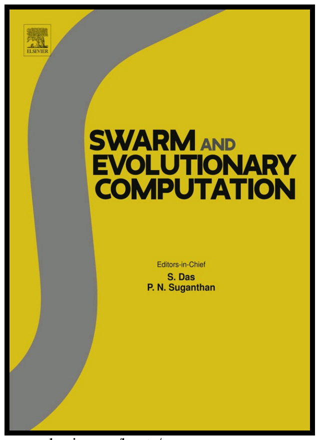
Effective heuristics for ant colony optimization to handle large-scale Problems

Hassan Ismkhan



www.elsevier.com/locate/swevo

PII: S2210-6502(16)30058-X
DOI: http://dx.doi.org/10.1016/j.swevo.2016.06.006
Reference: SWEVO223

To appear in: *Swarm and Evolutionary Computation*

Received date: 23 May 2015
Revised date: 24 May 2016
Accepted date: 20 June 2016

Cite this article as: Hassan Ismkhan, Effective heuristics for ant colon optimization to handle large-scale Problems, *Swarm and Evolutionar Computation,* http://dx.doi.org/10.1016/j.swevo.2016.06.006

Effective Heuristics for Ant Colony Optimization to Handle Large-Scale Problems[1]

Hassan Ismkhan
Faculty member at University of Bonab, East Azerbaijan, Iran
Fax: +98-4137740800, Email H.Ismkhan@bonabu.ac.ir (or Esmkhan@gmail.com)

Abstract
Although ant colony optimization (ACO) has successfully been applied to a wide range of optimization problems, its high time- and space-complexity prevent it to be applied to the large-scale instances. Furthermore, local search, used in ACO to increase its performance, is applied without using heuristic information stored in pheromone values. To overcome these problems, this paper proposes new strategies including effective representation and heuristics, which speed up ACO and enable it to be applied to large-scale instances. Results show that in performed experiments, proposed ACO has better performance than other versions in terms of accuracy and speed.

Keyword: Large-scale optimization; Ant Colony Optimization; ACO; Heuristics; Traveling Salesman Problem.

1. Introduction

ACO algorithms have successfully been applied to many optimization problems. These problems include combinatorial optimizations such as versions of scheduling problems , finding edge-disjoint paths problem , types of quadratic assignment problem , QoS-based service selection problems, types of vehicle routing problem , types of traveling salesman problem and knapsack problem. The ACO has also applications in mobile networks, community mining in social networks , solving clustering problems, finger print matching and many other problems. ACO algorithms also have successfully been applied to continuous optimization problems. A previous work proposes a type of ACO that is applied to continues function optimization by archiving solutions and utilizing a type of local search algorithm in its baseline algorithm. Paper suggests a niche ACO-based on the fitness sharing principle to solve continuous optimization problems. Continuous optimization problem have been solved by types of ACO in two other works .

Ant algorithms are easy to implement and cover wide range of applications , but their performance dramatically decreases in dealing with large-scale problems. The proposed algorithm in is a hybrid method, which combines ACO with swarm algorithm and a local search. The largest instance in the experiments of this paper is Kroa200 with size of 200 (as number of nodes). Reference proposes a version of ACO based on memetic algorithm. The size of largest instance used in the experiments of this paper is also 200. The largest instance used in , which is based on combing ACO with a type of gradient search, is also KroA200. The sizes of the largest instances in , which propose versions of ACO, is 250 or even less. The word "large-scale" is in title of , however, similar to , it has not been applied to instances with size of larger than 1400. The size of the largest instance in , which proposes an algorithm based on combination of ACO with Artificial Bee Colony (ABC) algorithm, is 724. The parallel algorithm proposed in tries to overcome the premature convergence, which is one of the major problems of ACO and leads to trap in local optimum. The largest instance considered in this paper is pr2392 with size of 2392. Where in almost all references in this paper, the versions of ACO have been applied to instances with the maximum size of less than 2300. The proposed parallel strategy in has been applied to Pcb3038, which is a TSP instance with size of 3038.

Some drawbacks of current versions of ACO are as follows: 1) Space complexity of holding pheromone in main memory is high and the most famous current solution to solve this problem is using candidate pheromone values which is too inflexible in many situations. 2) Current versions of ACO use the local search algorithms in their baseline algorithm, but they are not able to use pheromone information to operate more efficiently. 3) The time complexity of selecting next move is high. As the case 1, the current available solution to solve this problem is using candidates-sets, which reduces performance of ACO. This paper, states these drawbacks with more details. This paper also suggests some new and effective strategies to eliminate these drawbacks. These strategies increase performance of ACO and enable it to apply to large-scale problems.

Therefore, the rest of this paper has been organized as follows: This paper reviews ACO in section 2 and it points out some ACO drawbacks in section 3. Section 4 proposes and designs some new strategies to eliminate current

[1] This work is supported by University of Bonab (contract number : 94/D/AP/543)