Parallel Differential Evolution Based on Distributed Cloud Computing Resources for Power Electronic Circuit Optimization

Xiao-Fang Liu, Zhi-Hui Zhan* (Corresponding Author), Jun-Hao Lin, Jun Zhang
School of Computer Science and Engineering,
South China University of Technology, Guangzhou, China, 510006
*cszhanzhh@scut.edu.cn

ABSTRACT
Power electronic circuit (PEC) design and optimization is a significant problem in engineering area. Due to its complexity, evolutionary computation algorithms such as differential evolution (DE), genetic algorithms, and particle swarm optimization have been used successfully to obtain optimal components for PEC. However, since the fitness evaluation of PEC is often very expensive, these methods are computationally demanding and cannot easily be used for real time control or large scale problem. Therefore, finding a simple and powerful method to reduce the computational time is an important work. In this paper, a distributed parallel DE (PDE) is proposed to implement on a set of distributed cloud computing resources in order to accelerate the computation. The experimental results indicate that more computational resources for parallel implementation can indeed help to reduce the computational time efficiently. Therefore, the PDE paradigm significantly speeds up the computation for expensive fitness evaluation, making it more suitable for complex optimization problems in big data environments.

Keywords
Power electronic circuit, parallel, differential evolution, expensive fitness evaluation, big data.

1. INTRODUCTION
The power electronic circuit (PEC) design has been widely utilized in different areas, such as industrial, commercial, residential, and utility area [1]. The PEC consists of a number of components such as resistors, capacitors, and inductors [2]. The optimization of the structure and control parameters of the components is a challenging problem.

Genetic algorithm (GA) [2], particle swarm optimization (PSO) [3], normalization group brain storm optimization (NGBSO) [4], and differential evolution (DE) [5][6] have been reported successfully applied to solve the PEC problem. However, the fitness evaluation of candidate solutions for PEC problem is time-consuming. The long computational time costs may make evolutionary algorithms difficult to be widely applied on PEC problem or other expensive fitness evaluation, large scale, or real-time problems. Today, wide variety ways of access to resource, such as cloud computing [7], allow us to break through the traditional single processor limitation and significantly reduce the execution time through parallel computing.

In this paper, we propose a parallel DE (PDE) on cluster to speed up the optimization process when solving expensive fitness evaluation problems, and extend it to the PEC problem. The fitness values of the candidate solutions in the population are calculated in parallel to reduce the computational time. The distributed PDE is parallelized in a two-level parallel computing topology and implemented on a PC-cluster with multi-core physical machines. The PC-cluster contains one master node and a number of slave nodes.

The master node is to control the whole evolution of the PDE algorithm and assign computation tasks to the slave nodes. The evolution operation of the population is carried out on the master node. When the new individuals are generated on the master node, they are assigned to slave nodes to calculate the fitness values. And the slave node returns the fitness values to the master node once it has finished the calculation. In this process, master node transfers the candidate solutions to slave nodes and obtain their corresponding fitness values from slaves, while each slave node only receives candidate solutions from the master node and returns the corresponding fitness values to the master node. Note that there is no messages communication between slave nodes. Each master and slave node are implemented by a core on different processors. Since the information exchange is an important step in the parallel computation, we adopt the message passing interface (MPI) herein.

2. PERFORMANCE EVALUATION
The PDE uses the same DE operators as the one in [6], except that PDE uses a distributed parallel paradigm for fitness evaluations to reduce the computational time. The parameters are set to be population size NP = 100, scale factor F = 0.5, and CR=0.1. In all simulations, the number of maximal fitness evaluations (FEs) is set as 1.5×10^4. In the experiments, the components of PEC were computed under five cases with different number of cores available. If n cores are given, then one will be the master node, and the n-1 cores will play the role as slave nodes to calculate the fitness values. In special case, the algorithm executes serially when only one core is available. The average execution times of 10 runs in cases where given 1, 2, 6, 26, 51, and 101 cores are measured and reported in Table 1 and Fig. 1. The sequential implementation is executed on an Intel CPU i5-3470 with

GECCO'16 Companion, July 20-24, 2016, Denver, CO, USA.
ACM 978-1-4503-4323-7/16/07.
http://dx.doi.org/10.1145/2908961.2908972

3.20GHz in one core while the parallel implementation, on a cluster with 26 PCs with 4-core i5-3470 with 3.20GHz.

Table 1. The average execution time of PDE with different number of processors and the corresponding speedup compared with the sequential execution.

