the new nodes, since all such nodes are within distance two of every node in the graph. Thus, it consists of nodes present in G. Moreover, no two of these nodes can be neighbors in G, since then they'd be at distance two in G'. Thus this set of nodes forms an independent set of size k in G.

19 Problem 30

It turns out there's a reason for this: Minimizing an n-variable func- tion over the unit cube in n dimensions is as hard as an NP-complete problem. To make this concrete, let's consider the special case of poly- nomials with integer coefficients over n variables $x_1, x_2, \ldots, x_n$. To review some terminology, we say a *monomial* is a product of a real-number co- efficient c and each variable x_i raised to some nonnegative integer power a_i; we can write this as $cx_1^{a_1}x_2^{a_2}\ldots x_n^{a_n}$. (For example, $2x_1^2x_2x_3^4$ is a monomial.) A *polynomial* is then a sum of a finite set of monomi- als. (For example, $2x_1^2x_2x_3^4 + x_1x_3 - 6x_2^2x_3$ is a polynomial.)

We define the **Multivariable Polynomial Minimization Problem** as fol- lows: Given a polynomial in n variables with integer coefficients, and given an integer bound

B, is there a choice of real numbers $x_1, x_2, \ldots, x_n \in [0, 1]$ that causes the polynomial to achieve a value that is $\leq B$?

Choose a problem Y from this chapter that is known to be NP- complete and show that

$$Y \leq_P \text{Multivariable Polynomial Minimization.}$$

19.1 Construction

Subset Sum is polynomially reducible to this problem. Given an instance of a subset sum:

- A set of values $H = \{a_1, a_2, ..., a_n\}$
- A target value T

For the subset sum problem, there exists:

$$\sum_{a_i \in H'} a_i = T \tag{7}$$

Now we construct the new instance:

1. Assume there are n monomials, denoted by $\{m_1, m_2, ..., m_n\}$, corresponding to $\{a_1, a_2, ..., a_n\}$ which will be the universal set S.
2. Set the target value $B = T$
3. Therefore, the following equation is satisfied:

$$\sum_{m_i \in S'} m_i \leq B \tag{8}$$

because $x_n \in [0, 1]$ and each monomial $m_i \leq 1$.