A GPU-Based Lagrange Multiplier Optimization for Dynamic Economic Dispatch Aiming to Reduce Wind Power Curtailment

Miao Liu¹, Gang Liu¹, Guohui Zhao²*, Zhengwen Li¹, Shiyi Zhao¹, Chunyu Guo¹, Zhao Zhang¹ and Zhuang Chu²

¹State Grid Liaoning Power Co., Ltd. Science and Technology Project 2018YF-07, China
²Northeast Electric Power University, Jilin, China

*Corresponding author e-mail: 1575531323@qq.com

Abstract. Due to the large-scale integration and the high uncertainty of wind power, the curtailment of wind power happens from time to time. To some extent, the curtailment could be reduced by adjust short-term and ultra-short-term dispatching of power systems. In this paper, a mathematic model dynamic economic dispatch (DED) is set up aiming to reduce the curtailment of wind power in in power systems. Computation of the adjustment is needed to be completed in very short time because it will be carried out in following one more hour. Therefore, the technology of heterogeneous computing is introduced to accelerate the DED solving process of Lagrange Multiplier optimization, and graphic processor unit (GPU) is used to execute parallel computing multiple data with the same instruction threads. A case based on IEEE reliability system is used to test effectiveness and efficiency of the model and the algorithm.

1. Introduction

The problems associated with wind power curtailment [1-2] during power system operation have become a concern in recent years. The uncertainty of wind often leads to some difference between the predicted wind power and the actual available wind power, which is one of main reasons of wind power curtailment. Usually, the shorter the time between the predicting instant and the predicted instant, the more accurate the prediction is. Thus, an earlier prediction of wind power may differ considerably from a lately one. If the differences between the predicted wind powers and the actual ones are too large to be neglected, the dispatching scheme based on earlier prediction should be adjusted. Otherwise wind power curtailment will inevitably happen. In order to reduce the curtailment [3], adjustment of dispatching scheme is necessary. Adjustment computing has to be of great efficiency and cost little time for the adjustment is close to its implementation.

The adjustment of dispatching schemes aims at the sections of the entire scheduling and dispatching period, and each of these sections may cover a certain smaller number of intervals than the number of all. Accordingly, unit commitment is not necessary for it cannot be executed in such short time. The adjustment will take place in some continuous intervals and no generators will be switched...
on/off, and it could be formulated as a large-scale, continuous, and nonlinear optimizing model, which is the same as traditional dynamic economic dispatch.

Plenty of studies on DED have been published for decades. The objective function of early models was to minimize fuel costs on generating electric power, and the constraints included the minimum and maximum limits of generators’ active power output, and the active power balancing constraints of power systems at each dispatch interval between generation and loads. And spinning reserves of power systems, ramp constraints of thermal generators have been taken into consideration in further studies. With the advent of electricity markets, the models of DED have been also amended to meet different demands and needs. In general, with more and more influencing factors taken into account, the computing scale of DED has been increased with the increasing number of constraints, and effective and efficient methods are needed to solve it.

Nonlinear programing methods, such as quadratic programming [4], Benders decomposition [5], and interior methods [6-7], are suitable to be applied to solving the DED problem for it is a typical nonlinear optimization problem. Meanwhile, stochastic optimization methods, such as genetic algorithms [7], evolutionary programming, and particle swarm optimization [8-9], have been also used to solve DED. These methods were usually realized as serial computing, which were adaptable to hardware and software circumstances at the time. Parallel computing technologies have been developing rapidly in recent years. The compute unified device architecture (CUDA), based on graphic processing unit (GPU) and released by NVIDIA in 2007, is a promising parallel computing technology for its high efficiency with relatively lower costs. CUDA is an extension to the C Language that allows GPU code to be written in regular C. GPU and CUDA have been used exclusively for graphics, can be harvested to speedup scientific calculations. CUDA is a heterogeneous computing platform, and the GPU card must operate in conjunction with a CPU-based host. This kind of parallel computing has also been applied to power systems [10-11] In [10-12] the Newton-Raphson method for solving the power-flow problem is implemented on GPU, the calculation of the right-hand side of equation and the Jacobian matrix, LU decomposition and forward-back, substitution are implement on GPU; [12] propose a GPU-accelerated solution that creates an additional layer of parallelism among batch ACPF and consequently achieves a much higher level of overall parallelism.

