称号及び氏名　　博士(理学)　　Md Maruf Hussain

学位授与の日付　令和 2 年 6 月 30 日

論　　文　　名　　GPU-based Parallel Single and Multi-objective Particle Swarm Optimization for Large Swarms and High Dimensional Problems
単目的および多目的の高次元最適化問題を対象とした大規模粒子群最適化の GPU を用いた並列化

論文審査委員　　主査　　藤本　典幸
副査　　宇野　裕之
副査　　瀬田　和久

GPU-based Parallel Single and Multi-objective Particle Swarm Optimization for Large Swarms and High Dimensional Problems

Abstract

The Particle Swarm Optimization (PSO) algorithm has been first introduced by Eberhart and Kennedy in 1995, which is one of the most important population based nondeterministic optimization algorithms for single objective optimization problems. Since then, many successful applications of PSO have been reported. In many of those applications, the PSO algorithm has shown several advantages over other swarm intelligence based optimization algorithms due to its robustness, efficiency and simplicity. Moreover, compared to other stochastic algorithms, it usually requires less computational effort and resources.

The PSO algorithm maintains a swarm of particles, where each of which represents a potential solution. Here a swarm can be identified as the population and a particle as an individual. In a PSO system, each particle flows through a multidimensional search space and adjusts its position based on its own experience with neighboring particles. On a CPU, this process is implemented based on task scheduling into serial processing, whereas on a GPU, many particles can reach to their positions simultaneously, which improves the PSO efficiency significantly. In recent years, a GPU becomes a very popular platform for the realization of parallel computing, mainly due to changes in architecture and development of CUDA and OpenCL languages. Previously reported works have shown that the PSO implementation on a GPU provides a better performance than CPU-based implementations which makes us interested in this study.

At first, we implemented a good implementation for the Standard Particle Swarm Optimization (SPSO) on a GPU based on the CUDA architecture, which uses atomic

function, a fast pseudorandom number generator, coalescing memory access. The algorithm is evaluated on a suite of well-known benchmark optimization functions. The experiments are performed on an NVIDIA GeForce GTX 980 GPU and a single core of 3.20 GHz Intel Core i5 4570 CPU and the test results demonstrate that the GPU algorithm runs about maximum 170 times faster than the corresponding CPU algorithm. Therefore, this proposed algorithm can be used to improve required time to solve optimization problems. After that, we conducted experiments for testing the effect of the Pseudorandom Number Generators on the SPSO on a GPU. By using a single step TausStep of the combined Tausworthe generator, the proposed parallel implementation of SPSO provides up to 307 times speedup compared to a serial SPSO implementation. Speedup is greatly accelerated for high dimensional problems, large particles and complex benchmark functions.

The success of the PSO algorithm as a single objective optimizer has motivated us to extend its use in other areas. One of such areas is multi-objective optimization. Multi-objective optimization problems (MOOPs) are very common in real-world optimization fields, where the objectives to be optimized are normally in conflict with each other. Moore and Chapman proposed the first extension of the PSO strategy for solving MOOPs (Multi-Objective PSO, MOPSO) in an unpublished manuscript in 1999. With big data becoming more important as time goes by, the necessity for faster methods is growing. The previous implementations of serial and parallel cases of MOPSO do not meet the requirements of big data. In addition, those implementations could only handle a limited number of dimensions. The necessity for a better method is sorely needed.

This thesis paper presents a new GPU-parallelized implementation of MOPSO (GPU MOPSO) based on a master-slave model for large swarms and high dimensional optimization problems. This paper also presents a new serial implementation of MOPSO (CPU MOPSO). Our CPU program uses a single core only although our CPU has multiple cores. Our CPU MOPSO achieves faster performance for large swarms and high dimensional optimization problems. The experimental results show that the proposed GPU MOPSO increases the processing speed compared to previously proposed approaches on a GPU based on the CUDA architecture. The proposed parallel implementation of MOPSO using a master-slave model provides up to 157 times speedup compared to the corresponding CPU implementation. Here, we investigate a large number of iterations to reach good nondominated solutions which achieve good Pareto fronts. Pareto fronts of both CPU MOPSO and GPU MOPSO implementations match very closely to the true Pareto fronts. Performance of MOPSO is dependent upon an archiving technique. We propose a simple parallel archiving technique which significantly speeds up the process. Our serial archiving technique is the same as the parallel archiving except that it is executed in serial. In our GPU MOPSO, the used PRNG and coalescing memory access have a positive impact which improve computational time.

In the literature, several models for parallel MOPSO have previously been proposed. Some of these models are suited for costly platforms. For example, the island model is suitable for clusters and grids. The diffusion model of multi-objective evolutionary algorithms is also suitable for another costly platform, massively parallel processors. In contrast, there are models for more affordable platforms. The master-slave model and the hierarchical model are both suitable for GPUs, but the hierarchical model tends to be slower than the master-slave model.

