Hello? Yeah. No, I'm just asking. Yeah, okay I know. What's wrong with this? Okay. Now. Yes. Hi. Good afternoon. How are you today? Okay. I believe you're a little bit tired. Probably a little. Afraid of having three hours. Okay, I promise you two breaks. Okay, so you let me know when you're tired. After 15 minutes, maybe 15 minutes break, more 15 minutes break. And the remaining part of the lecture. All right, so you let me know. We are tired. Okay? We want to leave. Obviously toilet and water, things like that. Anyway, so I have two so three remarks. One remark is the third remark is about the individual coursework. We'll talk about that. The first remark is about next week. So next week according to our schedule. First of all I don't know if you notice that, but I didn't I didn't change anything regarding the length of each lecture. This lecture was, since the very first beginning is scheduled for three hours. Okay, just to let you know, I didn't change anything. When I want to change something in general, I make an announcement before. But anyway, so this is one thing. This is zero actually. Okay, regarding regarding next lecture, according to our schedule, next lecture will be dedicated to a case study. So we started a case study last Friday during our tutorial. I don't know if all of you guys were there, so my idea is to use 51 hour as a follow up for the case study, because I suppose we didn't have time to analyse appropriately. And I would like to show you guys how to provide a good or more comprehensive analysis. Okay, of case study. This can be very helpful for your life, for the project, for whatever. Okay. Like how to build a nice report, where to focus when you're running instances and experiments, what you test, your research questions associated with the computational experiments okay. So I suppose we can do that you one hour because we know the problem we call the charge. So please bring your laptops. So you have this one hour. The remaining hour I want to I want to propose project consultation again. So you'll be 50 minutes. Some groups they may meet, they have some questions. And it's good just to have all your guys here if you want to go for project consultation. Obviously this is not mandatory if you don't have questions, if you don't want to show me anything, that's fine. But if you have something already, this is a great opportunity for you guys to show me your model. The precise model. Because at this point, I understand you guys discussed several ideas. I'm okay with most ideas. Some ideas. I'm still a little bit puzzled with them, but I'm pretty sure you guys would discuss with your groups first. But anyways, you have at least one week to show me the mathematical optimisation because again, eventually the idea it's very difficult to grasp if I don't see the mathematical optimisation what you want to propose. So that's why for several different groups, my suggestion was okay, I'm not against this idea, but try to model this problem. Try to represent your decision making problem first. Try to code even. Very simple, a very simple instance or illustrative toy just to make sure things are okay are aligned. So if you guys bring me at least the model and eventually some intuition about the results, what to expect in terms of results, in terms of data collection or if you're generating data, whatever. Definitely my feedback will be way, way more useful. Okay. And this can help you guys a lot. So this is my proposition for next week. Is it clear? Unless you have something very against this idea. So yeah, let me know if this is the case. Okay. But other than that, I suppose we have the follow up of the case study. So please try to run try to analyse a little bit. First we can we can start with the idea of equity in that model when we realise that the allocation wasn't fair. I don't know if you remember this discussion, but yes, try to propose something we can discuss. And yes, and this is one point, right? Second point, not next week, but the week after we have our guest speaker in person here. So this guest speaker is coming from London. He works in a consultancy company of optimisation mathematical programming. That's why I ask him to come from London. And I suppose this will be very interesting because it's someone what is not an academic. I mean, sometimes it's very difficult to convince you guys that prescriptive optimisation, mathematical program is cool. People use that, even though if you were there in the Amazon seminar, you realise that it's crazy what Amazon is doing regarding mathematical programming, machine learning, everything. But now we have someone from a consultancy company working with several different projects, and this person is obviously is very different from Amazon because Amazon, they have this team of scientists. So everybody that works with optimisation, Amazon, like most people, they have at least a PhD or master's degree in this area. Consultancy companies a little bit different. Sometimes you don't need a master's degree or even a bag to be in charge of some optimisation projects. So I suppose will give you guys a different idea or different perspective on how to implement projects involved. Mathematical programming. Okay, so you'll be in two weeks. I really hope you guys are here. What else? Next week. On Thursday we have a seminar in prescriptive analytics as well. It's not for this course, it's for the business school as a whole. But I just like to extend the invitation to you guys is one of my closest collaborators in France. He's in Germany, but he's he's a professor in Brazil. But now he's on his sabbatical in Vienna in Germany. He works with vehicle routeing problems. So if you have groups that basically work with vehicle routeing bike sharing schemes, it's a very nice, interesting application. So his seminar will be on next Thursday. At 2 p.m.. Lecture Theatre five. Not super confident about lecture theatre five. But yes, I do make announcements. I'm just extending the invitation. So when we receive these announcements, you'll have like, what is that? I don't know, what is that? Obviously not mandatory. I'm just extending the invitation. The seminar will happen. Okay. For the business school as a whole. But anyways, feel free to join if you want to know a little bit more. He's an academic, so that talk could be a little bit more academic okay. Which can be good for you guys as well. Last thing obviously. Last but not least you are regarding the individual coursework. So what is individual coursework, as I told you guys in the first lecture. So the idea is we don't have an exam as such. So the the coursework, the coursework is like a very tiny project that you can do. Obviously you don't need for eight hours. This is started during Covid. Okay. Theoretically, this project was supposed to be done in a couple of three four hours maximum. But because of Covid, we start to, you know, to flexible things a little bit more because, you know, you know, everything was a little bit more flexible. And I maintain it because I realised this can be a good opportunity for you guys to learn a little bit as well during the exam. Obviously this is not ideal. The ideally you learn before the course assignment, but anyways, I understand that. But in any case, the individual assignment basically should reflect all the course content. And I and I was very clear about that during the first lecture. That's why we are discussing so many different topics, mostly focussed on model building using linear programming. Using integer programming, we change the modules, we change objective functions. I make a lot of questions about feasibility when it's feasible, it's not feasible, things like that. So you can expect the same type of questions and discussions in the individual individual assignment. You have questions I don't know how many questions in general. You don't have more than five questions of it all. Okay. In general, between 4 and 5 one year it was ten questions, but ten questions way more straightforward. For example, solving this problem. This, this kind of thing. But yes, I'm done with this type of question because it's true, straightforward. It's not helpful a lot for me to assess how much you grasp from this course. So what we have right now is a property between the course content and the group project. What type of questions that I can make regarding the project guys. Obviously I'm not making particular questions about your project because these would be impossible for me. I need you to make 12. I don't know how many groups we have or how many projects. Anyways, obviously this is not feasible for me. It's not feasible