The Effectiveness of Dynamic Ant Colony Tuning†

Adrian A. de Freitas and Christopher B. Mayer
Air Force Institute of Technology
Department of Electrical and Computer Engineering
2950 Hobson Way
Wright-Patterson AFB, OH 45431
Adrian.deFreitas@afit.edu, Chris.Mayer@afit.edu

ABSTRACT
We examine the Genetically Modified Ant Colony System (GMACS) algorithm [3], which claims to dynamically tune an Ant Colony Optimization (ACO) algorithm to its near-optimal parameters. While our research indicates that the use of GMACS does result in higher quality solutions over a hand-tuned ACO algorithm, we found that the algorithm is ultimately hindered by its emphasis on randomized ant breeding. Specifically, our investigation shows that tuning ACO parameters on a single colony using a genetic algorithm, as done by GMACS, is not as effective as it may first appear and has several drawbacks.

Categories and Subject Descriptors
I.2.6 [**Artificial Intelligence**]: Learning – *parameter learning.*

General Terms
Algorithms, Performance, Design, Experimentation, Theory.

Keywords
Ant Colony Optimization, Genetic Algorithms, Genetically Modified Ant Colony System, Traveling Salesman Problem.

1. BACKGROUND AND PURPOSE
Ant Colony Optimization (ACO) algorithms can generate quality solutions to many NP-hard problems [1]. Unfortunately, the performance of ACO algorithms depends on their parameter settings. The Genetically Modified Ant Colony System (GMACS) algorithm proposed in [3] attempts to replace the current practice of hand-tuning ACO parameters through the use of genetic algorithms. Under GMACS, a single ant colony is used. Each ant in the colony has its own set of parameters and each ant represents a chromosome. To begin solving, an initial ant population with randomly assigned parameters is generated. After each ant in the colony has solved, ants with favorable solution results are identified and randomly bred using traditional genetic programming techniques such as cross-over and mutation to produce a new population. Repeated rounds of solving and breeding are intended to produce both good solutions and identify highly preferable parameters for future use by the ACO algorithm (even if GMACS is not actively tuning).

We have implemented the GMACS algorithm for the Traveling Salesman Problem (TSP) and have run it on a number of standard problems in the TSPLIB [4]. Our investigation of GMAC's performance, detailed in the next section, raises serious doubts about the practicality of simultaneous tuning and solving problems using an approach like GMACS.

GECCO'07, July 7–11, 2007, London, England, United Kingdom
ACM 978-1-59593-697-4/07/0007.

2. INVESTIGATION AND ANALYSIS
Breeding and Solution Quality. In comparison to a hand-tuned version of Ant Colony System (ACS) [2], GMACS can show as much as a 3.25% increase in solution quality. This seems to indicate that GMACS' breeding process really works. However, we compared the performance of GMACS against a version of ACS in which random parameters are assigned to every ant at the start of each time step. This modified ACS algorithm produced comparable results to GMACS and in 75% of cases it actually generated higher quality solutions. This suggests that GMACS is no better than random guessing.

Parameter Convergence. Although the use of GMACS does cause an ant population to eventually converge on a set of parameters, we were unable to produce a single case in which an ant with these parameters found the best solution. Instead, the best solution in each experiment was produced by an ant that was not a member of the final population (sometimes they were not even close in the parameter space). We suspect that this behavior stems from GMACS' reliance on randomized breeding. Since a single good ant is just a small portion of a much larger breeding population, GMACS is apt to lose the ant's traits due to the randomized breeding process. This, in turn, causes the algorithm to converge on suboptimal parameters.

Pheromone Update. Because the early generations of GMACS' ants are randomly initialized, it is likely that these populations are of relatively low quality. These ants, however, are able to create pheromone trails that will sway the decisions of future generations. By preventing pheromone tables for the first several time steps (e.g., ten), we were able to improve solution quality.

3. REFERENCES
[1] E. Bonabeau, M. Dorigo, and G. Theraulaz (1999). *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, New York.

[2] M. Dorigo and L.M. Gambardella. *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*. IEEE Transactions on Evolutionary Computation, 1(1):53-66, 1997.

[3] D. Gaertner and K. L. Clark. *On Optimal Parameters for Ant Colony Optimization Algorithms*. IC-AI 2005: 83-89.

[4] TSP-Library (2001). http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

† The views expressed herein are those of the authors and do not reflect the official policy or position of the U.S. Air Force, Dept. of Defense, or the U.S. Government.