This paper presents an implementation of the Lagrange Multiplier Optimization, as it pertains to parallelizing and implementing in CUDA, for solving DED with reducing wind power curtailment. The remainder of this paper is arranged as follows. Section II introduces the mathematic model of DED aiming to reduce wind power curtailment. In Section III, a GPU-based Lagrange multiplier Optimization for dynamic economic dispatch is proposed. Numerical tests are carried out in Section IV to demonstrate the effectiveness of the proposed model and the efficiency of the GPU-based parallel computing method. Conclusions are drawn in Section V.

2. Research Problem formulation

When a lately wind power prediction differs from the earlier one on which the schedule and dispatch of power systems are determined, it is advisable to rearrange generators’ dispatch of the several following continuous periods in order to reduce wind power curtailment. The problem of re-dispatch takes on a form of dynamic economic dispatch (DED), for it will take a long time to re-schedule generators, and unit commitment is not needed taking into account.

It is not necessary to reduce the unbalanced wind powers between the two earlier and latest predictions in the entire dispatch period. Thus, the first and last interval that is taken into rearrangement are denoted by 1 and T.

A mathematic model of DED aiming to reduce wind power curtailment is as follows. The objective function is:

\[
\min f = \sum_{i=1}^{T} \sum_{g=1}^{G} [F_t(P_t)]
\]  

(1)
Here, \( T \) is the number of time intervals of DED, and \( G \) is the number of turbine generators; \( P_{it} \) is the active power output of turbine generator \( i \) at the \( t \)-th time interval; \( F_i(P_i) \) is the cost of turbine generator \( i \) with real power output \( P_i \), usually it has the form as \( F_i(P_i) = a_i P_i^2 + b_i P_i \) and \( a_i, b_i \) are coefficients of generator \( i \).

The constraints are listed as follows.

1) Active power output limits of each generator:

\[
P_{i,\text{min}} \leq P_{it} \leq P_{i,\text{max}}, \quad t = 1, 2, \cdots, T, \quad i = 1, 2, \cdots, G
\]  

(2)

Here, \( P_{i,\text{min}} \) and \( P_{i,\text{max}} \) are the allowed minimum and maximum real power output of turbine generator \( i \) respectively;

2) Real power output constraints of generators at the first and the last interval:

\[
P_{i,1} = P_{i,\text{ch}}^{\text{in}}, \quad i = 1, 2, \cdots, G
\]  

(3)

\[
P_{iT} = P_{i,\text{ch}}^{\text{in}}, \quad i = 1, 2, \cdots, G
\]  

(4)

Here, \( P_{i,\text{ch}}^{\text{in}} \) are the real power outputs of generator \( i \) at interval 1 and \( T \), in the scheduling and dispatching scheme in which the differences of wind power between earlier prediction and latest one.

3) Active power balancing of each bus of power systems at each interval:

\[
 B \theta_t = MP_{G,t} - D_t, \quad t = 1, 2, \cdots, T
\]  

(5)

Eq. (5) is a simplified power flow equation, \( \theta_t \) is the vector consisting of bus voltage phasors’ angles at each interval, and \( B \) is the node inductance matrix of power systems, in the process of forming which branch resistances are normally omitted. Here, \( D_t \) is a column vector consisting of each node’s real power demands of power systems at interval \( t \); \( P_{G,t} \) is a column vector consisting of the latest wind power predictions and each being dispatched traditional generator’s real power outputs at interval \( t \), and it will take the form as follows at bus \( j \) if both wind farms with predicted power output \( P_{i,j}^{w,j} \) and traditional generators with output \( P_{i,j}^{g,j} \) are located at bus \( j \):

\[
 P_{G,t}^j = P_{i,j}^{w,j} + P_{i,j}^{g,j}, \quad \forall i, j = 1, 2, \cdots, G, j = 1, 2, \cdots, N
\]  

Here, \( N \) is the bus number of power systems, and \( \forall i \in j \) means that generator \( i \) is located at bus \( j \). \( M \) is the incidence matrix between nodes and generators, of which rows correspond to the nodes, and columns correspond to the generators. If \( m_{ij} = 1 \), the generator \( j \) is located at node \( i \).

