Information Sciences 192 (2012) 82–97

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

The differential ant-stigmergy algorithm

Peter Korošec [a,b,*], Jurij Šilc [a,c], Bogdan Filipič [d]

[a] Computer Systems Department, Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
[b] Faculty of Mathematics, Natural Sciences and Information Technologies Koper, University of Primorska, Glagoljaška ulica 8, SI-6000 Koper, Slovenia
[c] Jožef Stefan International Postgraduate School, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
[d] Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 15 August 2009
Received in revised form 26 March 2010
Accepted 3 May 2010
Available online 6 May 2010

Keywords:
Global optimization
High-dimensional problems
Swarm intelligence
Stigmergy
Ant-Colony Optimization

ABSTRACT

Ant-Colony Optimization (ACO) is a popular swarm intelligence scheme known for its efficiency in solving combinatorial optimization problems. However, despite some extensions of this approach to continuous optimization, high-dimensional problems remain a challenge for ACO. This paper presents an ACO-based algorithm for numerical optimization capable of solving high-dimensional real-parameter optimization problems. The algorithm, called the Differential Ant-Stigmergy Algorithm (DASA), transforms a real-parameter optimization problem into a graph-search problem. The parameters' differences assigned to the graph vertices are used to navigate through the search space. We compare the algorithm results with the results of previous studies on recent benchmark functions and show that the DASA is a competitive continuous optimization algorithm that solves high-dimensional problems effectively and efficiently.

1. Introduction

Within the last two decades, various kinds of optimization algorithms have been designed and applied to multi-parameter function-optimization problems. Some of the popular approaches are Genetic Algorithms (GAs) [40], Evolutionary Algorithms (EAs) [7,42], Differential Evolution (DE) [35], Particle Swarm Optimization (PSO) [1], Artificial Immune Systems (AISs) [16], and most recently, ant-colony-based algorithms.

Ants have always fascinated humans. What is particularly striking to both the occasional observer and the scientist is the high degree of societal organization that these insects can achieve, in spite of very limited individual capabilities [10]. Such ants have also inspired a number of optimization algorithms, and these algorithms are becoming increasingly popular among researchers in computer science and operational research [3,6,12]. A particularly successful metaheuristic, Ant-Colony Optimization (ACO), as a common framework for the existing variants of ant-based algorithms and their applications, was proposed in the early 1990s by Dorigo [9]. ACO takes inspiration from the foraging behavior of certain ant species. These ants deposit pheromone on the ground to mark a favorable path that should be followed by the other colony members. ACO exploits a similar mechanism in solving combinatorial optimization problems where it has been proven to be one of the most successful metaheuristics.

However, a direct application of ACO to continuous optimization problems is difficult, since in this case the pheromone-laying method is not straightforward. The first ACO algorithm designed for continuous function-optimization was Continuous Ant-Colony Optimization (CACO) [2], which comprises two levels: global and local. It uses the ant-colony framework to

* Corresponding author at: Computer Systems Department, Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia.
E-mail addresses: peter.korosec@ijs.si (P. Korošec), jurij.silc@ijs.si (J. Šilc), bogdan.filipic@ijs.si (B. Filipič).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2010.05.002

perform the local search, whereas the global search is handled by a genetic algorithm. So far, there have been a few other adaptations of the ACO algorithm to continuous optimization problems:

- The simplified direct simulation of real ants' behavior: CACO with discrete encoding [19], API (named after *Pachycondyla apicalis*) [30], and Continuous Interacting Ant Colony (CIAC) [11].
- The extended ACO metaheuristic to explore continuous space with discretization: adaptive ant-colony algorithm [27], binary ant system [21], and multilevel ant-stigmergy algorithm [24].
- The extended ACO metaheuristic to explore continuous space with probabilistic sampling: extended ACO (ACO_{IR}) [34], continuous ant-colony system [31], aggregation pheromone system [38], direct ACO [22], Multivariate Ant-Colony Algorithm for Continuous Optimization (MACACO) [13], and differential ant-stigmergy algorithm [23].
- *The hybrid methods*: ACO_{IR} with Levenberg–Marquard algorithm [33], ACO with Powell search method [15], ACO with genetic algorithm [5], ACO with tabu-search method [20], orthogonal scheme ACO [18], and CACO with immune algorithm [14].

Although these algorithms have shown promising search abilities when applied to low-dimensional problems, many of them suffer from the "curse of dimensionality" which implies that their performance deteriorates quickly as the dimension of the search space increases.

To overcome this weakness, we propose an ant-based algorithm, called the Differential Ant-Stigmergy Algorithm (DASA), designed to successfully cope with high-dimensional numerical optimization problems. The rationale behind the algorithm is in remembering the move in the search space that improves the current best solution, and directing further search based on this move. For this purpose, we create a fine-grained discrete form of the continuous domain that represents the search space as a graph. This graph is then used as a means for ants to walk on.

Using a discrete form to solve continuous optimization problems is not new, but the way we use the discretized values is. The usual approach consists of determining a parameter range and discretizing it to a desired accuracy. This makes it possible to solve continuous problems with an algorithm primarily designed for combinatorial optimization. However, when the number of variables becomes high, the problem search space increases dramatically which disables the original algorithm to perform to its full potential [24]. We enhance the usual approach by introducing a new concept of solving continuous optimization problems using variable offsets. Instead of dealing with real discretized values, we deal with discretized offsets. This technique enables us to efficiently solve continuous optimization problems with accuracy limited only by the precision of presenting numbers in a computer.

This paper is further organized in the following way. In Section 2 we introduce the proposed optimization algorithm DASA. In Section 3 we present its experimental evaluation on high-dimensional benchmark functions, followed by the comparison of the algorithm with some recent population-based optimization algorithms. The experimental results are discussed in Section 4. The paper ends with the concluding remarks and directions of future work in Section 5.

2. Differential Ant-Stigmergy Algorithm (DASA)

Real-parameter optimization problem is to find a vector of parameter values, $\vec{x} = (x_1, x_2, \ldots, x_D)$ that minimizes a function, $f(\vec{x})$, of D real variables, i.e.,

$$\text{find}: \ \vec{x}^* | f(\vec{x}^*) \leqslant f(\vec{x}), \quad \forall \vec{x} \in \mathbb{R}^D.$$

To solve this problem, we create a fine-grained discrete form of continuous domain. Using this form, we are able to represent the problem as a graph. This enables us to apply the ant-based approach for solving numerical optimization problems.

2.1. The fine-grained discrete form of continuous domain

Let x'_i be the current value of the ith parameter. During the search for the optimal parameter value, a new value, x_i, is assigned to the ith parameter as follows:

$$x_i = x'_i + \delta_i. \tag{1}$$

Here, δ_i is called the *parameter difference* and is chosen from the set

$$\Delta_i = \Delta_i^- \cup \{0\} \cup \Delta_i^+,$$

where

$$\Delta_i^- = \{\delta_{i,k}^- | \delta_{i,k}^- = -b^{k+L_i-1}, \ k = 1, 2, \ldots, d_i\}$$

and

$$\Delta_i^+ = \{\delta_{i,k}^+ | \delta_{i,k}^+ = b^{k+L_i-1}, \ k = 1, 2, \ldots, d_i\}.$$