The doctoral thesis is organized as follows. In Chapter 1, we introduce our implementations. In Chapter 2, we sketch out briefly PSO, SPSO, MOOPs, MOPSO, GPU computing, an overview of CUDA architecture, coalescing memory access, random number generators, Thrust library, CUB library and parallel models. In Chapter 3, we present our CUDA Implementation of SPSO. In Chapter 4, we conduct experiments for testing the effect of ten Pseudorandom Number Generators on the SPSO on a GPU. In Chapter 5, we provide our MOPSO implementations on a CPU and a GPU, analyze experimental results and compare our implementation with the previous implementation in terms of execution time and speedup. Finally, in Chapter 6, we give some concluding remarks and point out directions for future work.

List of Publications

Refereed Journal Paper

1. GPU-based Parallel Multi-objective Particle Swarm Optimization for Large Swarms and High Dimensional Problems, Md. Maruf Hussain and Noriyuki Fujimoto, Parallel Computing, 19 pages, DOI: 10.1016/j.parco.2019.102589 (2020) (掲載決定).

Refereed International Conference Papers

1. Effect of the Pseudorandom Number Generators on the Standard Particle Swarm Optimization on a GPU, Md. Maruf Hussain and Noriyuki Fujimoto, Proc. of the 2018 International Conference on Computational Science and Computational Intelligence (CSCI), pp.295-300, DOI: 10.1109/CSCI46756.2018.00064 (Las Vegas, USA, 2018).
2. Parallel Multi-Objective Particle Swarm Optimization for Large Swarm and High Dimensional Problems, Md. Maruf Hussain and Noriyuki Fujimoto, Proc. of the 2018 IEEE Congress on Evolutionary Computation (CEC), pp.1-10, DOI: 10.1109/CEC.2018.8477848 (Rio de Janeiro, Brazil, 2018).
3. A CUDA Implementation of the Standard Particle Swarm Optimization, Md. Maruf Hussain, Hiroshi Hattori, and Noriyuki Fujimoto, Proc. of 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp.219-226, DOI: 10.1109/SYNASC.2016.043 (Timisoara, Romania, 2016).

学位論文審査結果の要旨

学位論文題目

GPU-based Parallel Single and Multi-Objective Particle Swarm Optimization for Large Swarms and High Dimensional Problems

（単目的および多目的の高次元最適化問題を対象とした大規模粒子群最適化の GPU を用いた並列化）

単一の実数値関数の最適化を行うメタヒューリスティクスとして 1995 年に Eberhart と Kennedy により提案された粒子群最適化（Particle Swarm Optimization、以降 PSO）アルゴリズムは、単目的最適化問題を解く多点探索アルゴリズムの中で最も重要なもののひとつである。PSO は多くの応用で、堅牢性、効率、単純さの点で他の群知能に基づく手法よりも優れていることが報告されている。単目的最適化での PSO の成功を受けて、PSO に対する様々な拡張が試みられたが、そのうちのひとつが多目的最適化への拡張である。多目的最適化問題は複数の実数値関数を同時に最適化する問題で、最適化の実応用問題でよく現われるものである。多目的最適化では、各関数は、ひとつの関数の値を改善しようとすると他の関数の値を悪化させるという競合状態にあるため、多目的最適化問題を解くためにはこの競合への考慮が必要となる。

本論文の主要な成果は次の通りである。

(1) 単目的最適化問題に対する標準 PSO アルゴリズムを GPU（Graphics Processing Unit）を用いて並列処理する手法を開発し、提案手法を実装した GPU プログラムが CPU の 1 コア上で標準 PSO アルゴリズムを実行する場合に比べて最大 170 倍高速であることを評価実験により示している。

(2) 確率的探索を行う標準 PSO において、用いる疑似乱数生成アルゴリズムのちがいが性能に与える影響を調べるため、10 種類の疑似乱数生成アルゴリズムを比較する実験を行い、Tausworthe のアルゴリズムが実行時間と得られる解の品質の点から最も優れていることを明らかにしている。

(3) 多目的最適化問題を解くための比較的単純で並列化が容易な新しい PSO 拡張手法を提案し、提案手法を実装した GPU プログラムが CPU の 1 コア上で実行する場合に比べて最大 157 倍高速であることを評価実験により示している。

以上のように、本研究では単目的および多目的の最適化問題に対する PSO を従来手法より高速化する手法の開発に成功している。この手法は特に高次元問題を大規模な粒子群を用いて解く場合に有効であり、多くの応用に適用できる。以上のことから、本委員会は本論文の審査、最終試験の結果に基づき、Md Maruf Hussain 氏に博士（理学）の学位を授与することを適当と認める。

学位論文審査委員会

主査　藤本　典幸

宇野　裕之

瀬田　和久