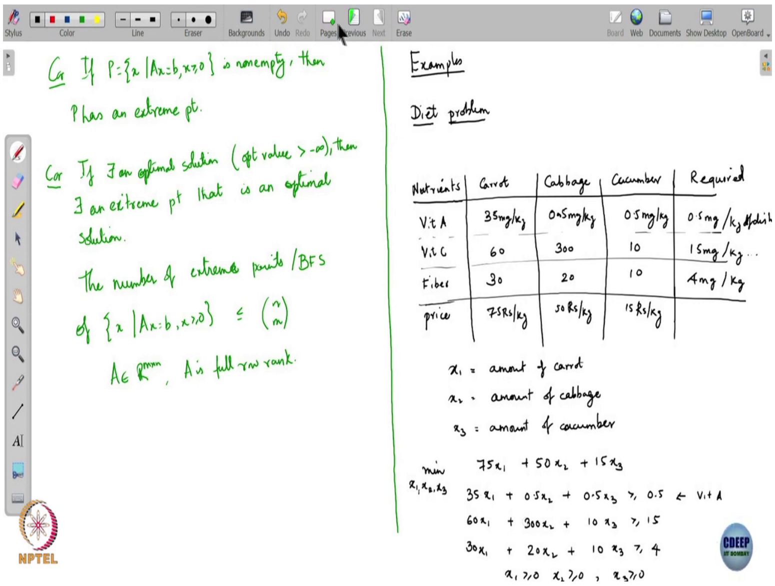
Optimization from Fundamentals
Prof. Ankur Kulkarni
Department of Systems and Control Engineering
Indian Institute of Technology, Bombay

Lecture - 11C
Examples of Linear Programming

Let us just do a couple of Examples of Linear Programming.

(Refer Slide Time: 00:25)



And so this is the popular example, it is called the problem is called the diet problem. So, imagine the Ministry of Human Resource Development that it which provides midday meals to children in school. It is suppose mandates that a certain nutritional requirement for the meals, right.

So, it says you have a list of nutrients; such as say vitamin A, vitamin C, suppose fiber, you have carbohydrates, proteins, etcetera. These are nutrients that you must that your foods must have. And the ministry specifies what is the amount of nutrients that you are that you need to have so, for example, it says that say vitamin A should be in every meal in every dish that you create it must have say half a milligram of vitamin A, ok.

So, the requirements here are all in. so in milligrams per kg of the dish ok. Vitamin C suppose in similar ways say 15 milligrams per kg of the dish. Say fiber suppose similarly is say 4 milligrams etcetera. Now, what you the choice you have is to look at the is to look at foodstuffs that are on offer ok, ingredients that are on offer, and to see what sort of dishes you can create out of this. Say for example, you have three food ingredients you can say one is carrot, suppose other is cabbage, and say cucumber.

Now, this does not complete a meal, but this is for this is just for illustrative purposes. So, these are the last column here are the required, required amounts of these nutrients. Now, what we know is say carrot has 35 milligrams of vitamin A per kg of the carrot; cabbage has 0.5 milligrams per of vitamin A per kg of the cabbage; cucumber has also 0.5 milligrams per kg.

The vitamin C is say similarly 60, here say 300 suppose, and say 10. Fiber, carrot has say 30, this is 20, and this is suppose 10, ok. Now, let us ignore the carbo and proteins for the moment. So, question to the school is to now say well this is what the ministry mandates that the children should be fed at least 0.5 milligrams per kg of vitamin A, they must be fed 15 milligrams of vitamin C per kg, 4 milligrams of fiber per kg.

And the and you have these three ingredients on offer which is a carrot, cabbage, and cucumber. What you need to do is come up with decide what should be the fractions of all of these in order to meet these requirement. Now, what is the objective? The objective is say suppose these are these all have some prices, ok. So, suppose the price let us let me erase this for the moment.

Let me look at the price. Let us say the price of carrot is say 75 rupees per kg, this is 50 rupees per kg, and this is 15 rupees per kg. These are the prices. So, if you the question for the school is to now say what is the cheapest combination that it can make, so that the requirements are satisfied alright.

