A Scalable MapReduce-enabled Glowworm Swarm Optimization Approach for High Dimensional Multimodal Functions

Ibrahim Aljarah, Department of Business Information Technology, The University of Jordan, Amman, Jordan
Simone A. Ludwig, Department of Computer Science, North Dakota State University, Fargo, ND, USA

ABSTRACT

Glowworm Swarm Optimization (GSO) is one of the common swarm intelligence algorithms, where GSO has the ability to optimize multimodal functions efficiently. In this paper, a parallel MapReduce-based GSO algorithm is proposed to speedup the GSO optimization process. The authors argue that GSO can be formulated based on the MapReduce parallel programming model quite naturally. In addition, they use higher dimensional multimodal benchmark functions for evaluating the proposed algorithm. The experimental results show that the proposed algorithm is appropriate for optimizing difficult multimodal functions with higher dimensions and achieving high peak capture rates. Furthermore, a scalability analysis shows that the proposed algorithm scales very well with increasing swarm sizes. In addition, an overhead of the Hadoop infrastructure is investigated to find if there is any relationship between the overhead, the swarm size, and number of nodes used.

KEYWORDS

Big Data, MapReduce, Multi-modal Functions, Optimization

1. INTRODUCTION

Optimization is the process of searching for the optimum solution from a set of candidate solutions based on specific performance criteria. There are many different optimization techniques that exist such as evolutionary computation and swarm intelligence.

Swarm intelligence (Engelbrecht, 2007) is an optimization technique that is inspired by processes of natural swarms by simulating the behavior of natural swarms such as ant colonies, flocks of birds, bacterial growth, and schools of fishes. The swarm behavior is based on the interactions between the swarm members by exchanging local information to reach the target, for example, locating the food source. However, instead of using the centralization concept in the swarm structure, all swarm members are equally engaged to achieve the main objective.

DOI: 10.4018/IJSIR.2016010102

Copyright © 2016, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
There are many swarm intelligence methodologies such as Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1997) which is inspired by bird flocks searching for optimal food sources, where the direction in which a bird moves is influenced by its current movement, the best food source it ever experienced, and the best food source any bird in the flock ever experienced. Ant Colony Optimization (ACO) (Stuetzle, 2009) is based on the behavior of ants searching for the optimal shortest path between a food source and the colony using the pheromone they leave while traveling through the paths. Bee Colony Optimization (BCO) (Wong et al., 2008) mimics the food foraging behavior of honeybee colonies using a combination of local and global searches.

Glowworm Swarm Optimization (GSO) (Krishnanand and Ghose, 2005) is inspired by the behavior of glowworms or lighting worms. These glowworms control the emission of their light and use it for different objectives such as attracting other worms during the breeding season. GSO’s implementation simplicity and the small number of required parameters to be tuned (Krishnanand and Ghose, 2005; Krishnanand and Ghose, 2008; Krishnanand and Ghose, 2009a) make it more applicable to be used in many application areas such as hazard sensing in ubiquitous environments (Krishnanand and Ghose, 2008), mobile sensor networks and robotics (Krishnanand and Ghose, 2005), and data clustering (Aljarah and Ludwig, 2013a).

Most swarm intelligence algorithms solve an optimization problem with a global solution that is easier than finding multiple solutions. However, GSO is differentiated by its ability to perform a simultaneous search of multiple solutions, and is therefore the perfect solution for solving multimodal functions. Multimodal functions (Barrera and Coello, 2009) are functions that have multiple peaks (local maxima) with different or equal objective values. The optimization of multimodal functions aims to find all maxima based on some constraints. High dimensional spaces increase the peaks count, which leads to each function evaluation requiring longer execution times in order to find optimal target peaks. Solving multimodal functions requires the swarm to have the ability of dividing itself into separated groups. Moreover, the number of individuals has to be increased to share more local information for locating more peaks. To tackle the high computation time for these situations, the algorithm must be parallelized in an efficient way to find the peaks in an acceptable amount of time.

An efficiently parallelized algorithm has a shorter execution time; however, depending on the nature of the algorithm some problems can be encountered such as communication inefficiency, or unfair load balancing, which make the process of scaling of the algorithm to large numbers of processors very difficult. Moreover, another cause of reduced scalability of the algorithm is node failure. Therefore, the parallel algorithm should handle large amounts of data and scale well with increasing numbers of compute nodes while maintaining high quality results.

One programming model that was developed by Google is MapReduce. MapReduce has recently become a very favorable model for parallel processing compared to the message passing interface parallelization technique (Snir et al., 1995). The strength of the MapReduce methodology is that no knowledge of parallel programming is required to perform the parallelization process. In addition, there are special characteristics that distinguish the MapReduce methodology such as fault-tolerance, load balancing, and data locality.

Apache Hadoop MapReduce (Apache, 2011) is another implementation of MapReduce besides Google’s implementation and Disco (Disco, 2011). MapReduce is a highly scalable and most applicable model when considering a data-intensive task. When computing resources have restrictions on multiprocessor and large shared-memory hardware, MapReduce is the most applicable and is therefore adopted by many companies in industry (e.g., Facebook (Facebook, 2011), and Yahoo (Yahoo, 2011)). In academia, researchers benefit from MapReduce for scientific computing, such as in the areas of Bioinformatics (Gunarathne et al., 2010), and Geosciences (Krishnan et al., 2010),
Unit Commitment by Evolving Ant Colony Optimization
www.igi-global.com/chapter/unit-commitment-evolving-ant-colony/65814?camid=4v1a

Analyzing Community Deliberation and Achieving Consensual Knowledge in SAM
www.igi-global.com/chapter/analyzing-community-deliberation-achieving-consensual/65796?camid=4v1a