Attract-Repulse Fireworks Algorithm and its CUDA Implementation Using Dynamic Parallelism

Ke Ding, Key Laboratory of Machine Perception (MOE), Peking University, Beijing, China & Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing, China

Ying Tan, Key Laboratory of Machine Perception (MOE), Peking University, Beijing, China & Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing, China

ABSTRACT

Fireworks Algorithm (FWA) is a recently developed Swarm Intelligence Algorithm (SIA), which has been successfully used in diverse domains. When applied to complicated problems, many function evaluations are needed to obtain an acceptable solution. To address this critical issue, a GPU-based variant (GPU-FWA) was proposed to greatly accelerate the optimization procedure of FWA. Thanks to the active studies on FWA and GPU computing, many advances have been achieved since GPU-FWA. In this paper, a novel GPU-based FWA variant, Attract-Repulse FWA (AR-FWA), is proposed. AR-FWA introduces an efficient adaptive search mechanism (AFW Search) and a non-uniform mutation strategy for spark generation. Compared to the state-of-the-art FWA variants, AR-FWA can greatly improve the performance on complicated multimodal problems. Leveraging the edge-cutting dynamic parallelism mechanism provided by CUDA, AR-FWA can be implemented on the GPU easily and efficiently.

Keywords: *Compute Unified Device Architecture (CUDA), Dynamic Parallelism, Fireworks Algorithm (FWA), GPU Computing, Swarm Intelligence Algorithms (SIAs)*

1. INTRODUCTION

Fireworks Algorithm (FWA) is a novel swarm intelligence algorithm (SIA) under active research. Inspired by the explosion process of fireworks, FWA was originally proposed for solving optimization problems (Tan, 2010). Comparative study shows that FWA is very competitive with respect to real-parameter problems (Tan, 2013). FWA has been successfully applied to many

DOI: 10.4018/IJSIR.2015040101

scientific and engineering problems, such as non-negative matrix factorization (Janecek, 2011), digital filter design (Gao, 2011), parameter optimization (He, 2013), document clustering (Yang, 2014), and so forth. New mechanisms and analyses are actively proposed to further improve the performance of FWA (Li, 2014; Zheng, 2014).

Although FWA, as well as other SIAs, has achieved success in solving many real-world problems where conventional arithmetic and numerical methods fail, it suffers from the drawback of intensive computation which greatly limits its applications where function evaluation is time-consuming.

Facing technical challenges with higher clock speeds infixed power envelope, modern computer systems increasingly depend on adding multiple cores to improve the performance (Ross, 2008). Initially designed for addressing highly computational graphics tasks, the Graphics Processing Unit (GPU), from its inception, has many computational cores and can provide massive parallelism (with thousands of cores) at a reasonable price. As the hardware and software for GPU programming grow mature (Kirk, 2010; Munshi, 2011), GPUs have become popular for general purpose computing beyond the field of graphics processing, and great success has been achieved in various applications, from embedded systems to high performance supercomputers (AMD, 2015; NVIDIA, 2015a; Owens, 2007).

Based on interactions within population, SIAs are naturally amenable to parallelism. SIAs' such intrinsic property makes them very suitable to run on the GPU in parallel, thus gain remarkable performance improvement. In effect, GPUs have been utilized to accelerated SIAs from the first days of GPU computing, and significant progress has been achieved along with the emergence of high-level programming platforms such as CUDA (Compute Unified Device Architecture) and OpenCL (Open Computing Language) (Zhou, 2009; Zhu, 2009). In the past few years, different implementations of diverse SIAs were proposed. Targeting on GPUs of various architectures and specifications, many techniques and tricks were introduced (Tan, 2015).

The first GPU-based FWA, named GPU-FWA, was proposed in 2013 which targets on GPUs of Fermi Architecture (Ding, 2013). GPU-FWA modifies the original FWA to suit the particular architecture of the GPU. It does not need special complicated data structure, thus making it easy to implement; meanwhile, it can make full use of the great computing power of GPUs. In the last few years, however, many advances have been achieved for both FWA and GPU computing. More dedicated and efficient implementations are possible. In this paper, a novel GPU-based FWA variant, Attract-Repulse FWA (AR-FWA), is proposed and discussed in detail. AR-FWA introduces an efficient adaptive firework search strategy and a novel mutation mechanism for spark generation. Thanks to the dynamic parallelism provided by CUDA, AR-FWA can be implemented on the GPU very easily and efficiently.

The remainder of this paper is organised as follows. In Section 2, the related work, Fireworks Algorithm (FWA) and General Purpose Computing on the GPU (GPGPU), are presented briefly. Section 3 discusses the proposed algorithm, Attract-Repulse Fireworks Algorithm (AR-FWA). The Adaptive Firework Search (AFW Search) and Non-uniform Mutation are presented in detail. Section 4 describes how AR-FWA can be implemented on the GPU using dynamic parallelism. Key kernel codes are also given out in this section. The experiments and analyses are given in Section 5. The performance of Non-uniform mutation against uniform mutation is studied, as well as AR-FWA against the state-of-the-art FWA variants and the speedup on the basis of extensive experiments. Finally, we conclude the discussion in Section 6.