Expert Systems with Applications 39 (2012) 706–712

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Two-stage updating pheromone for invariant ant colony optimization algorithm

Zhaojun Zhang *, Zuren Feng

State Key Laboratory for Manufacturing Systems Engineering, Systems Engineering Institute, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China

ARTICLE INFO

Keywords:
Ant colony optimization
Invariance
Two-stage
Solution order
Quality function

ABSTRACT

Ant colony optimization (ACO) is a metaheuristic approach for combinatorial optimization problems. With the introduction of hypercube framework, invariance property of ACO algorithms draws more attention. In this paper, we propose a novel two-stage updating pheromone for invariant ant colony optimization (TSIACO) algorithm. Compared with standard ACO algorithms, TSIACO algorithm uses solution order other than solution itself as independent variable for quality function. In addition, the pheromone trail is updated with two stages: in one stage, the first r iterative optimal solutions are employed to enhance search capability, and in another stage, only optimal solution is used to accelerate the speed of convergence. And besides, the pheromone value is limited to an interval. We prove that TSIACO not only has the property of linear transformational invariance but also has translational invariance. We also prove that the pheromone trail can limit to the interval $(0, 1]$. Computational results on the traveling salesman problem show the effectiveness of TSIACO algorithm.

1. Introduction

Ant colony optimization (ACO) is a cooperative algorithm inspired by the foraging behavior of real ant colonies. The first ACO algorithm, called ant system (AS), was initially proposed in Dorigo, Maniezzo, and Colorni (1996) and applied to the solution of the TSP, but it was not able to compete against the state-of-the art algorithms in the field. In the next years, many ACO algorithms have been developed to improve the performance of AS. The main ACO algorithms presented in the literatures are: Ant-Q system (Gambardella & Dorigo, 1995), ant colony system (ACS) (Dorigo & Gambardella, 1997), max–min ant system (MMAS) (Stützle & Hoos, 2000), rank-based ant system (AS_{rank}) (Bullnheimer, Hartl, & Strauss, 1999).

After the initial proof-of-concept application to the traveling salesman problem (TSP) (Ellabib, Calamai, & Basir, 2007; Favaretto, Moretti, & Pellegrini, 2006; Wu, Zhao, Ren, & Quan, 2009), ACO was applied to many other combinatorial optimization problems. Examples are the applications to quadratic assignment problems (QAP) (Gambardella, Taillard, & Dorigo, 1999), job-shop scheduling problems (Heinonen & Pettersson, 2007; Udomsakdigool & Kachitvichyanukul, 2008), vehicle routing problems (Bell & McMullen, 2004; Montemanni, Gambardella, Rizzoli, & Donati, 2005), and set covering problems (Lessing, Dumitrescu, & Stützle, 2004). With many more novel implementations of ant algorithms being developed, continuous optimization is an example of broad the application area for ACO (Socha & Dorigo, 2008). In comparison with the application on ACO algorithms in recent years, the theoretical foundation of this randomized search heuristic is rather weak. Up to now, convergence results (Gutjahr, 2000; Stützle & Dorigo, 2002) have been achieved showing that optimal solutions can be obtained in finite time with probability one. Recently, Neumann and Witt (2009) presented runtime analysis of an ACO algorithm called the 1-ANT with specific lower and upper pheromone bounds on the pseudo-Boolean function and One-Max function. Runtime analysis of ACO algorithm is attracting more attention (Gutjahr, 2008; Gutjahr & Sebastiani, 2008; Zhou, 2009).

Invariance property of ACO algorithms has also been studied by a few authors. This property is definitely desirable for at least two main reasons: first, it reduces possible numerical problems in the implementations and contributes therefore to enhance the stability of the algorithm; second, it greatly improves the readability of the solution process (Birattari, Pellegrini, & Dorigo, 2007). As ACO algorithms using the pheromone model belong to the class of model-based search (MBS) algorithms (Zlochin, Birattari, Meuleau, & Dorigo, 2004). The model does not change but the value of pheromone trail update by using the previously generated solutions to generate high-quality solutions increases over time at runtime. It is undesirable property that the pheromone values strongly depend on the scale of the problem. Blum and Dorigo introduced the hypercube framework (HCF) and were first to raise the issue on the invariance of ACO algorithms to transformation of units (Blum & Dorigo, 2004). In order to achieve the strong invariant property, the quality function is employed to normalize the added amount of pheromone in HCF. Birattari shown the

* Corresponding author. Tel.: +86 29 82667771; fax: +86 29 82665487.
E-mail address: zzj921@gmail.com (Z. Zhang).

0957-4174/$ - see front matter
doi:10.1016/j.eswa.2011.07.062

performance of standard ACO algorithms is independent of the scale of the problem under some conditions (Birattari et al., 2007). That is, the sequence of solutions of ACO algorithms finding is independent of the scale of the problem. The strongly invariant ACO algorithms were also introduced to compare with hypercube framework.