So, what are the variables here, and what are the constraints? What are the decision variables? The decision variables are how much how many kg's of how many kg's of carrot would you add, how many kg's of cabbage would you add, how many kg's of cucumber would you add, right.

So, suppose let us say x 1 is the amount of carrot, x 2 is the amount of cabbage, x 3 is the sorry amount of cabbage and this is the amount of cucumber. So, we can write this as a linear program. So, the objective is to come up with the cheapest combination. So, the cost, if you have x 1 amount of carrot, x 2 amount of cabbage and x 3 amount of cucumber, the cost that you will accrue is 75 into x 1 plus 50 into x 2 plus 15 into x 3, this will be this will be the cost that you will have per this would be the cost of, this would be the cost that you will encounter for per kg of the dish that you create.

And now, what does, how do, what are my constraints? My constraints are that my nutritional requirement should be satisfied. Now, if I have x 1 amount of carrot, then I know that it is going to give me 35 milligrams of vitamin A. So, 35 into x 1 is the amount of vitamin A that I get plus 0.5 into x 2 is the amount of vitamin A I get from cabbage, plus 0.5 into x 3 is the amount of vitamin A I get from cucumber.

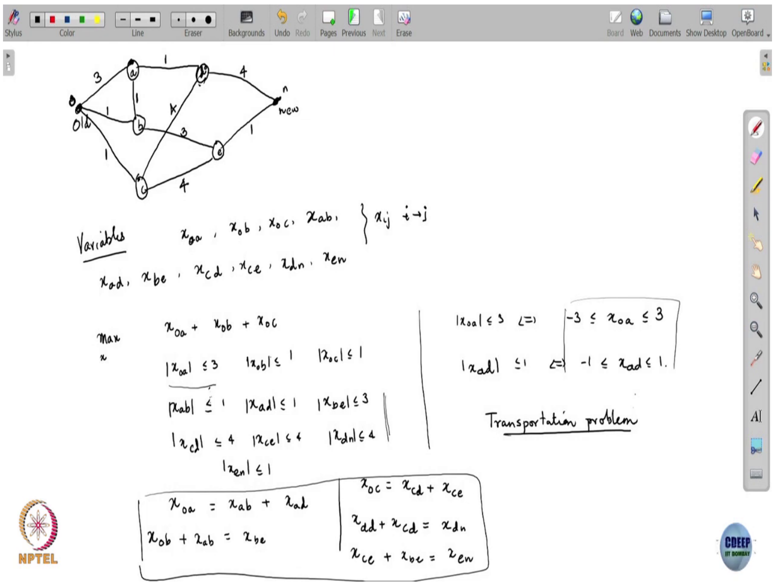
All of these, this should be greater than equal to the nutritional requirement which is 0.5, right. Similarly, I have an additional requirement which is 60 into x with this is for so this pair was for vitamin A; I have similarly for vitamin C. So, 60 into x 1 plus sorry 300 into x 2 plus a 10 into x 3, and this should be greater than equal to 15.

And finally, 30 into x 1 plus 20 into x 2 plus 10 into x 3, this should be greater than equal to 4. Any constraints that I am missing in this? Yeah. So, all these, these are this is the amount of cabbage etcetera, etcetera per kg of the dish, so consequently they have these all have to be greater than equal to 0. So, I need x 1 greater than equal to 0, x 2 greater than equal to 0, and x 3 greater than equal to 0. You can see this is now a linear programming in the variables x 1, x 2, x 3.

Now, this does not tell you a recipe. This just tells you the proportions ok. You can now using this proportion you would devise the actual thing that you want to create out of this. So, I will give you, let us do one. So, this was actually remarkably a very simple problem, but is a famous problem because this was actually one of the early problems studied.

So, I told you this in the as a story about midday meals for school children, it was actually studied as nutritional requirements for soldiers about what should they be you know what sort of meals should they be having in order to meet their nutritional requirement. So, this was one of the early problems that was actually formulated and studied as a linear problem, ok.