Number of Cores *n*	Time (second)	Speedup
1	1002.01	1
2	1003.26	1
6	404.846	2.5
26	84.0099	12
51	41.3589	25
101	21.5278	48

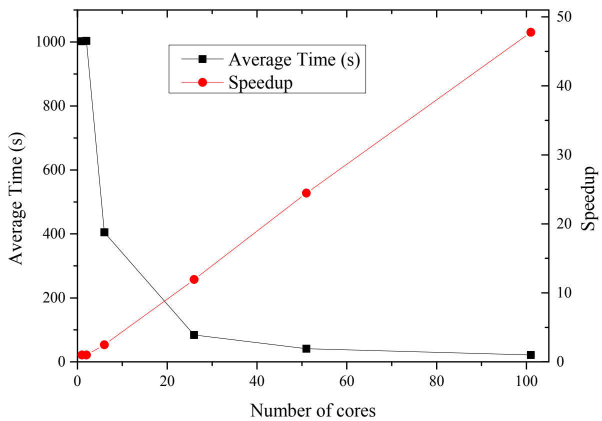

Fig. 1. The average execution time of PDE with different number of cores and the corresponding speedup compared with the sequential execution.

From Table 1, it can be observed that the execution time for the sequential program is 1002.01 seconds. If there are only 2 cores, then one core is the master node and the other is the slave node. In this condition, all the individuals are assigned to the only one slave node and are calculated sequentially. Therefore, the computational time is similar (1002.01 seconds and 1003.26 seconds) because the fitness values of the whole population are evaluated sequentially in both the 1 core and 2 cores cases. However, the execution time is significantly reduced for the parallel program with 6, 26, 51, and 101 cores, whose values are 404.846 seconds, 84.009 seconds, 41.3589 seconds, and 21.5278 seconds, respectively.

When considering the speedup by comparing the case with 2 cores and the case with 6 cores, the execution time of the 6 cores case did not speed up five times but only about 2.5 times. This may be caused by the transmission cost between physical machines when more processors are used. The speedup achieved by 6, 26, 51, 101 cores are 2.5, 12, 25, and 48, respectively. Therefore, the speedup increased near-linearly with the number of the available cores, as shown in Fig. 1.

From another view, compared with the case with 6 cores, the speedup achieved by 26, 51, 101 are approximately 5, 10, and 20, this shows that the proposed parallelism can effectively utilize the resources in return for the reduction of the execution time. The results show that the proposed distributed PDE can reduce the time significantly and perform high parallel efficiency for PEC problem.

3. CONCLUSIONS

This paper tackles the issue of the high time cost in solving PEC optimization problem. Since it is easier to access computational resources today, using more resources for distributed and parallel implementation is a natural and effective way to reduce the required execution time. Based on the consideration of the expensive fitness evaluation, a parallel implementation of the DE (termed PDE algorithm) was proposed to address the PEC problem. The fitness evaluation of the candidate solutions are calculated in parallel on a set of distributed processors to reduce the execution time. In the test cases, the parallel implementation proposed achieved a time speedup of a near-linear growth with the number of the available processors. The execution time is significantly decreasing from 1002 seconds to 21 seconds, which shows the efficiency of parallelism. We believe that this work provides a powerful solution for control of PEC and demonstrates a promising option of using distributed and parallel computing to solve expensive fitness evaluation problems, large scale problems, and real-time problems.

4. ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundations of China (NSFC) with No. 61402545, the Natural Science Foundations of Guangdong Province for Distinguished Young Scholars with No. 2014A030306038, the Project for Pearl River New Star in Science and Technology with No. 201506010047, the NSFC Key Program with No. 61332002, and the Fundamental Research Funds for the Central Universities (15lgzd08).

5. REFERENCES

[1] Z. Chen, J. M. Guerrero, and F. Blaabjerg, "A review of the state of the art of power electronics for wind turbines," *IEEE Trans. Power Electron.*, vol. 24, no. 8, pp. 1859-1875, Aug. 2009.

[2] J. Zhang, H. Chung, W. L. Lo, S. Y. R. Hui, and A. Wu, "Implementation of a decoupled optimization technique for design of switching regulators using genetic algorithm," *IEEE Trans. Power Electron.*, vol. 16, no. 6, pp. 752-763, 2001.

[3] J. Zhang, Y. Shi, and Z. H. Zhan, "Power electronic circuits design: A particle swarm optimization approach," in *Proc. The 7th International Conference on Simulated Evolution And Learning*, 2008, pp. 605–614.

[4] Z. H. Zhan, J. Zhang, Y. H. Shi, and H. L. Liu, "A modified brain storm optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2012, pp. 1-8.

[5] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[6] Z. H. Zhan and J. Zhang, "Differential evolution for power electronic circuit optimization," in *Proc. Conf. Technologies and Applications of Artificial Intelligence*, 2015, pp. 158-163.

[7] Z. H. Zhan, X. F. Liu, Y. J. Gong, J. Zhang, H. S. H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," *ACM Computing Surveys,* vol. 47, no. 4, Article 63, pp. 1-33, Jul. 2015.