G Model
EPSR-4726; No. of Pages 10
ARTICLE IN PRESS

Electric Power Systems Research xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Optimal power flow based on parallel metaheuristics for graphics processing units

Vincent Roberge*, Mohammed Tarbouchi, Francis Okou

Electrical and Computer Engineering Department, Royal Military College of Canada, Kingston, ON K7K7L6, Canada

ARTICLE INFO

Article history:
Received 2 April 2016
Received in revised form 1 June 2016
Accepted 3 June 2016
Available online xxx

Keywords:
Metaheuristics
Optimal power flow
Parallel processing
Graphics processing units

ABSTRACT

The smart-grid brings new challenges to the optimal dispatch of power. Current research aims to develop optimization techniques capable of handling large networks using accurate models and realistic constraints, all in the shortest possible execution time. For this purpose, this paper presents a metaheuristic-based parallel optimal power flow algorithm for graphics processing units (GPUs). Metaheuristics have the advantage of handling discrete variables and being resilient to premature convergence towards local optima. However, they require significant computing power which limits their use in online applications. The proposed implementation addresses this limitation and significantly accelerates the calculation by exploiting the massively parallel architecture of GPUs. The developed software uses a particle swarm optimizer and runs a full ac Newton–Raphson power flow analysis to evaluate the candidate solutions. The algorithm is tested on the IEEE 30-bus, 118-bus and 300-bus networks and provides a maximum speedup of 17.2×.

1. Introduction

The advancements in information and communication technologies allowed for the modernization of the power grid to form the smart-grid. Real time monitoring and control of the grid are now possible and introduce new control challenges. The optimal power flow (OPF) is one of the core optimization problems in power generation and transmission. It aims to find the optimal power generation dispatch in order to maximize a given objective function at steady-state operation of the power network. This problem was first formulated by Carpentier [1], but has since evolved to include security constraints, discrete control variables and multi-objective functions. In this regard, the OPF is a large-scale, non-linear, nonconvex and multimodal optimization problem with continuous and discrete control variables and remains a significant engineering challenge [2].

In [3,4], Frank et al. present a comprehensive literature review of current solutions to the OPF problem. They find that metaheuristics are particularly efficient at solving the OPF problem as they natively consider discrete variables such as the transformer tap ratios and the static volt-ampere reactive compensator (SVAR) settings. They also note that metaheuristics have the ability to escape local optima, another significant advantage considering the multimodal aspect of the OPF problem. They recommend the use of metaheuristics such as the bacterial foraging method (BF) [2], the particle swarm optimizer (PSO) [5] or the genetic algorithm (GA) [6].

Metaheuristics form a family of non-deterministic optimization methods that relies on the iterative improvement of candidate solutions to solve problems intractable by classical methods. When used to solve the OPF, they must compute a power flow (PF) analysis for hundreds of candidate solutions over several iterations. Their computational requirement is so high that their application is often limited to small networks. Solutions to larger and more realistic power systems lie in the field of parallel programming and high performance computing (HPC) [7]. Interestingly, graphics processing units (GPUs) are an emerging technology in the field of HPC that brings supercomputing to the masses. With the release of the NVIDIA® CUDA™ language in 2007, the massively parallel architecture of GPUs, formerly used solely for graphic applications, can now be harvested to accelerate scientific calculations.

This paper presents a PSO-based parallel OPF algorithm for GPUs. The solver computes the real power and voltage magnitude of generators, the tap ratio of transformers and the reactive power of SVAR systems in order to minimize an objective function based on generation costs, transmission losses or pollutant emissions. It uses the ac power model and relies on a parallel Newton–Raphson

* Corresponding author. Tel.: +1 613 541 5010x6492.
E-mail address: vincentroberge@gmail.com (V. Roberge).

http://dx.doi.org/10.1016/j.epsr.2016.06.006

Nomenclature

Objective function
$\bar{u}$ vector of control variables
$\bar{x}$ vector of state variables
$f(\bar{u},\bar{x})$ objective function
$g(\bar{u},\bar{x})$ equality constraints
$h(\bar{u},\bar{x})$ inequality constraints
$f_{costs}(\bar{u},\bar{x})$ total network generation costs ($)
$f_{losses}(\bar{u},\bar{x})$ total network transmission losses (MW)
$f_{emissions}(\bar{u},\bar{x})$ total network pollutant emissions (metric tons)
w_1, w_2, w_3 weight of the three terms forming the objective function

Transmission network
N_{bus} number of buses in the network
N_{branch} number of branches in the network
$N_{SLACK\,gen}$ number of generators at the slack bus
N_{gen} number of generators in the network including those at the SLACK bus
N_{trans} number of transformers in the network
N_{SVAR} number of SVARs in the network
V_i complex voltage at bus i (p.u.)
$|V_i|$ voltage magnitude at bus i (p.u.)
δ_i voltage angle at bus i (radians)
P_{Di} real power demand at bus i (p.u.)
Q_{Di} reactive power demand at bus i (p.u.)
P_{Gj} real power generation at generator j (p.u.)
Q_{Gj} reactive power generation at generator j (p.u.)
a_j, b_j, c_j generation cost coefficients for generator j
$\alpha_j, \beta_j, \gamma_j, \zeta_j, \lambda_j$ pollutant emission coefficients for generator j
T_k tap ratio at transformer k
q_{cm} reactive power injected at SVAR m (p.u.)
G_{ik} conductance of the branch connecting bus i to bus k (p.u.)
B_{ik} susceptance of the branch connecting bus i to bus k (p.u.)
S_i complex power flowing in branch i (p.u.)
$|S_i|$ apparent power flowing in branch i (p.u.)