4) Transmission constraints:

\[
 \left| \frac{\theta_{jt} - \theta_{kt}}{x_{jk}} \right| \leq P_{jk}
\]  

(6)

Here, \( \theta_{jt} \) and \( \theta_{kt} \) are the node voltage phasors’ angles of bus \( j \) and \( k \). \( x_{jk} \) and \( P_{jk} \) are the branch admittance and the power delivering limit of transmission line \( j - k \) separately.

5) Ramp constraints:
Here, $P_{i,t}^{\text{ramp-d}}$, $P_{i,t}^{\text{ramp-u}}$ are respectively the maximum increasing ramp and decreasing ramp of turbine generator $i$.

6) Spinning reserves constraints of power systems:

$$R_{t}^{\text{down}} + \sum_{i=1}^{N} P_{i,\text{min}} \leq \sum_{i=1}^{N} P_{i,t} \leq R_{t}^{\text{up}} + \sum_{i=1}^{N} P_{i,\text{max}}, \quad t = 1, 2, \cdots, T$$

Here, $R_{t}^{\text{down}}$, $R_{t}^{\text{up}}$ are respectively the decreasing reserve and increasing reserve of power systems at interval $t$.

Since the up-to-date wind predictions is dealt with as disturbances just like demands, the model constructed above is similar to the common DED model, which is a large-scale, and continuous nonlinear programming. The main difference is those constraints incorporated for adjustment of wind power predictions. In order to meet the need of high efficiency, a GPU-based parallel computing is adopted to solve the problem.

3. Basic Principles of Lagrange Multiplier Optimization

Lagrange multiplier method is a kind optimization to solve constrained nonlinear programming problems, which is used to solve DED in this paper.

The augmented Lagrangian function can be written as follows:

$$\phi(P, \lambda, \mu, \alpha, \beta) = f(P) - \alpha \lambda^T g(P) - \beta \mu^T h(P)$$

Here, $g(P)$ stands for all the equality constraints of the problem, $h(P)$ stands for all the inequality constraints, and $P$ is a vector that consists of $P_{i,t}$ ($t = 1, 2, \cdots, T, i = 1, 2, \cdots, N$). Eq. (5) constitutes equality constrains, and inequality constrains consist of Eq. (2), and (6) – (8). $\lambda$ is the vector that consists of Lagrange multipliers of equality constraints, and $\mu$ is the vector that consists of Lagrange multipliers of inequality ones.

Then the iteration of variables and Lagrange multipliers becomes

$$P^{(k+1)} = P^{(k)} - \alpha_k \left( \nabla f(P^{(k)}) + D_h(P^{(k)}) \lambda^{(k)} + D_g(P^{(k)}) \mu^{(k)} \right)$$

$$\lambda^{(k+1)} = \lambda^{(k)} - \beta_k h(P^{(k)})$$

$$\mu^{(k+1)} = \max \left\{ \mu^{(k)} + \beta_k g(P^{(k)}), 0 \right\}$$

Here, $Dh(P)$ is the Jacobi matrix of $h(P)$. $\alpha_k$ and $\beta_k$ are penalty factors of sufficiently large values, which should be adapted according to the problem and could be constant during iteration.

A concise flow chat of Lagrange Multiplier Method is shown in figure 1:
In the solving process shown in figure 1, the majority of computing time and resources are spent on three steps: updating the equality constraint multipliers $\lambda^{(i)}$, updating inequality constraint multipliers $\mu^{(i)}$, and updating the real powers $P^{(i)}$ of generators. According to the principles of the Lagrange Multiplier method, the three steps are executed step by step. Furthermore, in a typical series computing context, each element of $\lambda^{(i)}$ is updated one after another, and the same with each element of $\mu^{(i)}$ and $P^{(i)}$.

