Applied Soft Computing 28 (2015) 150–159

Contents lists available at ScienceDirect

Applied Soft Computing

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

# A new approach for dynamic fuzzy logic parameter tuning in Ant Colony Optimization and its application in fuzzy control of a mobile robot

Oscar Castillo*, Héctor Neyoy, José Soria, Patricia Melin, Fevrier Valdez

*Tijuana Institute of Technology, Tijuana, Mexico*

ARTICLE INFO

*Article history:*
Received 18 October 2012
Received in revised form
29 November 2014
Accepted 1 December 2014
Available online 10 December 2014

*Keywords:*
ACO
Fuzzy logic
Fuzzy control
Robotics
Fuzzy ACO

ABSTRACT

Ant Colony Optimization is a population-based meta-heuristic that exploits a form of past performance memory that is inspired by the foraging behavior of real ants. The behavior of the Ant Colony Optimization algorithm is highly dependent on the values defined for its parameters. Adaptation and parameter control are recurring themes in the field of bio-inspired optimization algorithms. The present paper explores a new fuzzy approach for diversity control in Ant Colony Optimization. The main idea is to avoid or slow down full convergence through the dynamic variation of a particular parameter. The performance of different variants of the Ant Colony Optimization algorithm is analyzed to choose one as the basis to the proposed approach. A convergence fuzzy logic controller with the objective of maintaining diversity at some level to avoid premature convergence is created. Encouraging results on several traveling salesman problem instances and its application to the design of fuzzy controllers, in particular the optimization of membership functions for a unicycle mobile robot trajectory control are presented with the proposed method.



## 1. Introduction

Ant Colony Optimization (ACO) is inspired by the foraging behavior of ant colonies, and is aimed at solving discrete optimization problems [8].

The behavior of the ACO algorithm is highly dependent on the values defined for its parameters as these have an effect on its convergence. Usually these are kept static during the execution of the algorithm. Changing the parameters at runtime, at a given time or depending on the search progress may improve the performance of the algorithm [25–27].

Controlling the dynamics of convergence to maintain a balance between exploration and exploitation is critical for good performance in ACO. Early convergence leaves large sections of the search space unexplored. Slow convergence does not concentrate its attention on areas where good solutions are found.

Fuzzy control has emerged as one of the most active and fruitful areas of research in the application of fuzzy sets and fuzzy logic. The methodology of fuzzy logic controllers is useful when processes are too complex for analysis by conventional quantitative techniques or when the available sources of information are interpreted in a qualitatively inaccurate or uncertain way [40].

Determining the correct parameters for the fuzzy logic controller is a complex problem and it is also a task that consumes considerable time. Because of their ability to solve complex NP hard problems we made use of ACO for the selection of those already mentioned parameters.

There is also some recent interest in using ACO algorithms in mobile robotics [5,28]. Nowadays robotic automation is an essential part in the manufacturing process. Autonomous navigation of mobile robots is a challenge. A mobile robot can be useful in unattainable goal situations due to geological conditions or where the human are being is endangered. So, mobile robotics is an interesting subject for science and engineering.

This paper explores a new method of diversity control in ACO. The main idea is to prevent or stop the total convergence through the dynamic adjustment of certain parameter of the algorithm applied to the design of fuzzy controllers, specifically to the optimization of membership functions of a trajectory controller for a unicycle mobile robot.

The rest of the paper is organized as follows. Section 2 presents an overview of ACO. Section 3 describes a performance analysis on several TSP instances. Section 4 presents a new method of parameter tuning using fuzzy logic, Section 5 shows some simulation

* Corresponding author. Tel.: +52 6646236318.
*E-mail address:* ocastillo@tectijuana.mx (O. Castillo).

http://dx.doi.org/10.1016/j.asoc.2014.12.002
1568-4946/

results in TSP problems, Section 6 describes the optimized fuzzy controller, Section 7 presents the considerations that are used to implement the ACO algorithm in the optimization of membership functions, Section 8 describes how the proposed method is applied, Sections 9 and 10 show simulation results in the membership functions optimization problem, and finally Section 11 presents some conclusions.

## 2. Ant Colony Optimization

The first ACO algorithm was called Ant System (AS) and its main objective was to solve the traveling salesman problem (TSP), whose goal is to find the shortest route to link a number of cities. In each iteration each ant keeps adding components to build a complete solution, the next component to be added is chosen with respect to a probability that depends on two factors. The pheromone factor that reflects the past experience of the colony and the heuristic factor that evaluates the interest of selecting a component with respect to an objective function. Both factors weighted by the parameters $\alpha$ and $\beta$ respectively define the probability $P$ in (1)

$$P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}}, \quad \text{if } j \in N_i^k \tag{1}$$

In (1) $\tau_{ij}$ represents the pheromone value between nodes $i$ and $j$ and $\eta_{il}$ represents the heuristic factor that evaluates the interest of selecting a component with respect to an objective function. Finally, $N_i$ represents a neighborhood of node $i$.

