Distributed Optimization and Machine Learning

Prof. Mayank Baranwal

Computer Science & Engineering, Electrical Engineering, Mathematics

Indian Institute of Technology Bombay

Week-9

Lecture 33: Algorithm for Uncapacitated EDP

So, essentially, we need to ensure two things, we need to ensure that there is consensus on lambda, there is consensus on lambda, because this particular lambda is going to be shared between different generators. generators have the dispatch value is going to be this particular quantity right. So, we are going to be ensuring these two things and this basically gives us our algorithm and we are going to then look at the proof of the algorithm to show that this algorithm indeed converges in a fixed time ok. So, algorithm for uncapacitated economic dispatch. So, before we before I specify the algorithm, we need to keep a couple of things in mind. So, one thing is we need to run consensus on lambda.

and the other thing is P i is going to be ok. So, we need to ensure this ok. So, this basically gives us the algorithm which is let me first write down the algorithm. So, e generator i it would be running So, pi dot t is going to be given by.

So, let me write this down first and then we will. So, every generator runs this algorithm. So, what is this algorithm? So, essentially pi dot. So, it is going to be updating its expected dispatch value. So, first of all because we are going to be running consensus on lambda, we do not know lambda to start with.

Every generator runs this algorithm

$$\begin{cases} \dot{P}_i(t) = \sum_{j \in N_i} \left[\operatorname{sgn}^{\mu_1}(\lambda_j(t) - \lambda_i(t)) + \operatorname{sgn}^{\mu_2}(\lambda_j(t) - \lambda_i(t)) \right] \\ \frac{\dot{\lambda}_i(t)}{2\alpha_i} = \dot{P}_i(t) + \left[\operatorname{sgn}^{\nu_1}\left(P_i - \frac{\lambda_i - \beta_i}{2\alpha_i}\right) + \operatorname{sgn}^{\nu_2}\left(P_i - \frac{\lambda_i - \beta_i}{2\alpha_i}\right) \right] \end{cases}$$

So, everyone has their own copy of lambda and correspondingly they will have their own, anyway they will have their own copy of the dispatch value pi. So, every generator is going to run this algorithm. So, they are going to be updating pi dot using this expression over here, which we know is very similar to the consensus algorithm. So, if I look at the fixed time consensus algorithm x i dot in terms of these quantities. I mean there is a negative here because it is x i minus x j, there I have written it as x j minus x i.

So, you can I mean negative sign is not there, but the right-hand side of this basically ensures consensus on x i's right. So, the same algorithm I mean similar kind of consensus algorithm is what I am using over here. Now, for the lambda variables right. lambda is essentially when what happens if this particular term is equal to 0. So, that means lambda i dot over 2 alpha i is same as p i dot and if I look at this expression that is what we have p i dot is same as lambda i dot over 2 alpha i because beta is constant.

$$\sum \dot{P}_i(t) = 0 \Rightarrow \sum P_i(t) = \sum_i P_i(0) = P_{tot}$$

So, if this is equal if this is guaranteed. So, then that means this is going to be satisfied right and if there is consensus on lambda that means pi is also pi dot becomes 0 ok. So, therefore we have solved the problem right. Is everyone with me on this? So, essentially it is a fixed time consensus kind of scheme both on pi as well as lambda both on this quantity because we want to ensure that everyone goes to. So, this is not a fixed time consensus in the sense that you are not exchanging information with neighbor, but you are ensuring that this I mean this quantity on the right hand side here, this converges to pi in a fixed time and this is what is happening in this expression.

There is no exchange of information with the neighbors by the way, there is no summation j and i. So, what I am saying is that in a fixed time, I want to ensure that this condition is satisfied. And once this condition is satisfied, then pi dot is essentially lambda i dot over 2 alpha i which is what I want. And once that is true, on top of it, if there is consensus on lambda, then pi dot also becomes 0. And once pi dot becomes 0, that means we have essentially solved the problem.

