Chapter 11
Ant Colony Programming:
Application of Ant Colony System to Function Approximation

Mariusz Boryczka
University of Silesia, Poland

ABSTRACT

Automatic programming is the method in which a computer program is constructed automatically based on the specification of goals which are to be realized. This chapter describes one of the methods for automatic function approximation (as a form of automatic programming) – ant colony programming (ACP). It is based on ant colony system (ACS) as a new method for solving approximation problems. While solving these problems by ACP two approaches are used: the expression approach and the program approach. Several improvements of this method are presented, including the elimination of introns, the use of a structure similar to the candidate list introduced in ACS, and parameter-tuning. The chapter first describes ACS and introduces the problem of symbolic regression. Then, ACP is defined. After that, improvements of ACP are presented. The main objective of the chapter is to give an overview of the published results of studies carried out on ACP, while at the same time present a new idea in the process of parameter-tuning.

INTRODUCTION

Automatic programming makes it possible to avoid the tedious task of creating computer programs. In automatic programming, the program is obtained by first specifying the goals which are to be realized by the program. Then, based on this specification, the program is constructed automatically.

This chapter describes the idea of one of the methods for automatic function approximation (as a form of automatic programming) – Ant Colony Programming (ACP). ACP is based on Ant Colony System (ACS). It has been introduced in Boryczka (2002), Boryczka & Czech (2002) and Boryczka,
Ant Colony Programming

Czech & Wieczorek (2003), and was mentioned in Engelbrecht (2005) as a new approach for solving approximation problems. While solving these problems by ACP two approaches are used. In the first, the expression approach, an approximating function in the form of an arithmetic expression written in the Polish (prefix) notation, is constructed. In the second, the program approach, the desired approximating function is built as a computer program, i.e. a sequence of assignment instructions which evaluates the function. In many cases, this method may outperform other ones (see Boryczka (2006)).

There are several similar proposals on the use of ACS for symbolic regression, but they deal with this problem in a different way. In Ant Programming (Roux, 2000; Chen, 2004), each ant builds and modifies expressions using a tree structure. Each node of the tree stores pheromone for all symbols (terminals and functions) and expressions are constructed based on the amount of pheromone for individual symbols. A second method, AntTAG (Abbass, 2002), combines ACS with Grammar-Guided Genetic Programming (GGGP). This is a hybrid approach which takes advantages of Tree-adjunct Grammar (TAG) representation and the search strategy of ACS. It uses a pheromone table, which is a two-dimensional table with entries containing the amount of pheromone deposited by ants while they construct their solutions. Finally, the Generalized Ant Programming (GAP) (Keber, 2002) uses a formal language specified by the context-free grammar. Expressions of the language are paths visited by the ants (sequences of terminal symbols and the corresponding derivation steps), where ants deposit their pheromone.

The chapter presents ACP as well as several improvements of the standard method. One of them is eliminating the so called introns which arise while generating solutions (Boryczka, 2005). As introns are the excerpts (sequences of symbols or instructions) which do not influence the quality of approximation, they should be eliminated, because they complicate the structure of solutions and increase the evaluation time.

The second improvement is the use of the specimen list, a structure similar to the candidate list, introduced in ACS. It allows the reduction of the algorithm’s execution time and improves the quality of the results (Boryczka, 2008).

Another problem lies in the process of parameter-tuning of ACP. There are two kinds of parameters: quantitative parameters inherited from ACS, and qualitative parameters, characteristic of the described method. All of them should be tuned as fine as possible to obtain good (maybe exact) solutions.

The chapter gives a summary of the results of studies carried out on ACP (Boryczka, 2002; Boryczka & Czech, 2002; Boryczka, Czech & Wieczorek, 2003; Boryczka, 2005; Boryczka, 2006; Boryczka, 2008), while at the same time presents a new idea in the process of parameter-tuning. The organization of the chapter is as follows. First, ACS is described and the problem of symbolic regression is introduced. Then, ACP is defined as a method adapting ACS to solve the problem of function approximation (symbolic regression). After that, improvements of ACP are presented. They are: eliminating of introns, the use of the specimen list and finally – automatic tuning of the parameters of ACP. Conclusions recapitulate the chapter.

ANT COLONY SYSTEM

Ant Colony System (ACS), nowadays a part of the ant colony optimization metaheuristic (Cordón, Herrera & Stützle, 2002), is derived from the research on systems inspired by the behavior of real ants. It was originally proposed by Dorigo, Maniezzo and Colomi (1991) as an ant system for solving the
Related Content

CAM: Clustering Algorithms for Multimodal WSN
www.irma-international.org/article/cam/103843/

VPRS-Based Group Decision-Making for Risk Response in Petroleum Investment
www.irma-international.org/chapter/vprs-based-group-decision-making/68225/

Unique Applications of Multi-Agent Models in Uncovering Language Learning Processes
www.irma-international.org/chapter/unique-applications-multi-agent-models/5137/

Portfolio Optimization Using Evolutionary Algorithms
www.irma-international.org/chapter/portfolio-optimization-using-evolutionary-algorithms/28381/

The Context-Security Nexus in Ubiquitous Computing
www.irma-international.org/article/the-context-security-nexus-in-ubiquitous-computing/118298/