4. A GPU-based Lagrange multiplier Optimization for dynamic economic dispatch

From the principles of Lagrange multiplier optimization and eq. (10)-(12), the computing time and resources are mainly used to update optimized variables $P$, multipliers $\lambda$ of equality constraints, and multipliers $\mu$ of inequality constraints. In this paper, the updating process of them are completed parallelly in GPU. In this section, the GPU parallel computing architecture is introduced briefly, which is followed by the iteration procedures of updating $P$, $\lambda$, and $\mu$.

4.1. GPU Parallel Computing Architecture

Single instruction multiple threads (SIMT) is the remarkable feature of computing provided by CUDA based on GPU, which is remarkably different from multiple instructions multiple data (MIMD). Inside GPU, eight or more Stream Processors (SPs) are attached to a Stream Multiprocessor (SM), and each GPU consists of a number of SMs, the structure of which is shown in Figure 2.
In SIMT, every 32 threads are scheduled as a thread warp and work concurrently. If a problem can be partitioned into coarse sub-problems that can be solved independently, it is possible to be solved in parallel by blocks of threads on any of the available multiprocessors within a GPU. When the number of sub-problems is huge, the GPU based parallel computing costs much less time than traditional CPU based serial computing. For example, calculating each element of Jacobi Matrix and Hessian Matrix, which are widely and frequently used in optimal searching, can adopt the GPU parallel computing [14].

4.2. Procedure of updating optimization variables

A GPU-based parallel computing is adopted to speed up the optimization. Eq. (10) can be rewritten in the form of each element of the vector $P$ as follows:

$$
\begin{align*}
P^{(k+1)}_{it} &= P^{(k)}_{it} - \alpha_k \left[ \nabla F_i \left( P^{(k)}_i \right) + \frac{d}{d P} h_i \left( P^{(k)}_i \right) \lambda_i + \sum_j \frac{d}{d P} g_j \left( P^{(k)}_i \right) \mu_j \right]
\end{align*}
$$

(13)

From Eq. (13), it can be seen that calculating the element of $P^{(k+1)}_{it}$ is irrelevant to other elements. Therefore, all the elements of $P$ can be calculated in parallel on GPU under the same single instruction. Theoretically, all these elements can be iterated at the same time. In practice, the number of parallel is limited to the number of GPU’s SMs.

Accordingly, although the process of updating $P^{(k+1)}_i$ consists of a series of multiplication and addition operation, the instructions and their sequences of calculating is identical for every $P^{(k+1)}_i$. Theoretically, the $G \times T$ elements, e.g. the $G \times T$ real power outputs of $G$ generators at $T$ intervals can be calculated in parallel.

4.3. Procedure of updating equality constraints and transmission constraints

The elements of $\lambda$ and $\mu$ can be calculated in parallel respectively, too. In addition, before the iteration of $\lambda$ and $\mu$, values of $h \left( P \right)$ and $g \left( P \right)$ must be calculated, which also could be made on GPU.

The values of node voltage phasors are calculated by Eq. (5), and are used to calculate the function values of Eq. (6) and their multipliers. The number of these phasors is $T \times (N-1)$, and these $T \times (N-1)$ phasors can be calculated in parallel. However, the calculating of these phasors is not necessary. From Eq. (5), node voltage phasors can be expressed as

$$
\theta_i = B^{-1} \left( GP_{Gi} - D_i \right)
$$

(14)

$H$ is assigned to designate the matrix made up of coefficients of $\theta_i$ in Eq. (6). Hence, the transmission constraints, namely Eq. (6) can be expressed as
\[ -P_{\text{max}}^{\text{TR}} \leq HB^{-1}(GP_{\text{o},t} - D_t) \leq P_{\text{max}}^{\text{TR}} \] (15)

Here, \( P_{\text{max}}^{\text{TR}} \) is the vector made up of \( P_{\text{max}}^{j,k} \), which is the power delivering limit of transmission line \( j - k \). Apparently, the phasors are not necessary for the verification of Eq. (15), so it is with Eq. (6). However, the real powers of all the generators at each interval is the requisite for the verification of transmission limits, and these powers can be obtained by adding active power balancing equations of each interval, e.g. the following equation, to the DED model as a kind of equality constraints instead of Eq. (5).

\[ P_t^{\text{WR}} + \sum_{i=1}^{N} P_{it} = D_t^{e}, t = 1, 2, \cdots, T \] (16)

Here, \( D_t^{e} \) is the entire real power demands of power systems at interval \( t \). The resultant of replacing Eq. (5) with Eq. (16) is that the amended model is of less constraints, and less state variables. Accordingly, the following computing will be of less time, and higher efficiency.