Particle swarm optimization
t iteration
$\bar{v}$ velocity of a particle
$\bar{y}$ position of a particle
$\bar{b}$ best position ever visited for each particle
$\bar{g}$ best position ever visited by any particle
$\bar{r}_1, \bar{r}_2$ vectors of random real values between 0 and 1
ω, c_1, c_2 inertia, personal influence and social influence weights

Fitness function
$\mathcal{F}(\bar{u},\bar{x})$ fitness function
$E(p)$ relative excess of parameter p where p could be the bus voltage magnitude $|V|$, the real power generation P_G, the reactive power generation Q_G or the branch apparent power $|S|$
$E_{total}(\bar{u},\bar{x})$ total relative excess for the network
$\mathcal{V}_{NORM}(\bar{u},\bar{x})$ normalized violation factor
$f_{NORM}(\bar{u},\bar{x})$ normalized objective function

(N–R) PF analysis on GPU to compute the full state of the network prior to evaluating the fitness of the candidate solutions. The effectiveness of the algorithm is tested on the IEEE 30-bus, 118-bus and 300-bus networks. By exploiting the massively parallel architecture of the GPU, the proposed software offers accelerations from 16.2× to 17.2× compared to a sequential execution on CPU.

The contribution of this paper is two folds. Firstly, it proposes a strategy to parallelize both the PSO and the evaluation of the candidate solutions on a GPU. It actually is the very first time a metaheuristic-based OPF solver is implemented on a GPU. Equipped with thousands of core, the GPU provides a significant speedup and allows for a very fast optimization. Secondly, it describes how to efficiently employ the PSO to solve the OPF problem with continuous and discrete variables and achieve better solutions than previous implementations. This improvement is possible by the use of a multi-phase optimization approach and a very large number of candidate solutions, thanks to the parallel implementation.

The remainder of this paper is organized as follows: Section 2 reviews some of the key publications on the topic; Section 3 outlines the mathematical formulation of the OPF problem; an overview of the GPU architecture is provided in Section 4; Sections 5 and 6 describe the optimization strategy used by the proposed OPF solver and its parallel implementation on GPU; and finally, experimental results are presented in Section 7.

2. Related works

The OPF problem was formulated by Carpentier in 1962 [1]. Due to the non-convexity introduced by the power flow equations, the OPF problem remains an ongoing research topic. Many deterministic and non-deterministic methods have been developed for the OPF problem over the years. A comprehensive survey is provided in [3,4]. Two families of deterministic methods that are particularly noteworthy are the interior point (IP) methods [8] and the semidefinite programming (SDP) methods [9]. The IP methods define barrier functions to handle the inequality constraints and restrict the search within an area of feasible solutions. Starting from a known solution, the direction of the search is calculated at every iteration and the method converges toward the closest local optimum. Unfortunately, due to the non-convexity of the problem, this local optimum is not guaranteed to be global. With the SDP methods, the OPF problem is formulated using a convex relaxation and can be solved in polynomial time. The solution is guaranteed to be global and correct if the gap between the relaxation and the actual OPF problem can be proven to be zero. This is the case for the IEEE 30, 118 and 300-bus networks as observed in [10]. However, a simple 3-bus example is given in [11] to show that this gap is not always tight. Latest research on the topic aims to identify the conditions under which this gap is zero [12,13] or to provide strategies for when it is not [14].

Non-deterministic methods, more precisely metaheuristics, are a popular approach to solve the OPF problem [4,15]. They can consider continuous and discrete variables, they allow for non-differentiable objective functions and they reduce the chances of a premature convergence toward a local optimum. In the case of the OPF problem, the transformer taps and the SVAR settings are discrete by nature. If treated as continuous during the optimization process and rounded afterward, they can lead to suboptimal or even non-feasible solutions, especially when step sizes as large as 40 MVar are common for SVARs [16]. In that aspect, metaheuristics are very advantageous methods for the OPF problem. To name a few examples, Soliman and Mantawy proposed in [5] the use of a PSO to minimize the generation costs in the IEEE 30-bus network. Their approach relied on a fast decoupled load flow (FDLF) to compute the dependent variables associated with each candidate solution before evaluating the objective function. The solutions computed provided significant generation cost savings, but the scalability of the optimizer was not tested on larger networks. In [6], Bakirtzis