for you guys. This is not the year to two types of questions from previous years. One question was is that your decision making problem? When you say your decision making problem, the decision making problem you studied in your group project is stated 100 words like, you know, it's like a peach. What is your decision making problem? The importance of dealing with this decision making problem, and justify to what extent stakeholders and overall decision makers they can benefit from the solutions provided by your model. Theoretically, you should be able to answer in five minutes those questions. If you really helped your group to build and to analyse the follow up question would be state. The mathematical model relates to your decision making problem. So you have to state you have to define index parameters. You know, the structured decision variables, objective function constraints. And you should be able to explain again you are being assessed. So you put a constraint and there is no oh this constraint is a logical constraint. What is that. I know he's a logical constraint, but I have just asked you guys to make sure you know what a logical constraint is or oh, this is a conservation flow constraint. I know what a conservation flow constraint is, but do you know. So you should be able because there is no final recipe for how many words, how much to explain it should be there. It should be clear. The explanation should be there. Because again, you can consult with any material with the lecture notes. So try to build a good answer. Try to explain as much as you can. Considering the limitation of space then definitely EU you have. And again, I know every year should just complain about the length of the individual. Course work is very limited. You guys okay? It's very limited. Trust me. If I'm putting for example, 500 words overall because it's doable in 500 words. So do not complain about the length because I'm not changing the length. And no, do not email me saying can I provide two pages more? No you can't. I'm just anticipating because every year is the same. Trust me, if you know something, if you know the concept, if you know the model, if you know the explanation, you should be able to summarise your main ideas. And we understand that you know. So don't trust me. So trust me. Sorry. Trust me, this is doable. So this is the type of questions are very a very common questions in the final assignment is. So all mathematical models they they have assumptions. They have limitations. You have to make assumptions to be able to develop your mathematical model. Right. So in this question I ask you can you please drop one of the main assumptions of your model, drop this assumption or drop or try to overcome the main limitation of your model. Propose a new model to address this assumption. This is a very, very common question. Okay, it depends on you guys for example. But my problem is just a facility location problem. And I am assuming I don't know, all clients must have their demands 100% match. This is one assumption. What would be a different assumption? Now? The assumption is the clients. They can be fulfilled at any coverage level, for example. So you can propose a partial coverage model in which you introduce, for example, the unmatched decision variable. And you have to penalise very it's very, very, very simple. Everything that the IRS guys should do, it's, trust me, 100% something that we discussed here. So catch up with the course material, catch up with the videos if you want or whatever. This is very important because in 48 hours, let's assume you never catch up with the light. It's doable. I understand that you have your own things you prefer to dedicate your time with, but my I strongly suggest that you start if you if you haven't started yet, to catch up with the course material in 48 hours, it is impossible because you have six. In terms of course content, you have five, five, six sets of slides, and even though some slides are only 2030 pages long slides, there are a lot of things. Sometimes there are links. They are like enclosed materials and chapters. Again, I'm not crazy. I'm very reasonable. Okay, I don't ask you guys to explain a detail of something that I never mentioned. I do not do that. The idea is not to be tricky, you know, to be that crazy assessment. This is not this is a very, very, very fair assessment. If you know the course material, okay. On average, obviously you can struggle with things like tiny things. But overall if you learn how to model how to represent something, trust me, the the final assignments will be a combination of the main problems we studied. For example, obviously there will be something regarding transportation problem because we mentioned transportation problem. How many times? 48 times. If my my mathematical is all right, 48 times in almost six weeks. I mentioned the transportation, probably the main transportation, probably the total transportation problem. Exactly. So obviously obviously the transportation problem will be there. How? I don't know, some discussion, maybe some extension of the transportation problem, a much better transportation problem. So we have to combine the transportation problem with the production planning problem. So transportation problem we've set up just giving you one of the questions okay. It's there. So what you need to know I have to memorise. No you don't you don't need to memorise. You need to know because model building is not okay. I'm seeing and I you replicate eventually. My question is very specific. So you really need to know how to formulate binary problems, how to formulate conservation flow constraints. The basic guys are discussed okay. The basics. This is based on today's lecture. Today's lecture is about solution methods. So will you provide an overview on LP solution methods and MIP solution methods? Obviously there will be something, but we are not giving all the details. I'm not asking you this here. Please solve this problem by hand using simplex. I'm not doing that. Why? Because I'm not explaining that. Okay, simple as that. On the other hand, I can ask you guys to solve a given problem. So you solve whatever you want. You want to use Gum's sex Excel solver. I can ask you in the in the final assignment. So which is help and analyse the shadow prices, analyse the the dual variables or provides the dual of this problem. If I ask for the dual be probably the transportation problem. Why. Because we did the dual. Okay. So that's the point in which I'm very reasonable. Okay guys, you just should really catch up with the course material because it's very helpful. Trust me. It's very helpful to build your project, to learn to be able to use in the future some tools. And don't worry about the coursework if you are prepared, if you know you have at least 30. 3540. I'm kidding, at least. Okay. You're so serious today. Oh my God. Okay, just a joke. You have at least 5060. On average, students have around 58. Between 58 and 62 over the past years. Because, again, people overlook a little bit because it's 48 hours. I can learn it. You can learn something in 48 hours, but you cannot learn all the course material in 48 hours to be able to replicate. And not even I'm not even saying to have your own ideas to replicate things. Eventually there some questions that I'm trying to observe if you can have a good idea, but this is very eventually okay. It's not all the exam based on that. It is clear there is something else you want to ask you about the individual coursework. This is the moment. No. It is clear. Yes. And again, I didn't say exactly how much of the exam. You'll be dedicated to. The first part. The second part. But again, some question about the group project. Yes. In general, go for this question. Propose an extension of your. Are decision making problem or your mathematical model proposed in the group project proposal extension. All my extension was very simple. Are you analyse if you are. If you critically analysed what you propose in a group project, if you understand what an extension is, if you can justify very well and motivate, even though it's a very simple thing, that's the main idea. If you propose something crazy with no justification, probably I would discount several marks. Because the manager is not to propose something very crazy and I be. I question if you really done that by yourself. Because he didn't propose something crazy in a group project out of blue. There is a crazy robust optimisation model. So this is not about proposing crazy and sophisticated things. It's about to make sure you know how to explain things, to motivate things, to justify