Distributed Optimization and Machine Learning

Prof. Mayank Baranwal

Computer Science & Engineering, Electrical Engineering, Mathematics

Indian Institute of Technology Bombay

Week-9

Lecture 34: Capacitated EDP

So, let us try and see how we can solve the capacitated or the more advanced version which is the capacitated economic dispatch problem. So, now we are going to focus on capacitated EDP And again we are going to be solving it in a distributed manner. So, all the previous constraints still apply that you cannot exchange your private cost coefficient with your neighbors and so on. So, let me rewrite the problem statement or the optimization problem. By the way gamma i as you would have noticed it did not have any role to play in the previous algorithm. The reason being gamma i are just fixed numbers right.

$$\min_{\{P_i\}} \sum_{i=1}^{N} \alpha_i P_i^2 + \beta_i P_i + \gamma_i$$

$$\text{s.t.} \quad \sum_{i=1}^{N} P_i = P_{tot}$$

$$\text{and} \quad \left.\begin{array}{l} P_i \leq P_{i,max} \\ P_i \geq P_{i,min} \end{array}\right. \quad \forall i \in \{1, \ldots, N\}$$

So, even if I remove this gamma i from the objective function it would not have any role So, think of gamma as the starting cost of the generator let us say you are starting the generator you would have to incur certain fixed starting cost and that is the starting cost and that and then depending on how much power you are generating then you are you have a dynamic cost to it or the operational cost. So, how do we, so again this is a constrained optimization problem and we would want to convert this to an unconstrained optimization problem using Lagrangian. So, this time we are going to include, I mean obviously some pi's are going to be your primal variables, lambda which is corresponding to this equality constraint that is going to be one of the dual variables and then you have let us say ri and si which are dual variables corresponding to this inequality constraint. and this is given as .

If you want to yes you. Yeah, yeah ok. So, this is the Lagrangian for this particular problem and obviously, you have ri and si these are greater than equal to 0. Are there any other constraints? So, when we try to find the optimality constraints, one is the derivative with respect to pi that should be equal to 0, right. So, this basically gives you 2 alpha i Pi plus beta i minus lambda plus r i minus s i, this is equal to 0, right.

$$L\left(\{P_i\}, \lambda, \{r_i\}, \{s_i\}\right) = \sum_{i=1}^{N} \alpha_i P_i^2 + \beta_i P_i + \gamma_i + \lambda\left(P_{tot} - \sum_{i=1}^{N} P_i\right)$$
$$+ \sum_{i=1}^{N} r_i \left(P_i - P_{i,max}\right)$$
$$+ \sum_{i=1}^{N} s_i \left(P_{i,min} - P_i\right) \quad \Big\} \; r_i, s_i \geq 0$$

or lambda is equal to 2 alpha i P i plus beta i plus r i minus s i ok. So, what else do we have in terms of the I mean one is the primal feasibility which is this. There is also dual feasibility and there is also complementary slackness right. And what does complementary slackness gives us here? So, if I look at the complementary slackness from there I know that ri times pi minus pi max this is equal to 0 for every i and likewise si times pi min minus pi this is also equal to 0 for every i right. if let us say pi happens to be a number let us say I solve this problem and the generator dispatch value it is basically sandwiched between pi min and pi max.

$$\frac{\partial L}{\partial P_i} = 0 \;\Rightarrow\; 2\alpha_i P_i + \beta_i - \lambda + r_i - s_i = 0$$
$$\lambda = 2\alpha_i P_i + \beta_i + r_i - s_i$$

Complementary slackness: $r_i\left(P_i - P_{i,max}\right) = 0 \;\; \forall i$

$$s_i\left(P_{i,min} - P_i\right) = 0 \;\; \forall i$$

So, then these complementary slackness conditions will only be satisfied if ri is equal to 0 and si is equal to 0. So, that means you recover the familiar lambda is equal to 2 alpha i pi plus beta i. Now, let us say for one of the generators for the ith generator, the value is saturated at the lower limit which is pi main. So, in that case si is going to be non-zero, ri is going to be 0, si is going to be non-zero because this is equal to 0. So, if si is going to be non-zero, so that means lambda is going to be less than 2 alpha i pi plus beta i, because si is greater than equal to 0.

