July 2017

High-level Parallel Implementation of Swarm Intelligence-based Optimization Algorithms with Algorithmic Skeletons

Fabian WREDE [a], Breno Augusto DE MELO MENEZES [a],
Luis Filipe DE ARAUJO PESSOA [a], Bernd HELLINGRATH [a],
Fernando BUARQUE DE LIMA NETO [b] and Herbert KUCHEN [a]
[a] *University of Muenster, Germany*
[b] *University of Pernambuco, Brazil*

Abstract Swarm Intelligence (SI)-based metaheuristics are frequently used to solve complex optimization problems, which are too hard to be solved by classic exact algorithms. Inspired by nature, SI particles move through a search space in pursuit of good solutions. Even using SI, solving some large problems still takes a lot of time, e.g., due to the high number of dimensions and large search spaces. In order to overcome this, parallel implementations of SI algorithms have been investigated. They are typically based on low-level approaches for parallelism, such as MPI, OpenMP, and CUDA, which are tedious and error-prone to use. To overcome these issues, frameworks for high-level parallel programming such as the Muenster Skeleton Library (Muesli) can be used. We show how two SI algorithms, namely Particle Swarm Optimization (PSO) and Fish School Search (FSS), can be implemented in Muesli easily. Experimental results demonstrate the obtained performance and good scalability.

Keywords. high-level parallel programming, algorithmic skeletons, swarm intelligence metaheuristics, particle swarm optimization, fish school search.

1. Introduction

Metaheuristics are often used to compute optimal or, at least, good solutions for hard optimization problems. Swarm Intelligence (SI) algorithms are population-based metaheuristics inspired by nature. Individuals move through a search space in order to find a good solution. The algorithms differ in the way the individuals move, share and store information [20].

Inside the vast set of SI algorithms, Particle Swarm Optimization (PSO) [9] is one of the most important. Inspired by the mechanisms of a bird flocking, it has been designed for optimization problems in a continuous domain. Also in the SI family, Fish School Search (FSS) [5] is inspired by the behavior of a fish school looking for food. The main characteristic of FSS is its ability to provide automatic switching between exploration and exploitation along the whole search process due to its contraction or expansion bias influencing individual fish movements. Although both approaches belong to the SI family, they differ in their operators and the way individuals move and take