I mean the values of conversion, we have solved the problem. So quick few things note here is what is summation pi dot at any time? This is 0 right because it is an odd function. So if I sum it over from y 1 through n and j again 1 through n and aij and so on. I mean that is something that we had already shown in previous lecture. So this is going to be 0.

So this implies summation pi t is always going to be summation pi 0 which is going to be So, that means, the whatever consensus happens on lambda, it would be corresponding to this p total. And therefore, and it is not just any other random consensus quantity that it is basically we are running the consensus on and therefore, this scheme is going to work. So, now we need to show like basically prove that this entire scheme converges in a fixed time and in order to show that we are first going to show that this converges in a time let us say fixed time T1. It takes T1 time to ensure that this happens. So, once this happens then we would show that in another time T2 which is greater than T1 this scheme would also converge.

the top one and then in total time T 1 plus T 2 when you would have the convergence of the entire algorithm. Is it? Yeah, so you are going to be updating two variables at a time. So, let us look at the first one. By the way, what does fixed time convergence result says? Like if you have the Lyapunov inequality satisfied, then there exists a capital time T, some settling time capital T such that trajectories converge before that settling time and then they stay converged for all future times. So, if there is a fixed time convergence that happens here.

So, not only we say that pi at any like after some capital time T or at times capital T1, this is not just pi is equal to lambda i minus beta over 2 alpha i, but it stays converged for all future times. So, the moment this convergence happens, it is not like when you run the algorithm, you are not going to remove this algorithm at all. I mean, because you want this to stay converged for all future times. So, you will always be updating, even though this may have happened, you would always be updating your lambda i dot over 2 alpha i, you will always be running this algorithm. because you want to stay converged for all future times as well and not just instantaneously.

So, this basically scheme which runs on both pi and lambda i this needs to be run the entire time till you get consensus on lambda i or till you get your pi's till you happen to have your pi's to converge to a common value or not a common value but to a constant for a given p total. Second is trying to yeah second is trying to ensure this this optimality constraint. So, this is the constraint at optimality right and second second one is trying to in fact in this algorithm what I mean what happens is this this constraint satisfied first followed by the consensus. So, we are going to show we are first going to show that the second ODE equilibrium is reached in a fixed time.

ok. And in order to see this, let me define the error e i which is going to be the error between p i minus ok. Is this clear? Right. So, what is e i dot? It is p i dot minus lambda i dot over 2 alpha and if I look at p i dot minus lambda i dot over 2 alpha i that is nothing but minus of this minus of this quantity right. So, which is going to be negative. and what is the term inside this bracket? It is a ei.

$$e_i = p_i - \left(\frac{\lambda_i - \beta_i}{2\alpha_i}\right)$$

$$\dot{e}_i = \dot{p}_i - \frac{\dot{\lambda}_i}{2\alpha_i}$$

$$= -\left[\operatorname{sgn}^{\nu_1}\left(p_i - \frac{\lambda_i - \beta_i}{2\alpha_i}\right) + \operatorname{sgn}^{\nu_2}\left(p_i - \frac{\lambda_i - \beta_i}{2\alpha_i}\right)\right]$$

So, what do we get? ei dot is negative. Is this clear? So, now I have to ensure that this ei goes to 0. and in fact not just this e for a particular i, but for all it is right. So, I need to define a Lyapunov function V which looks something like this. And if this Lyapunov

$$V = \frac{1}{2}\sum_{i=1}^{N} e_i^2$$

$$\dot{V} = \sum_{i=1}^{N} e_i \cdot \dot{e}_i = -\sum_{i=1}^{N} e_i \, \mathrm{sgn}^{\nu_1}(e_i) - \sum_{i=1}^{N} e_i \cdot \mathrm{sgn}^{\nu_2}(e_i)$$

function is 0 only when all e i's are 0, otherwise it is going to be positive definite.

So, in order to show that this converges in a fixed time, we need to show that V dot satisfies this inequality that v dot is less than equal to some c 1 V to the alpha 1 minus c 2 V to the alpha 2 something like that right. So, let us take V dot here and this gives you e i e i dot and that is going to be. And what was the definition of the signum function? This new or this signum new kind of this funny looking function. So, if I this is sign like signum mu x is essentially x times this quantity right. So, if I multiply if I include if I put let us say substitute ei for x here.