If $P_i = P_{i,min}$, then

$$\lambda < 2\alpha_i P_i + \beta_i$$

and if $P_i = P_{i,max}$, then

$$\lambda > 2\alpha_i P_i + \beta_i$$

if pi is equal to pi min then we have lambda less than 2 alpha i pi plus beta i and if pi happens to be pi max that means it is saturated at the upper upper limit ok. So, then lambda happens to be greater than 2 alpha i plus 2 alpha i pi plus beta i right why because if lambda pi is equal to pi max that means r i is non-zero s i is going to be 0 and if r i is non-zero then lambda value that is going to be greater than this right. So, these are the. So, essentially if I look at lambda this is equal to 2 alpha i pi plus beta i pi lies between strictly lies between pi min and pi max. this is greater than 2 alpha i pi plus beta i if pi is equal to pi max and it is less than 2 alpha i pi plus beta i if pi is equal to pi max.

$$\lambda = \begin{cases} 2\alpha_i P_i + \beta_i & \text{if } P_i \in (P_{i,min}, P_{i,max}) \\ > 2\alpha_i P_i + \beta_i & \text{if } P_i = P_{i,max} \\ < 2\alpha_i P_i + \beta_i & \text{if } P_i = P_{i,min} \end{cases}$$

Is this clear? In fact, this is what we are going to leverage to design is design a fixed-time convergent algorithm for the capacitated economic dispatch problem. So, let me also give you the, let me call it lambda star. How do you get something outside this? It is capacitated, it is saturated between the two, right? No, I am saying that like the solution, this is what the solution of would look like. we have not come to the algorithm part yet right. This is what the solution, this is how we characterize the solution.

So, the value of lambda star is going to be less than this, greater than this or equal to this

Θ: set of generators for which inequality constraints are active. $P_i^* = P_{i,min}$ or $P_i^* = P_{i,max}$

$$\boxed{\lambda^*_{cap} = \lambda^*_{un} + \frac{\sum_{i \in \Theta} \left(\frac{\lambda^*_{un} - 2\alpha_i P_i - \beta_i}{2\alpha_i} \right)}{\sum_{i \notin \Theta} \frac{1}{2\alpha_i}}}$$

right. So, this is all we are saying, we have not talked about the algorithm part yet ok. So, in fact you can show, so if let us say if I define theta to be set of generators for which inequality constraints are active. So, it basically is theta is a set of alternatives for which either pi is equal to pi min or pi is equal to pi max. So, essentially pi is pi min or P i is equal to P i max.

Suppose P i star rather let me call it the optimal dispatch value. Suppose this is the case then for the capacitated economic dispatch volume lambda star is essentially defined is you can show that this is equal to lambda star uncapacitated plus summation i in theta summation i not in theta 1 over 2 alpha x. So, this is the formula for the optimal value of lambda star for the capacitated economic dispatch value. So, if let us say theta happens to be the empty set that means everything is within the generation limit. Then this numerator is 0 and you get lambda star is same as lambda uncapacitated star which was the previous one that we had obtained.

If this is not the case then you are going to get a slightly different value. And the algorithm that we are going to design, we are first going to solve the uncapacitated economic dispatch volume and obtain this number. And once I get this uncapacitated value lambda uncapacitated star, after that what I am going to do is, I am going to find all the generators for which the generator generation dispatch value is either greater than pi max or lesser than pi min. So, then what I am going to do is, I am going to saturate them at those limits. and then also basically add those generators to this index set theta and basically using this information I will be computing this quantity ok.

So, is the overall idea clear and because, so I will come to the challenge part of it later and because they are only going to be finite number of generators in the network there will only be finite number of this capacity violations right. So, the every time you that need I mean let us say you every time that you need to basically update this lambda cap capacitated star that is going to be only finite number of times. So, if you can. So, the challenge here lies in computing this term right. Why? Because I need to sum this quantity.

So, that means again I need to exchange certain information with my neighbors. to be able to get the numerator and the denominator. So, let me first write down the algorithm and I think it would be it would be clear as to how this algorithm works otherwise it will be difficult for me to explain as well. So, algorithm for capacitated EDP. So, step 1 is your this theta set for which the generator constraint violations or the capacity constraint violations are there, you are going to initialize this to an empty set.

So, then what you do is you solve the uncapacitated economic dispatch problem. So, solve uncapacitated economic dispatch problem and this is going to give you lambda star and Pi star. So, this is for the uncapacitated let us call it uncapacitated just to be clear, but this is the uncapacitated economic dispatch column. Then you would set this lambda star to be because if let us say there are no violations in this lambda star is same as uncapacitated star. So, you would set this to be uncapacitated star and then you will run a while loop and you will see if there are any constraint violations.

So, while you have generator or generation constraint violation. So, what do you do, you compute, you basically obtain the set of all generators for which these violations are there. So, we define the set omega, which is going to be all the all the generators which are not already in your. So, again theta is the set that we are where we are going to be maintaining a list of all the generators for which there are going to be capacity constraint violations right. So, for all the generators for which there are no capacity constraint violations, and it turns out that when you run this, which is already not included in theta, either pi is less than equal to p is less than pi min or pi is that means you have you have a violation right pi is greater than pi max.