(Refer Slide Time: 10:29)



So, here is another here is another example. This is a slightly more complex example. So, suppose you have an old computer, an old computer o here ok, old computer. And you want to you are going to migrate yourself to a new computer this. So, this is o, and this is n. So, this is your a new computer.

Now, when you want to migrate from old to new; suppose, it so you need to now move your data from old to new and now the computer man you do not have say you do not have a say huge amount of data you do not have a pendrive or something that can carry all of it.

What you can do is transfer the data over the over a network, ok. So, the data can be moved around like this. So, suppose this was so you have a network like this comprising of these various computers. So, here is a, here is b, here is c, and then there is d, and then there is e, alright.

And the network is connected like this, o is connected to a; o is connected also to b, sorry it will be like this a is connected to b; b is connected c; o is connected to c; c is connected to d. So, c is connected to d like this; a is connected to d; b is connected to e; c is connected to e; and e is finally, connected to n; and d is also connected to n.

So, your data starts from o here and has to get to n. Now, it can flow over any of these, these linkages that are present between the computers, ok. So, imagine these are some fiber optic cables or whatever. And they it can flow over any of these linkages. There are limitations in the sense that you cannot pump more data through a link than its capacity, ok.

So, the capacities of these links is given is written like this. So, here is the capacity of o to a is 3; capacity of o to b is 1; capacity of o to c is 1; capacity of c to e is 4; capacity of c to d is 4; capacity of b to e is 3; a to b is 1; a to d is also 1; and d to n is 4, alright. Now, the way we interpret this, these capacities, this is a capacity for flow in either direction.

So, you can send data when you when I say that the capacity of c d link is 4, it means that 4 mbs per second can move from c to d, or from d to c ok, that is the that is what we mean by that is the meaning of this capacity, alright. It let us also assume for simplicity that although these are computers lets also assume that they do not have any local storage.

So, whatever data that comes in has to be then transferred has to be sent out through the same node instantaneously to another whatever data that comes in to the node has to be also shipped out to another node, alright. So, question now is what is the what is the maximum flow that what should be the, how should you send this data, so through this network, so that you get the maximum data rate? e to n is 1, this, sorry right. So, what are the variables in this?

Student: (Refer Time: 14:28)

Yeah. So, the variables, the variables here are the variables are the these variables are x ij where x ij just talks of how what is the mbs per second of data that is flowing on link i to j. Now, we do not we will not make any assumptions about the direction let the direction get determined on its own, alright. So, what, we will just have these variables let us declare them. So, x o a is from old to a; x ob, x oc, then x ab, x ad, then x be, then x cd, x ce, x dn, and x en, these are the variables.

Now, what do you want to what do you want to maximize? You would like to maximize that the flow that goes out of the flow that goes out of a out of o. Now, the flow that goes out of o, because nothing is stored anywhere on the network, the flow that goes out of o is also equal to the flow that reaches n per, ok. The mbs per second that flows out of o is also equal to the mbs per second that will reach n, because nothing is stored in between whatever comes in must go out, ok.

So, let us say we want to maximize the maximize x oa plus x ob plus x oc, this is the flow that is going from that is going out of o, alright.

Now, what are the constraints? The variables are this whatever be the direction of the flow, you are constrained by the capacity, which means that the if you look at x oa in absolute value it must be less than equal to 3; x ob in absolute value must be less than equal to 1; x oc likewise in absolute value should be less than equal to 1, etcetera.

I can write I can write all of these. So, let us maybe just let us just write them out. So, x ab has to be less than equal to 1; x ad has to be less than equal to 1; x be absolute value less than equal to 3; x cd less than equal to 4; x ce less than equal to 4; x dn less than equal to 4; and x en is less than equal to 1.

The these, this these variables the, so these constraints encapsulate that your capacity of the network of every link in the network must be respected. Now, we need to also use the fact that the network does not store anything, right.

So, whatever comes into a is also what goes out of a. Now, let us we the way we have the way this is all written we have a convention here about that x oa, when I am saying I am maximizing this oa plus ob plus oc, it means it is flow that is going from o to a, o to b, and o to c.