things. Yes. Good. Yes. I sorry, I just said one confusion. So the first part will be a reflection. Project. No, I don't like this word reflection because for human sciences, reflection is something qualitative. You talk, talk, talk. No it's not. In terms of the project, it's based. I don't know how many questions I would say maybe one or maybe two or maybe one question with several, several parts. This is in general my exam question one item a b c d e. First explain something to solve three proposes some change in constraint number four five. Run again the model five. Explain again the results. Something like that. Regarding the group project again is related to. There is the qualitative part which is obviously guys explaining what you what you've done, for example. But the quantitative part keep in mind this is the most important part of everything. Group project is a quantitative course. You should be able to define and to put in mathematical terms with rigour. Enough. Okay. On. What if? Not applicable to our project. What do you mean? Give you one example. Type of constraint. I know the the questions are very general. Everything is 100% adapted to your group projects. Don't worry, this is not happening. Are the dates that are on the know. They are not correct. I ask how they're like, there's something wrong because theoretically. I. And today they don't have markers just because I forgot my marks. Sorry. So, guys, a it's not a party. I missed you here. So theoretically, I the deadline for the course. The individual assignment is 7th December. No. So you'll be. We list. December 5th and obviously for eight hours. So. Right. And mathematics. Right. Yes. Yeah. So you'll be released and the deadline. I ask Heather to release at one because this is a Tuesday. So it's like to be in class. So we have two hours here. Two hours. I would say 80% of the theoretical exam you should be able to do. If you know the course content and 20% you can do in the remaining 36 46 hours. But the deadline will be. Okay. This is the deadline. The group project. It's wrong as well. Because you remember I told you guys. So when is the workshop? Yes. So this is the workshop. And remember, I told you guys that the idea of the workshop is, are you provide for the feedback because you have everything, you have model results, discussion everything, and you have some time to take into account my feedback if it's the case to further submit. So I want to extend the submission of the obviously don't need to. You can submit on 29 I suppose on learning the date is to. Huh? It says individual is due on 29th and group is due on. Oh my God. It's. Do not I ask her to to change that. Okay, so this is this is done. Okay. I'm just saying that these are. This is not changing. But I like to give you at least a a couple of days. I don't need you, but. So I would you like to put the deadline for. December. This is a Monday. Okay guys, again. But this is on Monday and this is the exam guys. This is on you. You can submit before this. Okay. You can submit next today or two days after or three days after or four days after. I can continue till December 4th. Okay. This is just to give you some time, some flexibility if you need this flexibility okay. Otherwise I could maintain 29. But I don't think this is reasonable. Okay. Are you okay with these deadlines? Yes. Okay, so I actually I sent an email to to Heather before. So they should revise this deadline's all right. Yes. Are we done? Questions? No. 6,040%, 60 for the group and 40. I don't remember. Sorry. It's probably it's described the unlearning assessment. Go for that. I don't remember exactly the wages the group purchase. Bigger percentage? I don't think so. The group is only 30. Seriously? No. I'm not sure about that. To make sure I go to DPS. After that, you go to the rp's. Anyways, if you guys if you see inconsistency, please email me sometimes. Yes, because I don't feed the learning. I just I just put my slides. I don't know what it's happening like, but eventually it can be my mistake obviously. But I don't think so because in general I have the individual. The individual coursework is always has always a bigger wage. So I think it's 6 to 40 individual and project I'm pretty sure. It's not like that. I thought it was. It's just like that. I thought it would. Oh, no, no I can't. No, no. This is regulation. This universe of Edinburgh regulation. Individual components are always. The weight is always larger than group projects for other purposes. As far as I know, it's regulation. Exactly. To avoid being troubles with your if your group, because this happens a lot in the past. So sometimes I mean you are a good student, but something messy happened in terms of group projects, you know what I mean? And in the end of the day, this doesn't reflect you, but individual components is individual. It cannot blame on anyone by you. So that's the main rationale. But maybe this change I don't know guys okay. But anyways guys okay let's let's start then. Yeah okay great. If you have any further questions please email me okay. Guys so today I saw the main ideas I which is low to them. You'll be tired today. But yes the main idea is. The main idea is to provide an overview on linear programming. Okay. The main idea of the linear programming methods, okay. Because we have different ways to solve an LP. But the main rationale, the geometry of those methods in general is very, very similar. And introduction to integer programming methods. And I'm basically talking about branching boundary, which is one of the most famous methods and still very used and widely adopted. You solve integer programming formulations. So we will start by describing a little bit how we can represent graphically an LP. Okay. And I suppose this couple of slides actually I put in lecture 2 or 3 and extra material, I don't know if you went through this material because it made sense for you guys to understand a little bit more about what can happen with a feasible region of a given mathematical optimisation model. Okay, if it's unbounded, if it's infeasible, if it's feasible, and things like that. So we have this this standard optimisation model. We want to maximise a given function subject to a constraint and the domain of the decision variables x. So basically we have a way, a certain methodology to represent this mathematical optimisation method and to find its optimal solution. Okay. It's very simple. But the problem is these methods can be applied to problems involving, for example, two decision variables and a couple of constraints. More decision variables. Even three decision variables can be a little bit difficult to y, because you have to represent in each axis one decision variable. So if you have, like for example in r2 x one and x two, you can represent very well. If you have a three decision variables, you need x, y and z for example. So you b represent the feasible region in AR3. Okay. Space. This can be very complicated to visualise. What if you have four decision variables? You'd be able to represent the feasible region in R4. In four dimensions. You know how to do that. No. Me neither. Nobody knows. Obviously. Oh, you can project. Yes, you can project. You can basically fix always one dimension and you can project the other three. And it's still doable. What about five six? Obviously we have a problem. So the idea of the graphical message or the graphical representation, it just to give us some insights on how to find the optimal solution. But definitely we don't solve large scale problems, large scale problems, as we we have been dealing over the past five, six weeks. We need a solver, commercial solver or you have you can implement your own methods. Obviously this is not the course to implement your own methods. That's why we are using the algorithms provided by guns. Okay. But anyway, so in terms of linear programming representation we have five main steps. First we will represent the constraints through lines. Remember we are solving LP. So the equations the inequalities they are always linear. So they are a line. They are not a polynomial. They are not something crazy okay. After that you have to find the feasible region which is basically the intersection okay, of the areas provided by each line or each constraint. Find the coordinates or the corner points. After that we will we can evaluate each coordinate point. And finally we can choose the optimal solution okay. This is the main aspects. So we have one example here a very simple object function and subject to three constraints okay. In this case is a constraint involving X1X2. In this case this is a lower bound. See