So, signum ei ei. So, this would be e i square times nu norm of e i. So, e i square I can write this as e i square times e i nu minus 1 or nu minus 1 rather. So, this becomes nu plus 1.

$$\dot{V} = -\sum_{i=1}^{N} |e_i|^{1+\nu_1} - \sum_{i=1}^{N} |e_i|^{1+\nu_2}$$

$$= -\sum_{i=1}^{N} (e_i^2)^{\frac{1+\nu_1}{2}} - \sum_{i=1}^{N} (e_i^2)^{\frac{1+\nu_2}{2}}$$

$\nu_1 \in (0,1)$ $\quad \in (0,1)$ $\qquad \nu_2 > 1$

$\Rightarrow \frac{1+\nu_2}{2} > 1$

e i is a scalar. So, here, but then you would still have absolute value instead of. So, this becomes V dot is okay. Is this clear? Now, I can write this as Let us say I would rather want to write it in terms of ei square, because I want to recover my original Lyapunov function. So, this would be i square 1 plus nu 1 by 2 and minus. So, I am almost there except that the summation is outside instead of inside.

Ideally I would want to write this as summation i 1 through n e i square 1 plus nu 1 this thing and so on. So, if you assume let us say nu 1 was a number between 0 and 1. So, then 1 plus nu 1. So, this quantity is also a number between 0 and 1.

This also belongs. And if nu 2 is greater than 1, this implies 1 plus nu 2 by 2, this is also greater than So, remember if we had looked at one particular result that if z i's are positive, z i is greater than 0, then summation i 1 through n, z i to the power p, this is greater than equal to, if p is a number between 0 and 1. if p is more than 1, then this basically becomes N 1 minus p, if p is greater than 1 right. So, we had already looked at this particular use this particular result in the consensus case. So, we are going to use the same thing over here. Now, this e i square is here, e i square is like your z i right, which are positive numbers and in this case p the power p is basically the exponent p is between 0 and 1.

$$z_i > 0$$

$$\begin{cases} \sum_{i=1}^{N} z_i^{p} \geq \left(\sum_{i=1}^{N} z_i\right)^{p} & \text{if } p \in (0,1) \\ \sum_{i=1}^{N} z_i^{p} \geq N^{1-p} \left(\sum_{i=1}^{N} z_i\right)^{p} & \text{if } p > 1 \end{cases}$$

So, you can use this to essentially approximate it with respect to this and in this case exponent is more than 1. So, you would be able to use this particular result to look at this case right. So, this implies V dot is less than equal to ok. And what is this summation ei square? This is 2 times your Lyapunov function right 2 times V right. So, you get V dot is less than equal to negative 2 V 1 plus mu 1 by 2 minus and this is now you have brought this into the familiar way.

$$\Rightarrow \dot{V} \leq -\left(\sum_{i=1}^{N} e_i^2\right)^{\frac{1+\nu_1}{2}} - \left(\sum_{i=1}^{N} e_i^2\right)^{\frac{1+\nu_2}{2}} N^{1-\frac{1+\nu_2}{2}}$$

$$\Rightarrow \boxed{\dot{V} \leq -(2V)^{\frac{1+\nu_1}{2}} - N^{\frac{1-\nu_2}{2}} (2V)^{\frac{1+\nu_2}{2}}}$$

So, V dot is less than equal to some c 1 alpha V V to the alpha 1 minus c 2 V to the alpha 2 and therefore, you can guarantee convergence in fixed time ok. So, that means there exists some time T1 which is finite such that second ODE converges to its equilibrium in a fixed timeline t less than equal to T1. So, that means after time T1 we know for sure that this constraint is always satisfied or EI is always 0. Is this clear? So, we know that after time capital T1, so this term becomes 0, this term becomes 0 and this derivative is equal to this particular derivative.