So, x ij is flow from i from i to j, right. So, that is, so if I so the flow that comes into a then is flow that comes into a there is only one in terms of the variables that I have defined, in terms of the variables that I have defined, there is only one variable that stands for flow that comes into a and that is x oa, right. So, x oa is the flow that comes into a the flow that should be equal to the flow that goes out of a.

What are the flows that go out of a? The flow that goes out of a is ab plus ad, right. So, x oa is equal to x ab plus x ad, ok. Now, likewise let us do say x ob this is the flow that goes in what is the flow that comes into b? It is ob plus ab, right. So, remember i x ij is flow that goes from i to j, right. So, x ob plus x ab, this should be equal to the flow that goes out of b, which is x be. What is the flow that comes into c? It is x oc that should be equal to x cd plus x ce, flow that comes into d. What is the flow that comes into d?

Student: ad.

ad plus cd right; x ad plus x ce and that should be equal to the flow that goes out of d, which is x d dn ok. And finally, for e the flow that comes into e is x ce plus x be, and that should be equal to the flow that goes out of e which is x en. Now, if you look at these equations the ones that are written here in these in this box, look at these five equations, you will see that if I just add them all up, what would I get?

What I get on the left hand side is x oa plus x ob plus x oc several terms are going to cancel when I add them up x oa plus x ob is so this plus this guy when I add, for example, x ab will cancel with x ab here. So, x ab and x ab will cancel here; x ad here will cancel with this x ad; x be will cancel with this x be; x ce will cancel with this x ce; x cd will cancel with this.

So, eventually on the left hand side, all I will be left with is x oa, x ob, and x oc, right which is my objective x oa plus ob plus oc. What will I will what will be left on the right hand side? It will be x dn plus x en, these are the only terms that do not have a corresponding thing on the left hand side. Everything else is going to get cancelled. So, that is what I said at the start that this is this here is the flow that goes out of o, and that will be equal to the flow that enters n, alright.

So, you can model this problem in either way. So, long as you have written your flow constraint correctly, you will both will give you the same answer right. You can model it as floor leaving o or you can model it as entering n alright ok, alright. Now, this is still not the linear program. Why is this not a linear program? It is not a linear program because my constraints here, these constraints are all non-linear, because I have an absolute value here, right.

But I can write these in a linear form in a very easy way. So, any of these let us say for example, if I take this one, this mod absolute value of this less than equal to 3 is the same as saying x oa less greater than equal to minus 3 less than equal to 3, right. And similarly say let us do one more x mod of x ad less than equal to 1 that is equivalent to minus 1 less than equal to x ad less than equal to 1, ok.

Student: (Refer Time: 23:28)

Link capacity is not negative.

Student: (Refer Time: 23:33)

Yeah. So, it is the x oa greater than equal to minus 3 means just the flow from the flow from a to o can be at most 3 ok, yeah. So, this by doing this I can convert this to a linear program. Am I missing any constraint? Do I need to put x greater than equal to 0, x oa greater than equal to 0, x ab greater than equal to 0, etcetera? Yes or no? Ok.

So, if you put that then you are already presuming a certain direction for the flow. The advantage of not of keeping all this unsigned is that let you let the program determine the sign of the direction of the flow, right. So, what you fix is only the convention of for interpreting the variables, not the direction to begin with.

If you wanted to put a direction for x oa and as greater than equal to 0, then you need to declare another variable because there could be a possibility for flow in the opposite direction also, right. We would have to then declare another variable for flow in the opposite direction. Is it clear?

So, since we are not keeping any we are letting the sign of x get determined on its own, we do not need to we do not we do not need to do that, alright. So, this completes the problem formulation all of these linear constraints this and the corresponding one for these absolute value constraints, and these linear equations.

So, this is an example of what is called a transportation problem. So, they have this particular the transportation problem always has this particular feature that if you add up all if you add up all the rows, you would get back on the left hand side or on the right hand side the objective of your problem, ok.

So this is. so, you can of course, use this to model not just data flow you can use this model the material flow, flow of, you know flow of troops, flow of cargo, etcetera, etcetera, ok. So, we will stop here. And next time we will continue with the Duality Theory of Linear Program.