this is a very simple lower bound for this constraint. And again this is a very simple lower bound. Basically we have one constraint in two lower bounds plus this domain that basically in this case is completely redundant because we know x one and x two they are obviously greater than zero. So first things first what's the idea. The idea is to represent now these constraints by means of lines. To do that it's very simple. I suppose you already know we transform those constraints in equalities or equations. So x one plus x two equals seven. And we have to represent this line. You know how to do that right. So for example x zero if x one is 0X2 and the same seven zero so zero seven and seven zero okay. So you have two points and you can draw a line okay. This is a anyway it's very simple. So what we have here yes we have here this line. This is guys this is I know this is very silly but okay. We have here the line. The line is important. But now we have to identify which region of this line is feasible for the problem okay. So we have this line. But you have to keep in mind that the constraint actually is x one plus x two less than or equal to seven. So how do you identify if the feasible region is actually below this line or above the line. How do you how do you do that. How do you make this decision. Okay. That's exactly. So you can try for example, this one I always try. 000000. Yeah. Less than great. So if this is feasible obviously. We are interested in all the solutions. Below the line. All right. Yeah. Great. Perfect. This is the yellow region so far. Now we have. This line. This line refers to this constraint x one greater than or equal to one. Same rationale applies. Which region should. To the right or to the left. Perfect. And finally this one, which the second constraint I don't know, it was supposed to be yellow. But anyways, again this one. This is the line x2 greater than 200 is not feasible in this case. So we are talking about this region. This region and this region. So the intersection with this triangle. Yeah. Perfect. So this is the feasible region. Well very simple. Okay. I'm sorry. This is very simple. But just to make sure you know, so this is the feasible region. Well so now the main idea is okay we have the feasible region okay. You have the feasible region. So all the optimal solutions. And you see a very nice theorem in a couple of seconds. We have this theorem in linear program because we have a convex region here. If you have a convex region out here and everything is very well behaved, this is a continuous region. So we know the optimal solution will be. The corner points. Okay. The extreme points of the feasible region. So it's enough should be suffices to look for the corner points. Obviously you can evaluate all corner points and obviously you are maximising. You choose that corner points that solution X1X2. That leads to the maximum okay. The maximum objective function. Another way of of obtaining the optimal solution is draw a line. Passing through the feasible region with the property is all solutions along this line. There you give the same the same value. What? This is what you call. We have two. We have ISO profits and ISO costs. This terms are not important. So the main idea here is just to identify okay the gradient of the objective function and to build parallel lines okay. And after that we identify with the optimisation direction if you're minimising or if you're maximising. So if you're increasing these ISO profits or ISO cost lines or if you are decreasing them okay. By means of this let me show. So this is the first this is the vector. This vector is associated with the objective function. You know how to obtain this vector. This vector actually represents the direction of the optimisation. And there is there is a very simple way in in linear programming to identify this vector. Again because the objective function is linear. So basically this vector here the components of this vector here are the coefficients are the cost coefficients of the objective function are five and three. This is. Five and three. Okay. We five and three because the gradient which is this line, this vector here is the partial derivative of the objective function relates to each component. You know the partial derivative. So we are finding the derivative regarding x one was the derivative of the objective function regarding x one. Five. Yeah this is constant for x one. It's a partial derivative. What's the partial derivative of the second term regarding X23. So five three now is our vector. So we build this vector now and the ISO profit or ISO cost lines be. Perpendicular to this vector, and all these lines should be parallel to each other. What's the idea? The idea is to keep increasing these lines the size of your eyes across lines, until finding the extreme point that is the optimal solution. Okay, the maximum increasing or the maximum decreasing depending, or if you are maximising or if you are minimising a given function. So it would be something like that. For example the first one. And we can evaluate the first one because we know this corner point. This corner point is x one is one and x two is two. Okay. So five times one plus three times two. 11. Right? Yeah. So 11. So this is a profit line value is 11. Here. You can say okay you can stop it here. Can we stop it here. Is this a feasible solution? Huh? Yes yes yes. Yes, this is a feasible solution. Obviously all corner points are feasible solutions and more any. One solution in the size of this region is also feasible. Everything here is feasible. Okay. The thing is, the solutions that are here inside the region are not optimal. The corner points. Eventually we have to find an optimal solution, and this optimal solution be chosen among as the corner points. Okay. So you have here next one. Next one. Next is a profit. It's this one. We know this one. This one is one six. So five times one plus three times six. And the result is 23. So basically now we are comparing the previous solution whose ISO profit is 11 with this one whose ISO property is 23 weeks better. So obviously this corner points is better than one, but both are feasible. We still don't know if this is the optimal one. Let's keep trying to find the other ISO property or ISO cost lines. Finally. Actually I own three in this case okay. So obviously it's pretty straightforward. Now we have five for x one and two for x two again five times five plus three times two right. Yeah. And the value is 31. So 31 is greater than the previous values. And this is optimal. Is this solution feasible as well. Of course to be optimal. And it should be feasible. So this is the corner point the optimal corner point or the optimal extreme point okay. So this is the main methodology. Very very simple. Obviously we have another one. This is not even in the feasible region okay. Just to show that obviously you can keep increasing forever. But after you pass the feasible region there is no way to find anything that is feasible or optimal. Okay. So finally we have here the optimal solution 31 are an optimal objective function value. Sorry 31. And the optimal solution X15 and X22 okay. This is a. A way to represent very simple linear programming formulations. Okay. So now we have I want to I just want to give you an overview of these two theorems here that are very important, just to make sure that we know that the corner points are potentially optimal solutions. So if you have here our region are our yellow region. If this region is a convex polygon okay. Obviously it needs to be convex. Now we know it's a convex polygon. If I visit a region it's a convex polygon for a linear programming problem. And let z equals 2AX plus b y be the object function. Obviously in this case the cost coefficient a and b, which is the the gradient as well the the pair a b be the object function when z, the objective function has an optimum value, maximum or minimum, where the variables x and y are subject to constraints described by linear inequalities, which is x sorry a x less than or equal to b. For example, this optimal value must occur at a corner point or a vertex of the feasible region. Okay. So we have this theorem. So we know if you are solving a linear programming formulation in which all the constraints are given by linear equations or linear inequalities, and the domain of the decision variables is simply greater than or equal to zero. So if this happens so we can make sure the feasible region will be convex and I's if it's a convex and if the objective function is linear. So we know the optimal solution will be found one of the corner points of this