# Load Balance Aware Distributed Differential Evolution for Computationally Expensive Optimization Problems

Ning Ma[1], Xiao-Fang Liu[2], Zhi-Hui Zhan[2](Corresponding Author), Jing-Hui Zhong[2], and Jun Zhang[2](Corresponding Author)

[1]School of Computer Science and Engineering, Beihang University, Beijing, China, 100191
[2]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, 510006
Corresponding Author Email: zhanapollo@163.com; junzhanghk@qq.com

## ABSTRACT[1]

Computationally expensive problem challenges the application of evolutionary algorithms (EAs) due to the long runtime. Distributed EAs on distributed resources for calculating the individual fitness value in paralllel is a promising method to reduce runtime. A crucial issue in distributed EAs is how to scheduling the individuals to the distributed resources. Different resources are often with different load and the resource with slow computation ability often limits the parallel speed. To improve the performence, the load information of each resource is considered and used for resource allocation strategy in this paper. We proposed a distributed differential evolution (DDE) algorithm with a load balance strategy to efficiently utilize the concurrent computational resource for power electronic circuit design, which is a computationally expensive optimization problem. This way, the topology related to the individuals and the resources will be adaptively changed. Experiments on distributed resources are carried out to evaluate the effect of the load balance based allocation strategy. The results indicate that the proposed load balance strategy is able to significantly reduce the runtime.

## CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence** → *Search methodologies*;

## KEYWORDS

Power electronic circuit, distributed differential evolution, expensive fitness evaluation, load balance

**ACM Reference format:**

N. Ma, X. F. Liu, Z. H. Zhan, J. H. Zhong, and J. Zhang. 2017. Load Balance Aware Distributed Differential Evolution for Computationally Expensive Optimization. In *Proceedings of ACM GECCO'17*, 2 pages
DOI: 10.1145/3067695.3075602


*GECCO '17 Companion, July 15-19, 2017, Berlin, Germany*
ACM 978-1-4503-4939-0/17/07.
http://dx.doi.org/10.1145/3067695.3075602

## 1 INTRODUCTION

Nowadays, computationally expensive optimization problems often challenge the application of evolutionary algorithms (EAs) due to the long runtime. For example, in a power electronic circuit (PEC) design, there are multiple components, such as inductors, resistors, and capacitors to be optimized [1]. The components relationship is complex and the objective function is difficult for optimization. To solve the PEC, EAs such as genetic algorithm (GA) [2] and differential evolution (DE) [3] are applied and have shown good performance. Nevertheless, the expensive fitness evaluation causes a long computation time and limits the application of EAs on PEC. The same challenge emerges when extending EAs to other computationally expensive optimization problems in complex environments like big data and real-time. By exploiting the population-based model, EAs are able to be naturally parallelized and distributed [4]. A distributed DE based on distributed computational resource has been designed to reduce the computational time [5]. The population are evenly allocated to computational nodes and evaluated concurrently. However, distributed resources (e.g., the cloud virtual machine resources) are often shared by different applications. Moreover, different resources have different computational speed, such as heterogeneous processors, or homogeneous processors with different loads. The processor with high load often requires longer execution time for the same task compared with the one with light load. Therefore, uniform allocation strategy is simple but not efficient. The slow-performance resource limits the parallelism speedup. To improve the parallel efficiency and further reduce the computational time, we consider the compute capacity of each resource and design a distributed mode based on load balance.

In this paper, we propose a distributed DE (DDE) with load balance strategy on distributed resources to further speed up the optimization for expensive fitness evaluation problems. The candidate solutions in the whole population are allocated to the processors based on their loads. Herein, we use the CPU utilization information as load indicator. The processors with higher load will be assigned fewer solutions (individuals) for calculating the fitness values while the lighter processors are assigned more solutions (individuals).

The rest of the paper is organized as follows. In Section 2, the proposed algorithm is described. In Section 3, the experiments are performed and the test results are reported. Finally, conclusions are summarized in Section 4.

## 2 LOAD BALANCE BASED DDE

The DDE is implemented in a master-slave parallel computing topology. The master node controls the evolutionary process of the DE algorithms while the distributed slave nodes evaluate the solutions in parallel. In each generation, the master node generates a new population and then collects the load information from each slave node. Then the master node assigns all the individuals of the population to the slave nodes based on their load information. Assume that there are $T$ slave nodes and the CPU utilization of slave $i$ is $l_i$, then the number of solutions (individuals) $N_i$ allocated to slave node $i$ is calculated by (1) and (2):

$$S_i = \frac{1.0 - l_i + \xi}{\sum_{k=1}^{T}(1.0 - l_k + \xi)} PS \quad (1)$$

$$N_i = \lfloor S_i \rfloor \quad (2)$$

where $\xi$=0.01 is used to avoid a zero denominator in (1), and $PS$ is the population size. If there are remaining unassigned solutions, the slave nodes are sorted by fractional part of $S_i$ and then allocate one more solution to the slave nodes in order until there are no remaining individuals. The slave nodes receive the tasks (e.g. $N_i$ solutions) for fitness evaluation and return the corresponding fitness values to the master node. This way, the topology related to the individuals and the resources will be adaptively changed according to the load information of different resources.