4.4. Procedure of updating other inequality constraint multipliers

As for inequality constraints, it gets a little more complicated for Eq. (3)-(6) are of different forms. Only the constraints that has the same form can be dealt with by SIMT parallel computing. Therefore, the procedure of updating \( \mu^{(t)} \) should be divided into several separate sub-procedures, each of which is corresponding with one form of constraints.

The upper and lower limits, namely Eq. (2), need not dealing with by means of Lagrange multipliers. If the value of some variable exceeds the range of themselves, the value of it should be fixed on the exceeded upper/lower boundary limit. It can be accomplished with a simple kernel function. Real power output constraints of generators at the first and the last interval, namely, eq. (3)-(4), can be dealt with in the same way.

Eq. (7) and (7) are ramp constraints, and spinning reserves constraints respectively. They are of similar forms, and every expression contains two constraints. For example, if \( i \) and \( t \) are fixed, eq. (7) can be expressed equivalently as follows:

\[ \left( P_{i(t+1)} - P_{it} \right) - P_{i}^{\text{amp}u} \leq 0 \]
\[ P_{i}^{\text{amp}d} - \left( P_{i(t+1)} - P_{it} \right) \leq 0 \]

Still, the two constraints and their multipliers should be calculated in the same kernel function. In the function, the two constraints and multipliers are calculated in series. And so are with Eq. (8), and Eq. (15).

5. Numerical tests

To show the effectiveness and efficiency of GPU-based Lagrange Multiplier Algorithm, IEEE reliability system is tested respectively with serial computation on CPU and with parallel computation on GPU&CPU. The CPU is Intel core i5 4690, the computer memory is 16GB, and the GPU is NIVIDIA 960 with 4GB graphic memory.

Since the Lagrange multiplier optimization has been applied to solving DED problems successfully for years, the effectiveness and convergence of it have been testified. Therefore, it is only the difference between the GPU-based heterogeneous and parallel computing and traditional serial computing that is worthy of analyzing and discussing.
Table 1. Comparison of elapse time with cases of different sizes

| Case No. | 1     | 2     | 3     | 4     |
|---------|-------|-------|-------|-------|
| Number of generators | 26    | 26    | 52    | 104   |
| Number of intervals   | 24    | 192   | 192   | 192   |
| Time with GPU&CPU(s)  | 0.018 | 0.040 | 0.092 | 0.342 |
| Time with CPU(s)      | 0.005 | 0.035 | 0.0588| 1.142 |

Table 1 lists the time spent with GPU&CPU and CPU that in four different computing scales. It shows that with a small computing scale, the serial computing on CPU is faster than the heterogeneous computing on GPU&CPU. Only when the scale expands to certain extent, the superiority of parallel computing can show itself.

From Table 1, it is can also be seen that the time with CPU in case 2 which is of 192 intervals is as seven times as in case 1 which is of 24 intervals. While the time with GPU & CPU in case 2 is approximately as twice as in case 1. The increasing trends of spent time can be drawn in Figure 3. Curve 1 is with heterogeneous and parallel computing on GPU & CPU, and Curve 2 is with serial computing on CPU. It can be seen that Curve 2 is increasing faster than Curve 1.

Figure 3. Typical architecture of graphics processing units

With the computing scale increasing, both the heterogeneous computing and the serial computing need taking more time and more resources to obtain the solution of DED. And the larger the computing scale is, the more time can be saved with the heterogeneous computing. Under the circumstances in which the penetration of large-scale wind power has added more complexity to the DED problem, and has needed more computing resources, the heterogeneous computing based on GPU & CPU can play an important role in the computing of power system generation’s scheduling and dispatching.