feasible region okay. And the second theorem let R be the feasible region for a given LP. And again the same objective function description if R if this feasible region is bounded okay, as it was our case with the yellow region, then the objective function that has both maximum and minimum value of R and each of these occurs at the corner point. So it means that you have for example this. We have this region here we are maximising. What this theorem is saying is we don't care if it's a maximisation or minimisation. Okay. What you're saying that if they're forced, if it's a bounded region, which is this case. So we are saying that this for this region here we can have both a maximum and a minimum. And both values obviously occur at the corner points. Okay. So the main takeaway is you have a linear if you have a given LP and you are changing the objective function, but you are maintaining the same constraints in terms of feasibility, it doesn't change. This is very common okay. But I change the objective function. That's why now the solution is infeasible. No objective function doesn't change feasibility of the problem. Constraints change feasibility of your problem okay, because you can play with the constraints. You can be more strict. You can for example, shrink this feasible region or it can enlarge the feasible region. So this changes this can mess up with the feasibility of the problem. But the objective function no, the objective function will mess with the problems. Optimality okay. With the corner points where the corner points you be the optimal corner points, you'll be okay. Yes. This is the main takeaway okay okay. Great. I suppose. Lecture three I showed some convex regions. So this is a convex region is not the type of region we are dealing here okay. But this is a convex. What about this. In LP. Can we have this type of feasible region? No, this is not convex okay. If you draw a line, for example, between point A and B here, the line will be outside the polygon. So this is not convex convex. What about this. It's convex but not bounded, which can happen. So eventually we can have for example. Oh, eventually we can have a feasible region of this type. Have two constraints here. Sorry, this is one constraint. So we have here, here, here. Okay forever, actually. Okay. This is a feasible region. It's it's a convex region, but is not bounded. Okay is not upper bondage. This this region. Eventually we can have a finite solution for this problem, but eventually, depending on the corner points and the optimisation direction, considering this region eventually can have an unbounded solution, which is the same way of saying no solutions found. Okay. Now I want to discuss very briefly what can happen, but obviously this extends to the case in which we have several variables. Okay. We don't need to represent this problem with several decision variables to understand what happens with a given with a given LP depending on the feasible region okay. Shape. So obviously in that case we have in this current case the yellow region, we have an optimal solution okay. But we can have an infinite number of optimal solutions, what we call alternative or multiple optimal solutions. We can have infeasible and infeasible solution for example. And we can have them bounded which is just described here. Eventually we can have an about solution in terms of. No feasible solution or infeasible solution. So let's assume we are maximising x plus two y and we have two constraints x plus y less than or equal to 1X2 greater than two, and y greater than or equal to zero. So basically the feasible region would be the intersection of these individual regions defined by each constraint. What you can see. So. The intersection set is empty. Perfect. So if this is the case, this means that no feasible solution every time you are in gums or no feasible solution for an LP is something like that. Okay, in general guys, when we get infeasible solutions, it it indicates that something wrong with your model because again it is pointless. Keep this in mind forever. It's pointless to build a model. That can yield an infeasible solution. There is no analysis. You can you can provide based on infeasible solution. That's why sometimes I see okay what about these models feasibility. What about this model feasibility when this model is feasible things like that. It's important for you guys to build a model that is feasible okay. Regardless of the data instance you're using to feed the optimisation, the numerical problem okay. So in this case intersection is empty. No feasible solution. We have a very simple case unbounded solution maximising x objective x greater than or equal to zero. Right. You can keep increasing X as much as you can, and the solution will be won't be optimal because you always can increase a little bit more, a little bit more infinitely. Okay. This is a very common case as well. When you start working and running other problems, eventually you see, oh, but this is unbounded. The solutions are bounded when the solution is unbounded in general means that your optimisation direction is wrong. In general you have to remember this okay. In general, unbounded means. If the constraints are correct, the optimisation is wrong or you forgot to define the domain of your decision variables because for example, if you are minimising overall costs production planning problem, you're minimising costs and you forget to define x which the production cost is greater than or equal to zero. Very simple and you're minimising you're minimising x. X can be anything. So X eventually got you. Minus infinity. And this would be unbounded. So these kind of things are very simple errors that can that can happen. So you should be able to identify and to visualise okay this is about. So maybe there is a problem with optimisation direction is the feasibility problem not it's unbounded is just infeasible. So probably constraints. It's not objective function constraints. No unique multiple solutions. This is very very common. But in general we cannot or we don't have tools or very straightforward tools to identify if you if you have a case of multiple solutions. This happens all the time, especially in when we are dealing with large scale problems. It means that C. This is the our ISO in this case maximising ISO profit lines. Okay. So we are going this direction here. So this is the we are assuming this is the feasible region. So this is parallel. See if this is parallel it means that this ISO line will coincide with. This arc here, this line here. So all this arc is optimal. However, keep in mind you be different solutions, but leading to the same optimal value. Okay, when we say no unique multiple solutions means that different pairs of solutions, but leading to the same optimal value, the optimal value must be the same or otherwise not optimal. Otherwise you can rank and you can find the optimal okay. So in this case. Coincides with the entire line. When you run that code against an impassable local solution. There is a way. In general, we have cases in which the problem and we are not discussing at this characteristic. When the problem is very degenerate, a very degenerate. The generator or the RC is like a characteristic of of model building. Let me. I don't think this purple was supposed to be used here. This is not good. So what? We. So let's assume you have a given region here. And we have. I don't know if you can see it. So we have one line here and you have one line here. How many lines we need. To have a point here to determine a point. How many lines? Two. More than enough. Where the generic problem is, which is very common when you are dealing with this real decision making problems in which you are representing for the first time, I give a decision making problem. This can happen because very the problem is what they have. You have the same corner point being determined by more than two lines, which is each line is one constraint. So eventually we have the same corner point being defined by several constraints. For example. These one and. These one. But every time I add this constraint here, I'm not changing the feasible region. I'm not cutting off the feasible reading. I'm not shrinking the feasible region. So basically what we have here is a lot of redundant constraints. This is very common. So when you identify that your problem has several redundant constraints, maybe your problem is. Degenerate. Okay. And if your problem is the generates the generate, eventually you can have a problems in solving the problem. Because from the point of view of linear programming for example, what's the idea? The