* $\Theta = \phi$
* Solve uncapacitated EDP $\rightarrow (\lambda^*_{un}, \{P_i^*\})$
* $\lambda^* \leftarrow \lambda^*_{un}$
* While Generation constraint violation.
  - $\Omega \triangleq \{ i \notin \Theta : (P_i < P_{i,min}) \text{ or } (P_i > P_{i,max}) \}$

    $\Theta \leftarrow \Theta \cup \Omega$
  - Calculate optimal dispatch

    $$P_i \leftarrow \begin{cases} \frac{\lambda_i - \beta_i}{2\alpha_i}, & \text{if } i \notin \Theta \\ P_{i,max} \text{ or } P_{i,min} & \text{if } i \in \Theta \end{cases}$$
  - $$(y_i(0), z_i(0)) \leftarrow \begin{cases} \left(\frac{\lambda - 2\alpha_i P_i - \beta_i}{2\alpha_i}, 0 \right) & i \in \Theta \\ \left(0, \frac{1}{2\alpha_i} \right) & i \notin \Theta \end{cases}$$
  - Run avg. consensus on $\{y_i\}$ and $\{z_i\}$
  - $\lambda^* \leftarrow \lambda^* + \frac{y_c}{z_c}$

end while

So, what you do is you find such sets and you update your theta to be your current theta union omega ok. So, all the generators which are not originally included in your theta set now you are going to append that your to your theta set right because you are going to be maintaining a list of all the generators for which there is violation ok. you are going to be calculating the optimal dispatch which is simply going to be pi is going to be 2 alpha i if this pi is or if this let us say i is not in this set theta. So if i is not in the set theta, then this is the optimal dispatch value. And if it is in the set theta, it is either going to be pi max or pi min if i is in theta.

So this is the dispatch value that you are going to be. This is how you are going to be updating your dispatch value. Is this clear? again this basically comes from here right. If there is no generation constraint violation then lambda gets this value otherwise it is going to be either saturated P i max or P i min and that is what we are doing. Now, after this, we need to compute this particular term right.

So, run this algorithm and what we do is we define y i 0 and z i 0, let me just write this first and then it would be clear. So, remember what do we want here, we want to compute this thing right. Essentially, let us say if every generator is trying to like let us say for this numerator, I get I basically I get to this particular quantity, this whole thing and for the denominator, I get to this quantity. So, this 1 over n, 1 over n cancels out and then what you get is a numerator over denominator So, essentially we are because I mean this would require to come like exchange certain information with my neighbors to be able to compute these numerator and denominator right. And this is what we do, you define your initial y i 0 and z i 0 to be this number.

And because you are going to run average consensus, so this summation is going to be the same as this particular thing right. And if I sum it over, I will basically get, if I run the consensus, I will get to the average value of this, which is going to be summation 1 over n summation like 1 over n summation i 1 through n for all i in the generators generation set here right. And when i is not in the generator set, you are essentially going to get 0. So, the generators which are not included in theta, they are going to be broadcasting 0 and the generators which are going to be included which are going to be in theta they are going to be broadcasting this number. So, if you run the average this is what instead of y 1 through n you will get y in this set theta.

Likewise for generators which are not in theta they are going to be broadcasting 1 over 2 alpha a and which the one which are in n theta they are going to be broadcasting 0 and that is how you are going to get this sum. So, what you will do is then you will get your Once you run this consensus algorithm, so you are going to run average consensus on y i and z i and you can run fixed time consensus scheme on this and then you basically update your lambda star which is going to be current lambda star plus the consensus value of y and consensus value of z. like this. So, that you get this term right and this is when you end the while loop here and this basically solves the capacitated economic dispatch problem. Now, why does it run in fixed time? Why? Because we run this consensus scheme in a fixed time, there are finite number of generators So, you are going to be taking a fixed amount of time to be able to solve this.

So, the first like whenever you see a constrained or constrained or capacitated economic dispatch problem, you are first going to be solving an uncapacitated one. If the generator constraints are already satisfied and you do not care, if they are not satisfied, then you are going to saturate those and then that is where you run this consensus. Only on those generators for which you essentially have this capacity constraints getting violated. If the moment there are no violations, then you just exit this while loop and then you are done. So, this pretty much I mean sums up today's lecture that is all I wanted to cover in today's lecture which is an application of distributed economic dispatch problem or distributed optimization problem towards economic dispatch rather. And in the next class I am going to be solving the general summation fi xi kind of distributed optimization problem both looking at both using discrete time algorithms as well as continuous time algorithms. Thank you very much.