## 3 EXPERIMENTS AND RESULTS

The DDE uses the same DE operators as the one in [5]. The parameters are set as population size $PS$=100, $F$=0.5, and $CR$=0.1. In all test cases, the maximal fitness evaluations (FES) number is set as $1.5\times 10^4$. The average runtime on 10 independent runs are reported. Three cases with 32, 48, and 64 cores for DDE are carried out to test the proposed load balance strategy. In each case, $n$ cores are given. The first core named $c_0$ is performed as master node and the remaining $n$-1 cores named as $c_1$ to $c_{n-1}$ are performed as slave nodes. In the experiments, the cores are homogeneous with the same calculation capacity. To simulate the different loads on different cores, we run multiple compute-intensive applications on cores $c_8$-$c_{15}$ and make them present high load and CPU utilization. The other cores are in idle state. The sequential implementation executed on only one core is also performed for comparison. All the simulation experiments are carried out on the cluster with 8 PCs with Intel i7-4790 CPU with 8 logical cores and frequency of 3.60 GHz.

The proposed load balance based DDE are compared with the DDE with uniform allocation strategy. In uniform allocation strategy, each core $c_1$-$c_{n-1}$ is allocated with $PS$/($n$-1) solutions firstly, and the remaining unassigned individuals are allocated to $c_1$-$c_{n-1}$ one by one in order. The results are reported in Table 1, where the "uniform" column represents the results of uniform allocation strategy and the "load balance" column represents the results of the proposed algorithm by using the CPU utilization information for the load balance allocation.

Table 1 shows that the sequential program requires 780.31 seconds for execution. The parallel versions significantly reduce the time. Compared with the uniform allocation strategy, the CPU utilization based load balance strategy use less time in each test case. It shows that the proposed allocation strategy can improve the parallel efficiency. It is interesting to see that the runtimes "Uniform" used in cases with 48 and 64 cores are both approximate 93 seconds. It is because that the runtimes for these two cases with uniform allocation strategy is limited by the fitness evaluation time of solutions on high load cores $c_8$-$c_{15}$. However, the CPU utilization based allocation strategy can adaptively reduce the number of allocation solutions on high load cores and reduce the runtime. The load balance based allocation strategy performs high parallel efficiency for PEC problem.

**Table 1: Runtimes for DDE with Uniform Allocation Strategy and Load Balance based Allocation Strategy**

| Number of cores $n$ | Time(second) | |
|---|---|---|
| | Uniform DDE | Load Balance DDE |
| 1 | 780.31 | - |
| 32 | 125.71 | **118.88** |
| 48 | 93.70 | **84.53** |
| 64 | 93.99 | **67.69** |

## 4 CONCLUSIONS

This paper tracks the parallel efficiency problem on distributed heterogeneous resources with different computation speed. The resource with high load often needs more runtime. We propose a load balance based DDE to further reduce the runtime for PEC problem. The number of candidate solutions assigned to the distributed resources is determined by their load information. Simulation experiments on distributed resources with different loads are conducted to evaluate the effect the proposed allocation strategy. The experimental results show that the proposed load balance based DDE can improve the computational speed.

## 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, GDUPS (2016), the NSFC Key Program with No. 61332002, and the Fundamental Research Funds for the Central Universities.

## REFERENCES

[1] Z. Chen, J. M. Guerrero, and F. Blaabjerg. 2009. A review of the state of the art of power electronics for wind turbines. *IEEE Trans. Power Electron.* 24, 8 (2009), 1859-1875.

[2] J. Zhang, H. Chung, W. L. Lo, S. Y. R. Hui, and A. Wu. 2001. Implementation of a decoupled optimization technique for design of switching regulators using genetic algorithm. *IEEE Trans. Power Electron.* 16, 6 (2001), 752-763.

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

[4] Z. H. Zhan, X. F. Liu, H. X. Zhang, Z. T. Yu, J. Weng, Y. Li, T. L. Gu, and J. Zhang. 2017. Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version. *IEEE Transactions on Parallel and Distributed Systems*. 28, 3 (2017), 704-716

[5] X. F. Liu, Z. H. Zhan, J. H. Lin, and J. Zhang. 2016. Parallel differential evolution based on distributed cloud computing resources for power electronic circuit optimization. In *Proc. GECCO' 16*. ACM, New York, NY, 117-118.