So, that is all that we know so far. So, what I can write then is this implies that after time T1, we have lambda i dot over 2 alpha i is equal to, is this clear? Why? Because pi dot is equal to lambda i dot over 2 alpha i after time capital T 1. So, I can write this. So, this looks like a very familiar consensus update on lambda. The only difference is it comes with this coefficient 1 over 2 alpha i which is something that we have not seen. So, we are very quickly going to use the same results, but this time with 2 alpha i.

After time T_1, we have

$$\frac{\dot{\lambda}_i}{2\alpha_i} = -\sum_{j \in N_i} \left[\mathrm{sgn}^{\mu_1}(\lambda_i - \lambda_j) + \mathrm{sgn}^{\mu_2}(\lambda_i - \lambda_j) \right]$$

So, is everyone with me on this that after time T 1, this is what we are going to obtain. So this is pretty much consensus on lambda, something that we had already seen in the previous lecture where we ran consensus on x size right. The only difference is this coefficient 1 over 2 alpha i. So therefore, because of this coefficient 1 over 2 alpha i, so earlier when we did not have this 2 alpha i. what was the consensus value average of the initial ones right.

In this case $\lambda_c = \frac{\Gamma}{N} \sum_{i=1}^{N} \frac{\lambda_i}{2\alpha_i}$

where, $\Gamma = \frac{1}{\sum_{i=1}^{N} \frac{1}{2\alpha_i}}$

So, now it is not going to be the average of this it is in fact. So, in this case so we define this lambda consensus lambda c that is essentially going to be gamma over n times summation 2 alpha i where gamma is okay. So, this is the this. So, instead of converting to the average, it converges to this particular average of lambda is this weighted average, why because, because of this 1 over 2 alpha. And how do we show fixed-time consensus on this or at least convert this to the form that we are familiar with

$$\tilde{\lambda}_i = \lambda_i - \lambda_c$$

$$\dot{\tilde{\lambda}}_i = \dot{\lambda}_i$$

$$V = \frac{1}{2} \sum_{i=1}^{N} \frac{\tilde{\lambda}_i^2}{2\alpha_i} \longrightarrow \dot{V} \le -c_1 V^{\frac{1+\mu_1}{2}} - c_2 V^{\frac{1+\mu_2}{2}}$$

So, we are going to be defining lambda tilde i which is going to be lambda i minus this lambda c. So, we know that lambda tilde i dot is going to be lambda i dot right that is going to be there for sure. and you can if you just sum it over with 1 over 2 alpha included, you would show that this is going to be equal to 0. So, this is in fact this is the value that it would converge to. The difference lambda i minus lambda j is going to be same as lambda tilde i minus lambda tilde j and therefore, you will be able to convert this and the Lyapunov function that you need to choose in this case is that would be lambda tilde i square over 2 alpha i. So, with this choice of Lyapunov function, you would be able to follow the proof as it is as we had used in previous lectures and you would be able to show that v dot is less than equal to some c 1 V to the 1 plus mu 1 by 2 and therefore, this consensus would happen in a fixed time. So, that this means in a fixed time you are able to. So, let us say this happens in capital T2 time. So, in time T1 plus capital T2 because this scheme is valid only after time T1 right.

Only after time T1 this scheme is valid and let us say it takes capital T2 time to converge. So, in this total time T1 plus T2 this algorithm converges. This entire algorithm converges that means you are able to solve the uncapacitated economic dispatch problem in this in a distributed manner in a time capital T1 plus T2. So, this lambda is also called I mean it is a dual variable or the Lagrange multiplier, but I mean we have already looked at the interpretation in previous lectures right. It is also called incremental cost variable and there is also popular algorithm called incremental cost consensus algorithm or the ICC algorithm, which is a discrete-time algorithm that is often used to solve uncapacitated economic dispatch problem.

and this is the fixed-time variant of that discrete algorithm. If you have a discrete algorithm, as I said in continuous time you can design much faster algorithms using this novel insights and this is what we have achieved using fixed-time stability. So, lambda is also called incremental cost variable. And there is a popular discrete time algorithm or discretized algorithm known as incremental cost consensus or ICC used to solve uncapacitated EDP. Thank you.