6. Conclusion
Due to the large-scale integration and the high uncertainty of wind power, a new model has been set up in order to reduce the curtailment of wind power that happens from time to time. And a heterogeneous and parallel computing Lagrange multiplier optimization based on GPU & CPU has been applied to solving the proposed model. It could be seen that the GPU-based parallel is of higher efficiency, is of the same effectiveness, and is suitable for optimization computing of power systems.
Acknowledgments
This work was supported by the Science and Technology Foundation of State Grid Liaoning Electric Power Supply Company under Grant 2018YF-07.

Liu Miao, Liu Gang, Li Zhengwen, Zhao Shiyi, and Guo Chunyu are with State Grid Liaoning Electric Power Supply Company, Shenyang, Liaoning Province, China (e-mail: liugang_ln@126.com).
Chu Zhuang, and Zhao Guohui is with School of Electrical Engineering, Northeast Electric Power University, Jilin City, Jilin Province, China (e-mail: chuzhuang@hotmail.com).
Zhang Zhao is with Liaoning Electric Power Research Institute, Shenyang, and Liaoning Province, China (e-mail: 534998653@qq.com).

References
[1] Xiaowei Dui, Guiping Zhu, Optimal unit commitment based on second-order cone programming in high wind power penetration scenarios, IET Renewable Power Generation, 2018, 12 (1): 52-60.
[2] Haibo Li, Zongxiang Lu, Ying Qiao, etc., A Non-Sequential Probabilistic Production Simulation Method for Wind Energy Curtailment Evaluation Considering the Seasonal Heat Supply Constraints, IEEE Transactions on sustainable energy, 2018, 9 (1): 462-473.
[3] Cui Yang, Chen Zhi, Yan Gangui, etc., Coordinated Wind Power Accommodating Dispatch Model Based on Electric Boiler and CHP With Thermal Energy Storage, Proceedings of the CSEE, 2016, 36 (15): 4072-4080 (in Chinese).
[4] Wang Bao, Xu Xujian, Sun Yuanzhang, etc., Stochastic Dynamic Economic Dispatch of Power Systems Considering Wind Power Based on Versatile Probability Distribution, Automation of Electric Power Systems, 2016, 40 (6): 17-24 (in Chinese).
[5] Chenhui Lin, Wenchuan Wu, Boming Zhang, etc., Decentralized Solution for Combined Heat and Power Dispatch through Benders Decomposition, IEEE Transactions on sustainable energy, 2017, 8 (4): 1361-1372.
[6] Yang Liuqing, Lin Shunjiang, Liu Mingbo, etc., Multi-objective Dynamic Optimal Dispatch for Large-Scale Power Systems Considering Wind Power Penetration, Transactions of China Electrotechnical Society, 2014, 29 (10): 286-295 (in Chinese).
[7] Hong Bowen, Guo Li, Wang Chengshan, etc., Model and Method of Dynamic Multi-objective Optimal Dispatch for Microgrid, Electric Power Automation Equipment, 2013, 33 (3): 100-107 (in Chinese).
[8] Zhou Wei, Peng Yu, Sun Hui, etc., Dynamic Economic Dispatch in Wind Power Integrated System, Proceedings of the CSEE, 2009, 29 (25): 13-18 (in Chinese).
[9] Chen Gonggui, Chen Jinfu, Environmental/Economic Dynamic Dispatch Modeling and Method for Power Systems Integrating Wind Farms, Proceedings of the CSEE, 2013, 33 (10): 27-35 (in Chinese).
[10] Chen Deyang, Li Yalou, Jiang Han, etc., A Parallel Power Flow Algorithm for Large-scale Grid Based on Stratified Path Trees and Its Implementation on GPU, Automation of Electric Power Systems, 2014, 38 (22): 63-69 (in Chinese).
[11] Yang Gang, Yu Le, Liu Mingguang, etc., A Method for Accelerating Power Flow Calculation Based on Multiple Light Threads, Power System Technology, 2013, 37 (6): 1666-1671 (in Chinese).
[12] Gan Zhou, Yanjun Feng, Rui Bo, etc., GPU-Accelerated Batch-ACPF Solution for N-1 Static Security Analysis, IEEE Transactions on sustainable energy, 2017, 8 (3): 1406-1416.