In this paper, we introduce two-stage updating pheromone for invariant ant colony optimization (TSIACO) algorithm. In TSIACO algorithm, ordinal update rule is introduced, the pheromone update by two stages and the range of values the pheromone trail is limited to the interval. Compared with standard ACO algorithms, the quality function use the solution order as its independent variable in one iteration. The benefit of using order is that the pheromone values can independent to objective function values, so the ACO algorithms equipped with it have the invariance naturally. The bound and added pheromone trail is independent to the function values. To make this novel work more useful in practice, two specific algorithms are given in following section.

The rest of this paper is organized as follows. In Section 2, we review the AS based on construction graph and introduce some preliminary concepts used in following section. We propose two stages invariance ant colony algorithm and prove its invariance property in Section 3. In Section 4 presents some computational results which take the example of TSP. In Section 5, we conclude this paper and outline future work.

2. Ant system and preliminary definitions

Ant system (AS) and a number of fundamental concepts that will be needed in the following sections are introduced in this section.

2.1. Ant system

Consider a minimization problem (S, f, Ω), where S is the set of (candidate) solutions, f is the objective function, which assigns to each candidate solution $s \in S$ an objective function value (cost) $f(s)$, and Ω is a set of constraints, which defines the set of feasible candidate solutions. The goal of the minimization problem is to find an optimal solution s^*, i.e., a feasible candidate solution of minimum cost.

Definition 1 (*Construction graph*). In ACO, a combinatorial optimization problem is mapped on a graph $G = (\mathcal{N}, \mathcal{A}, \mathcal{T}, \mathcal{H})$, where $\mathcal{N}$ is the set of nodes, $\mathcal{A}$ is the set of arcs and variables $\mathcal{T}$, $\mathcal{H}$ are vector gathering called pheromone trail and heuristic information which are associated with the arcs in $\mathcal{A}$. The graph G is called construction graph.

The solutions of minimization problem are mapped to paths on G. Artificial ants build candidate solutions by performing randomized walks on the completely construction graph G according to the strength of the pheromone trail and heuristic information currently on the arcs. While moving from one vertex to another, constraints Ω are used to prevent ants from building infeasible solutions. Once the ants have completed their walk, pheromone trails are updated. A Procedure of a generic ACO algorithm is given in Table 1 (Dorigo & Stützle, 2004).

Table 1
The procedure ACO algorithm.

Procedure ACO algorithm
Set parameters, initialize pheromone trails
While (termination condition not met) do
Construct ants solutions
Update pheromone trail
End
End

2.1.1. Initialize pheromone values

At the beginning of the algorithm all the pheromone values are initialized to the numerical value $c > 0$. This value is small as in most ACO algorithms.

2.1.2. Construct solution

Initially m ants are put on randomly chosen node. Each ant applies a probabilistic action choice rule, called random proportional rule, to decide which node to visit next, at each construction step. Suppose ant k in node i in iteration t, then it will choose the next node j with a probability

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} & \text{if } j \in N_i^k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where η_{ij} is a heuristic value, $\tau_{ij}(t)$ is pheromone trail value, α and β are two parameters which determine the relative influence of the pheromone trail and the heuristic information, and N_i^k is the feasible neighborhood of ant k when being at node i. We call this rule as random proportional rule of ACO algorithm.

2.1.3. Apply pheromone update

Once all ants have completed a tour, the pheromone trails are updated. The update follows this rule

$$\tau_{ij}(t+1) = [(1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k(t)] \quad (2)$$

where $\rho (0 < \rho \leqslant 1)$ is the evaporation rate, $\Delta\tau_{ij}^k$ is the quantity of pheromone ant k laid on the visited path and is given by

$$\Delta\tau_{ij}^k(t) = \begin{cases} F_1(s^k(t)), & \text{if } (i,j) \in s^k(t) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $s^k(t)$ is solution generated by ant k, and $F_1 : S \mapsto \mathbb{R}^+$, called quality function, is a function such that $f(s_1) < f(s_2) \Rightarrow F_1(s_1) > F_1(s_2)$, for $\forall s_1, s_2 \in S$, and $s_1 \neq s_2$. Generally, let $F_1(s^k(t)) = 1/f(s^k(t))$.

Up to now, most improvement work is concentrated to pheromone update. They use different devices of utilizing the pheromone to overcome the drawbacks of AS and achieve much better performance. Recent studies have uncovered that considering the solution bias in pheromone model does lead to the best performance of in ACO algorithms (Blum & Dorigo, 2005; Montgomery, Randall, & Hendtlass, 2008).

As the function quality use the solution for its independent variable, it is difficult to keep invariant property for the standard ACO algorithms described in the previous section. In order to get invariant property, Blum and Dorigo have introduced hyper cube framework for pheromone trail update rule defined as

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho \sum_{k=1}^{m} \frac{F_1(s^k(t))}{\sum_{h=1}^{m} F_1(s^h(t))} \quad (4)$$

The main contribution of hyper cube framework is that the pheromone trail is dimensionless via using normalization of quality function. The similar method is employed by Birattari to prove that the standard ACO algorithms are indeed weak invariant if some conditions are fulfilled. But it does not fundamentally eliminate the effect of solution uncertain.