Chapter 1
An Optimization Algorithm Based on Brainstorming Process

Yuhui Shi
Xi'an Jiaotong-Liverpool University, China

ABSTRACT

In this chapter, the human brainstorming process is modeled, based on which two versions of a Brain Storm Optimization (BSO) algorithm are introduced. Simulation results show that both BSO algorithms perform reasonably well on ten benchmark functions, which validates the effectiveness and usefulness of the proposed BSO algorithms. Simulation results also show that one of the BSO algorithms, BSO-II, performs better than the other BSO algorithm, BSO-I, in general. Furthermore, average inter-cluster distance D_c and inter-cluster diversity D_e are defined, which can be used to measure and monitor the distribution of cluster centroids and information entropy of the population over iterations. Simulation results illustrate that further improvement could be achieved by taking advantage of information revealed by D_c, which points at one direction for future research on BSO algorithms.

INTRODUCTION

Many real-world applications can be represented as optimization problems of which algorithms are required to have the capability to search for optimum. Originally, these optimization problems were mathematically represented by continuous and differentiable functions so that algorithms such as hill-climbing algorithms can be designed and/or utilized to solve them. Traditionally, these hill-climbing like algorithms are single-point based algorithms such as gradient decent algorithms which move from the current point along the direction pointed by the negative of the gradient of the function at the current point. These hill-climbing algorithms can find solutions quickly for unimodal problems, but they have the problems of being sensitive to initial search point and being easily trapped into local optimum for nonlinear multimodal problems. Furthermore, these mathematical functions need to be continuous and differentiable, which instead greatly narrows the range of real-world problems that can be solved by hill-climbing algorithms. Recently, evolutionary algorithms have been designed and utilized to solve optimization problems. Different from traditional single-point based algorithms such as gradient decent algorithms, each evolutionary algorithm is a population-based algorithm,

DOI: 10.4018/978-1-4666-6328-2.ch001

which consists of a set of points (population of individuals). The population of individuals is expected to have high tendency to move towards better and better solution areas iteration over iteration through cooperation and/or competition among themselves. There are a lot of evolutionary algorithms out there in literature. The most popular evolutionary algorithms are evolutionary programming (Fogel, 1962), genetic algorithm (Holland, 1975), evolution strategy (Rechenberg, 1973), and genetic programming (Koza, 1992), which were inspired by biological evolution. In evolutionary algorithms, population of individuals survives into the next iteration. Which individual has higher probability to survive is proportional to its fitness value obtained according to some evaluation function. The survived individuals are then updated by utilizing evolutionary operators such as mutation operator, crossover operator, *etc*. In evolutionary programming and evolution strategy, only the mutation operation is employed, while in genetic algorithms and genetic programming, both the mutation operation and crossover operation are employed. The optimization problems to be optimized by evolutionary algorithms do not need to be mathematically represented as continuous and differentiable functions. They can be represented in any form. Only requirement for representing optimization problems is that each individual can be evaluated as a value called fitness value. Therefore, evolutionary algorithms can be applied to solve more general optimization problems, especially those that are very difficult, if not impossible, for traditional hill-climbing algorithms to solve.

Recently, another kind of algorithms, called swarm intelligence algorithms, is attracting more and more attentions from researchers. Swarm intelligence algorithms are usually nature-inspired optimization algorithms instead of evolution-inspired optimization algorithms such as evolutionary algorithms. Similar to evolutionary algorithms, a swarm intelligence algorithm is also a population-based optimization algorithm. Different from the evolutionary algorithms, each individual in a swarm intelligence algorithm represents a simple object such as ant, bird, or fish, *etc*. So far, a lot of swarm intelligence algorithms have been proposed and studied. Among them are particle swarm optimization(PSO) (Eberhart & Shi, 2007; Shi & Eberhart, 1998), ant colony optimization algorithm(ACO) (Dorigo, 1996), bacterial forging optimization algorithm(BFO) (Passino, 2010), firefly optimization algorithm (FFO) (Yang, 2008), bee colony optimization algorithm (BCO) (Tovey, 2004), artificial immune system (AIS) (Castro, 1999), fish school search optimization algorithm(FSO) (Bastos-Filho, 2008), shuffled frog-leaping algorithm (SFL) (Enusuff, 2006), intelligent water drops algorithm (IWD) (Shah, 2009), to just name a few.

In a swarm intelligence algorithm, an individual represents a simple object such as birds in PSO, ants in ACO, bacteria in BFO, *etc*. These simple objects cooperate and compete among themselves to have a high tendency to move toward better and better search areas. As a consequence, it is the collective behavior of all individuals that makes a swarm intelligence algorithm to be effective and efficient in problem optimization.

For example, in PSO, each particle (individual) is associated with a velocity. The velocity of each particle is dynamically updated according to its own historical best performance and its companions' historical best performance. All the particles in the PSO population fly through the solution space in the hope that particles will fly towards better and better search areas with high probability.

Mathematically, the updating process of the population of individuals over iterations can be looked as a mapping process from one population of individuals to another population of individuals from one iteration to the next iteration, which can be represented as $P_{t+1} = f(P_t)$, where P_t is the population of individuals at the iteration t, $f()$ is the mapping function. Different evolutionary algorithm or swarm intelligence algorithm has a different mapping function. Through the mapping

Related Content