idea is we are testing different vertexes, the methods. So we are in one vertex, we jump to another words. It doesn't change anything. The new value is still the same and you keep changing. Vertex is the same. Optimal value is the same value. We don't know if it's optimal yet, so that's the problem. It can be like ages to solve a given problem. Just because you didn't manage to identify before. You have a number of constraints. If you have that, for example, if you have a given LP and it takes ages to give the solution, probably you have in your problem is not a giant. It's manageable. The size is manageable. So problem your problem probably your problem is very degenerate and probably you have a lot of redundant constraints or redundancy and you can get rid of those redundancy constraints. All right. This is one way. But yes Gumbs is not telling you okay. Your problem has multiple optimal multiple optimal solutions. For example it's not showing that. Okay. And what if we have more than two variables. If you have more than two variables okay guys all this graphical solution method is very nice okay. It's very it's very simple to understand. But if you have two decision variables, if we have more than two, we cannot even represent graphically okay. In the real search for example. So obviously we need a way to solve this problem. Then we have these obviously different ways. So we have something here that you already know about where to find the optimal solution. We have this this theorem that you are not stating as theorem, but we call the extreme point theorem. If you have an LP and has a unique optimal solution. So the optimal solution is unique. Optimal solution is at the corner points of the feasible region. Okay. So all of these theorems, the main takeaway is you don't need to try to enumerate all the solutions of a given problem. It's enough to enumerate the corner points of the feasible region. And if you intimidate the corner points of the feasible region, obviously, if the problem has an optimal solution at the very first beginning, you don't know. So you need to try an error. If you use this this method, this enumeration method, you go for all corner points. You identify the solution and you see if it's feasible or if it's optimal or not, and so on and so forth. Obviously this can be crazy difficult. And you have, I don't know, a problem with several variables and several constraints. So you have like a crazy polygon. So obviously this is not reasonable. We don't do that. We don't keep enumerating corner points because we have a more elegant way to identify promising good solutions or potential optimal solutions. Okay. And this is the simplex method okay. Break. Yes. Yeah. Okay. Okay. What do you say? That's very true. We will. I just thought that. So that's probably direction to go. You. All. Right. So. I think we could go for. The. Like. What? Yeah. Yes. That. She was helping. Me, I guess. That was. My fault. What's. That? I. Don't know. I think it would be best. All my love. Yeah, I feel like. I don't know what. I. Was it like. For. The first. That's how. We. Got. Follow me. Because I've been living with the mouse. Because I know that I. Yeah. So. Let's. Take a look. At. What? You. Can. Generally. Thank you. So much. I. Don't. Know. Why? I would. I think I. Lost. You should be able. Over. You. It's. You. I. My. Husband. I think. I'm going to have. Because. You know. I think. Yes, I love that. You. Don't. Need. To. Well, this. No. No, no. I. Know. You want. That's. I. It, doesn't it? Hasn't even been 15 minutes. Oh, yes. Of course. Let's go. And. See? Hello. Let's start. Let's restart. Okay. Okay. Let's go guys, wake up! Hello. Okay. Great. Let's let's restart. Let's finish. So okay just to summarise then for LPs we can draw the feasible region. The feasible region is determined by the constraints which corner point is optimal. It depends on the optimisation direction. We have the idea of drawing as a profit as a cost lines to find this corner point. Or we could evaluate all the corner points. Why we don't do that for real world applications? Because this would be very time consuming. We need like centuries to reach all the corner points. Okay, but the good thing is, we know if there is an optimal solution of some solution would be on the corner points, and based on that in different theorems. Okay, we know that there is a more and more elegant way to obtain the optimal solution which the simplex method. So the simplex method was proposed by this very nice guy Dante, George Bernard Dante. He was proposed. This was when he was born okay. This was when he proposed the simplex method. Okay. 1947 he passed, I suppose some years ago I was, I suppose I was, I was in my, my first year. I suppose it was 2000. 5 or 6, and he work in 94, 95 years, and he was still publishing and supervising students seriously until more than 92 years old. Anyways, this guy is very famous for a reason. He's the the father of linear program, but more than that, he was the first researcher to propose to optimisation under uncertainty and stochastic programs that some of you guys if you if you're changing stochastic optimisation next semester, you learn stochastic programming. And again stochastic was proposed by this guy. So this is really and we are talking about this very big change that happened in 20th century. Simplex method was considered one of the most important algorithms into a 20th century. Are top ten algorithms of the century. Okay. And obviously we have all the heavy Monte Carlo methods, one of those algorithms as well, in the list of the top ten list of most influential and most important algorithms of the of the century. Monte Carlo probably heard of Monte Carlo for simulation. Metropolis algorithm this is one, but simplex is there side by side with Monte Carlo. And it's crazy because I would say that most people, they don't know that even very simple applications and apps, actually they are running a simplex algorithm in the background. This is very, very, very, very common. You have no idea. So many apps and applications and packages that have simplex or simplex based methods, because there is no only one way to implement simplex method, because ultimately it's a solution method, it's a numerical solution method. And obviously the proper implementation, the computer is the difference between having a successful method or a method that will fail depending on your problems characteristics. Okay, but this is done with probably he was my age when he proposed this groundbreaking thing, but there is still time for me whenever. No. Yes. So anyways, it's a very simple idea. Again, the the idea of this lecture is not to give all the details about the simplex every. I changed a little bit three years ago. Yes, I had two class dedicated simplex, but it's a bit pointless, especially in today's world in which you guys are not implementing simplex. There is no way you'll be implementing even if you work with data operations, research or optimisation. We have so many amazing algorithms and solvers and free solvers that you can freely download. For example, we have a very good one if you want you want to try, which is LP solver. The name LP solver is a very good one. In Python we have some free available ones, but they are not good. Okay, but you can try anyways. So that's why I suppose I've chosen to go for this routeing, which at least I give you the intuition underlying LP methods. Okay, so the main idea of the simplex method and again this is very simplified. All right. We have to start from an extreme point. We know that right. We always start from simplex based methods. We always start from an extreme point of the feasible region. These extreme points in simplex terminology is what we call a basic feasible solution. Okay. This is a basic feasible solution because this is related to a base in a given space. And we have to prove this is a base and some things like that. But anyways, we are starting from an extreme point of the feasible region. After having this solution, this corner point, we check whether or not exists, a better solution among the neighbours of the current solution we have like one corner punch of a lot of neighbours. Okay, corner points to which like okay, we do we have better or promising better solutions in other corner points. Yes. So jump to these next solution that can give you a better optimal value. No, we don't have any. So you are in the optimal solution okay. So very very simple. So basically let's assume we have this polygon here in a given space. Assuming this is a crazy projection. We start for example from this corner point and we have it