After all ants have built their tours, the pheromone trails are updated. This is done by decreasing the pheromone value on all arcs by a constant factor (2), which prevents the unlimited accumulation of pheromone trails and allows the algorithm to forget bad decisions previously taken.

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij}, \quad \forall (i,j) \in L \tag{2}$$

And by depositing pheromone on the arcs that ants have crossed in its path (3). The better the tour, the greater the amount of pheromone that the arcs will receive. In (2) $\rho$ represents the rate of pheromone evaporation, which is a value between 0 and 1.

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{n}\Delta\tau_{ij}^{k}, \quad \forall (i,j) \in L$$
$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{1}{C^k}, & \text{if arc } (i,j) \text{ belong to } T^k; \\ 0, & \text{otherwise;} \end{cases} \tag{3}$$

In (3) $C$ represents the cost of an arc in a graph. A first improvement on the initial AS, called the elitist strategy for Ant System (EAS) is as follows. The idea is to provide strong additional reinforcement to the arcs belonging to the best tour found since the start of the algorithm (4) [8].

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{n}\Delta\tau_{ij}^{k} + e\Delta\tau_{ij}^{bs}, \quad \forall (i,j) \in L$$
$$\Delta\tau_{ij}^{bs} = \begin{cases} \dfrac{1}{C^{bs}}, & \text{if arc } (i,j) \text{ belong to } T^{bs}; \\ 0, & \text{otherwise;} \end{cases} \tag{4}$$

In (4) the term $\Delta\tau$ represents the pheromone increment and the *bs* indication is to distinguish the best-so-far ant. Another improvement over AS is the rank-based version of AS (denoted $AS_{Rank}$). In $AS_{rank}$ each ant deposits an amount of pheromone that decreases with its rank. Additionally, as in EAS, the best-so-far ant always deposits the largest amount of pheromone in each iteration [8]. In (5) $w$ represents a number of ants considered in the ranking and $r$ is an index for the ants in this set of $w$ ants.

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{r=1}^{w-1}(w-r)\Delta\tau_{ij}^{r} + \Delta\tau_{ij}^{bs} \tag{5}$$

**Table 1**
TSP instances considered.

| TSP | Number of cities | Best tour length |
|---|---|---|
| Burma14 | 14 | 3323 |
| Ulysses22 | 22 | 7013 |
| Berlin52 | 52 | 7542 |
| Eil76 | 76 | 538 |
| kroA100 | 100 | 21,282 |

**Table 2**
Parameters used for each ACO variant.

| ACO | $\alpha$ | $\beta$ | $\rho$ | $m$ | $\tau_0$ |
|---|---|---|---|---|---|
| AS | 1 | 2 | 0.5 | $n$ | $m/C^{nn}$ |
| $AS_{Rank}$ | 1 | 2 | 0.1 | $n$ | $0.5r(r-1)/\rho C^{nn}$ |
| EAS | 1 | 2 | 0.5 | $n$ | $(e+m)/\rho C^{nn}$ |

$m = n$.
$C^{nn} = 20$ for each tsp except burma14 where $C^{nn} = 10$.
EAS: $e = 6$.
$AS_{Rank}$: $r = w - 1$; $w = 6$.

**Table 3**
Performance obtained for the TSP instance Burma14.

| ACO | Best | Average | Successful runs |
|---|---|---|---|
| AS | 3323 | 3323 | 30/30 |
| $AS_{Rank}$ | 3323 | 3329 | 19/30 |
| EAS | 3323 | 3323 | 30/30 |

**Table 4**
Performance obtained for the Ulysses22 TSP instance.

| ACO | Best | Average | Successful runs |
|---|---|---|---|
| AS | 7013 | 7022 | 30/30 |
| $AS_{Rank}$ | 7013 | 7067 | 19/30 |
| EAS | 7013 | 7018 | 30/30 |

**Table 5**
Performance obtained for the Berlin52 TSP instance.

| ACO | Best | Average | Successful runs |
|---|---|---|---|
| AS | 7542 | 7557 | 2/30 |
| $AS_{Rank}$ | 7542 | 7580 | 17/30 |
| EAS | 7542 | 7554 | 6/30 |

## 3. Performance analysis of ACO

To analyze the performance of the AS, EAS and $AS_{Rank}$ variants, 30 experiments were performed by method for each instance of the examined TSP (Table 1), which are in the range of 14–100 cities, all extracted from TSPLIB [33], using the parameters recommended by the literature (Table 2) [8].

The behavior of AS and EAS is very similar in all experiments (Tables 3–7), the performance of the three variants began to worsen

**Table 6**
Performance obtained for the Eil76 TSP instance.

| ACO | Best | Average | Successful runs |
|---|---|---|---|
| AS | 547 | 556 | 0/30 |
| $AS_{Rank}$ | 538 | 543 | 1/30 |
| EAS | 544 | 555 | 0/30 |