different potential corner points or solutions. The idea is we don't have any promise of a better solution in any of these corner points. Okay. So this corner point is the optimal one. Great. The algorithm just stops. We have a potentially a better solution. So we go to the next solution. And again we check if the neighbours of the next solution. One of these neighbours at least she has a better solution. If not this is the optimal solution. Otherwise you jump again and jump again and you jump again. And this is the main idea. Okay, but as I said, to give you all the details, we need to. If you read the textbooks and the chapters that I made in closed Unlearn, you see that the first, first step is to basically to find the standard form of a given. Are of a given optimisation model. After finding a standard form, the standard form is always trying to. No, this is not. For example. So if you have this, for example, this optimisation problem, the main idea. The main idea of finding a base or a basic feasible solution is try to put this formulation here into its what we call a standard form. Even though the name is standard form. Different textbooks, they present different standard forms. Okay. The main idea is to transform all the. Inequalities into equations. So this is standard. But eventually the idea of the terminology can be a little bit different. So for example sorry this was less than or equal to three. To transform that into an equation what we have to do because we have like x one plus 2X2 less than or equal to three. Right. But I want to transform that in a equation. So to transform this an equation what I need to do. Okay. And I suppose. Close to three. We needed to equalise and to equalise. In this case we need to add. I knew decision variable. Right. So for example plus x three. For. For this constraint, this constraint was greater than no. I forgot to know what they put. Anyway, let's assume this is greater than or equal to. If it is greater than or equal what we do here. Ops. Okay and how to determine the are the basic solution to start with. In general, we. Consider all the decision variables zero, but the new ones. So this is the first way to determine a solution. So for example X10X20. So if it happens x three is three. Yeah. This is not. This is not good. Yes. I want you to work with. Otherwise would be infeasible anyways from the start. So let's assume this is not the problem. And in this case x for. Okay, so we start. This is the initial. Basic solution. Obviously this is. There are no coefficients for X3X4. So it's zero. So these would be the first problem to solve. Obviously have so many variations. In this case we found a feasible solution because anyways I changed a little bit eventually this is the case. So we have a phase one phase two of the simplex. Phase one. It's just to try to find a feasible solution to start the method okay. So this is the initial basic solution. Let's try to reflect now on the graphical representation of this solution where to find this. So each basic solution must correspond to a corner point of this feasible region. Can you see that? So which corner point is the solution? Does the solution refer to? Zero zero. So when we have when we have our graphical solution representation, we don't care about x three and x four. As such we still are in our 2X1 and x two. So this the corner point is zero zero and the corner point actually is zero zero. Three four. This is the solution. So we start the method from this basic solution. Okay. The next iteration the question would be do we have a better solution? If you have a better solution, there will be a way to find whether or not there is a better solution. If there is. We will move to the next iteration of the method, and probably there is a better solution because this solution is zero. The zero. The coefficients of x3 index for are zero, so this is zero. Obviously this is a rubbish solution okay, so I don't know if this is even feasible. Okay guys I just put any numbers. But considering there is a feasible solution for this problem, consider the feasibility region is no empty. There is an optimal coordinate point. And we have you have the the solutions. Okay. Basically this is how the method two works. Very very very roughly okay. Questions. Of course. No, I did not provide even the details. But you don't need to know all the details, okay? You need to know more details about these methods and will give some more details of this method. Okay, so this is linear programming. So let's assume now we don't have a linear programming. We have an integer or a mixed integer program that is at least some decision variables. They are either integer or binary. If this is the case. We cannot use a simplex. Why? Why you cannot use a simplex method if you have integer or binary decision variables. I don't know if it's. Okay. We cannot define that. The variable. Well. First, because simplex was not designed to deal with integer requirements of the decision variables, because again, we need to have a lot of theorems regarding convexity and continuity of the feasible region, which is not true when we have our discrete problems to remember the discrete problems, the feasible region. Obviously we you have a polygon, but inside the polygon what we have. Referring to the optimal, I'd say corner points eventually, but integer. So simplex cannot be used to solve integer programming formulations. That's why years ago, after simplex was proposed, a group of researchers proposed this method called branching bounds, whose main idea is a very simple idea. The main idea is if you have a given MIP, mixed integer or integer program formulation, you start by solving its corresponding LP relaxation. Do you remember LP relaxation? Yes. What is a relaxation? Is the MIP formulation when we drop the integer requirements of all decision variables. So you have the linear relaxation. After defining the linear relaxation, the idea is to update lower and upper bounds. That's why we discussed lower and upper bounds. After that, you start updating lower and upper bounds and try to impose some constraints to make sure the decision variables in the end you be integer. This is the main idea is a very simple idea as well. Anyways. In general are methods for solving integer or mixed integer programs. They fall into the three main categories. The branching boundary is what we call enumeration algorithms. Enumeration. Because the main idea is to enumerate potential solutions of the methods. But again, this is an intelligent enumeration. We are not exhaustive enumerating all the potential solutions, but we are doing that in a very intelligent way. We have a cutting plan, algorithms that manage to use simplex, and it's derived from the simplex. And the idea is to include constraints as well to in attempt to shrink or to make the feasible region is smaller, and thus to trying to find the integer or solution of the problem. And we have hybrids. Hybrid methods are very common. Most commercial solvers who are using simplex are using to solve for MIPs. They use actually a hybrid approach. The hybrid approach, the most famous one is branch and catch. So we have the branching bounds and iterative really. We try to enforce the integer requirements by means of extra constraints. Okay. But let's focus so far in at now branch on branch inbound. So branch inbound is a solution. And the name is basically it's the manager is branching boundary to divide and conquer. The main idea is if you have a given feasible region that refers to your integer programming or mixed integer programming formulation is to divide into more manageable subdivisions or what we call sub problems, and the idea is to solve these problems iteratively until to get the optimal solution or a given stopping criteria okay, of the method of the algorithm. The partitioning process of the solution space is what we call as branching process. Okay. In order to avoid unnecessary branching, abounding scheme is usage. That's why the name is branching bound. And we'll see like one example, there are different ways to build branching bound trees. Branching algorithms. Sometimes you have specific algorithms for problems that have specific characteristics. For example the knapsack problem. We have branching bound specifically for knapsack problems or specific for transportation problems that exhibit a given special structure when the matrix is not totally modular, for example, and we have different versions, this was proposed. The branching boundary was proposed in 1960 by these two very nice ladies, Lynch and Deutsch. And basically since the since the 1960 branching boundary is to be used in two different algorithms to try to find the optimal solution or of more complex problems. Okay, I suppose one of these ladies she passed years ago, but one is I don't remember which one. One is in Melbourne. Do you remember? No. You never, never heard of these ladies? Anyways, now you know they are a little bit different from this. Probably. Well, obviously one is that anyways. Anyway, so let's start with the message in a very simple way. So let's assume you have this example. We have this is our IP. We want to solve a big function 3X1 plus 4X22 constraints less than or equal to. And how you know this is an IP because x one and x two are created equal to zero and integer as well. This is the original IP we want to solve. Okay. First thing to be able to use branching bounds to define the linear relaxation of this IP. The linear relaxation basically is. Same problem, same object function, same constraints. However. The integer requirement is. Drop it. Okay, we scrap the linear collection. If you scrap the linear relaxation. What we have here an LP. Yeah. If you have an LP we know how to solve. That was the rationale obviously in this x two because we know there is the simplex. Simplex is very efficient to solve most problems. So great we scrapped the integer requirement. And we can solve the problem using linear programming. Okay. So. This first what we call LP zero. I put LP zero. This is the linear relaxation okay. The linear accession. It's all also called root node problem okay. Why root node. Because the main edge of the branch is to build. It's tribute at three. So this is their wood node. After having this solution of the root node, we start building our branch and bound tree. The idea of building a branch in boundary Tree is to understand how to proceed with the branching scheme, and how to proceed with the bounding scheme and this branching. The main idea of bringing a given node to find the solution is exactly to enforce some constraints. In attempt to make the fractional solution of the linear luxation problem an integer one. Okay. To find an integer solution out of the LP problem okay, so the idea in the end of the day is we keep solving LP s is a number of LP. That's why I told you remember I told you in general is more time consuming to solve MIP problem than an LP. Because LP if this is an LP, I solve. Once. Optimal solution. It's one resolution. Obviously different variations, but one resolution of one LP. But an MIP or an IP, it's different to a number of a number of different runs, because I need you to solve a number of LPs at each iteration of the message. So that's the main difference okay. We see that. So okay we solve this linear relaxation problem. This is the optimal solution okay. Absolute solution is 2.25 and 1.5. And this is the object function 12.75 okay. This is feasible for the original problem. It's not. Why? It's fractional. It's fractional. Eventually. Let's assume. We solve, it's integer. Done. This is the solution. Simple like that. If you're unsure. Okay, maybe. Okay, I don't know. Find the linear relaxation. So the solution is option. Great. If the solution. Sorry. If the solution is integer. It's done. You don't need to proceed with the in biology because there is nothing to branch. There is something to branch. Until we have fractional solutions where they don't have fractional solutions anymore, it's done. Okay. So that's the main idea of the of the method. So okay in this case we will obtain this fractional solution to 2.5 and 1.5. And the optimal value 12.75. What can we expect in terms of the optimal solution of the original problem? Regarding this value here. So do you think if when we solve the IP. To optimality. Do you think the optimal solution of this IP less than or equal to, equal to or greater than or equal to. Less than. Less than why? Because they're maximising. And in this case, the LP relaxation gives us an upper bound. Okay. Obviously we cannot have fractional values here. We can analyse that all coefficients are integer in this case. So best case scenario you have an optimal value of. Perfect. Okay, exactly. So we know that because of integrity constraints. But anyways, we start with the healthy relaxation and we are initialising our algorithm here. We will initialise the lower bound on the optimal value of the original problem to minus infinity. Why? Because we don't have a lower bounds so far. What will be a lower bound for this problem? A lower bound for this problem, considering we are maximising first, is a feasible, feasible solution. A feasible solution must be. In this case integer. So the first integer solution we find when we are branching and bounding okay. Our problem will give us the upper bound finite upper bound because we have this but is minus infinity which is pointless, right? For us this is pointless. It's obvious. In this case we could we could set the lower bound to. Zero. This is the general case. But in this case, we know, right? We know it will be zero, but it's pointless for us at this point. So the main idea is when we find it really a better solution. So at this point on we can update the lower bound. All right. Lower bound was set and upper bound was set to 12 0.75 which is the optimal value of the LP relaxation okay. So now we have to choose okay. The solution is we have two potential branches to choose from. What the age of branching. Then the age of branching is to choose when. This is the basic version of principles. Okay, we have more complicated ones, but the main idea is we have to choose one variable to start the branching scheme. So we can start with x one or x two. Okay. Was the idea of branching include further constraints to avoid this fractional solution. Here, because we know the optimal solution should be integer. So the idea is to avoid, for example, having a fractional solution of fractional value. Okay, both in this case have fractional and fractional part right points 25 and point five. So both are candidates to be branded. All right. We just have to choose the one. It's very common to choose. Are the two. Choose first to branch a decision variable that is further for having an integral value. In this case, 25 is closer to. Zero. For example, 0.25, then 0.5 0.5 is exactly in the middle of the interval. Okay. So that's the main idea then. So keep in mind. So what we have here. Zero. Oh, no. No, I want to say that. Okay? Yes, I want you to do that. Okay. We have here. This is the integer values. Okay. Our solution is here. What we want to do. We want to avoid this fractional solution. How to avoid this fractional solution by means of constraints. No. No. You're not surrounding anything. This is not around procedure. But. Yeah. Bear in mind we have run the procedure, but not not here. Eventually. How to how to impose constraints. C 1.5 is in the middle of one and two. So if we impose the constraint x two this is extra. Is less than or equal to one. Okay. Remember this is integer programming. So considering this interval zero one and two we are not increasing anything in the middle of the interval. We are interested in either zero or 1 or 2. Right. So when you do x two less than or equal to one we are not excluding any feasible solution of the original problem. So less than or equal to. So we are saying. Obviously anything here. Okay. Or or. X two. Greater than. Perfect because we are not excluding anything. We are dealing with this interval. What we are avoiding exactly. Fractional solutions between 1 and 2 such as this 11.5. So this is the main idea of the method. So for each variable fractional variable the idea is to identify. These constraints, these bounds on the on the decision variables such that in the end, you have so many of those bounds that the solution will be either infeasible or integer feasible optimal. Okay. So this is the main idea. What we have here LP one. LP to. See this is a tree. So this is the root obviously upside down. But yeah this is what we call the branching tree root node LP relaxation. First node for this first node. Will be basically what we want to solve here. The linear relaxation because again we only solve LP. So this is the original LP plus. This constraint. Okay. This LP will be the original Linearisation plus. This constraint. Oh, can I put these two constraints together? This is a question. Can we. Of course you know why. Because either or. Is true. We cannot have x two satisfies both constraints and alternatively this is the intersection of this constraint is. Zero. The set is empty okay. So that's why we define it this LP and this LP and we solve again. So let's do it. This LP. The original LP will.