Near Parameter Free Ant Colony Optimisation

Marcus Randall

Meta-heuristic Search Group
Bond University, QLD 4229, Australia
Phone: +61 7 55953361
mrandall@bond.edu.au

Abstract. Ant colony optimisation, like all other meta-heuristic search processes, requires a set of parameters in order to solve combinatorial problems. These parameters are often tuned by hand by the researcher to a set that seems to work well for the problem under study or a standard set from the literature. However, it is possible to integrate a parameter search process within the running of the meta-heuristic without incurring an undue computational overhead. In this paper, ant colony optimisation is used to evolve suitable parameter values (using its own optimisation processes) while it is solving combinatorial problems. The results reveal for the travelling salesman and quadratic assignment problems that the use of the augmented solver generally performs well against one that uses a standard set of parameter values. This is attributed to the fact that parameter values suitable for the particular problem instance can be automatically derived and varied throughout the search process.

1 Introduction

Meta-heuristic search strategies, including tabu search, simulated annealing, GRASP and ant colony optimisation (ACO), invariably require a set of parameters in order to solve combinatorial optimisation problems. These parameters directly impact on the performance of the solver and as such, researchers and practitioners will often "hand tune" parameter values before the application of the production meta-heuristic or use a set of values that have been found to be traditionally "good" by other researchers (i.e., a standard set).

Relatively little research has been conducted into either the analysis of parameter values or the ways in which they can be automatically derived or tuned by meta-heuristics themselves. In this paper, ant colony optimisation is examined as it is an optimisation framework that has been successfully applied to a range of combinatorial optimisation problems [3, 7]. ACO represents a group of constructive meta-heuristics (often coupled with local search) that use collective intelligence present in insect colonies. The reader is referred to Dorigo and Gambardella [5] and Dorigo and Di Caro [3] for an overview and background of ACO. In this work, ant colony system (ACS) [5] is used as it is a robust and reliable technique.

In regards to ACO studies in which parameters have been analysed, Colorni, Dorigo and Maniezzo [2], Dorigo and Gambardella [4], Dorigo, Maniezzo

M. Dorigo et al. (Eds.): ANTS 2004, LNCS 3172, pp. 374–381, 2004.

and Colorni [6], Maniezzo and Colorni [9] and Shmygelska, Aguirre-Hernández and Hoos [13] have each compared and contrasted various parameter values on particular problems (most notably the travelling salesman problem (TSP) and quadratic assignment problem (QAP)) in order to derive suitable parameter sets.

In terms of automatic parameter adaptation, Ingber [8] developed an implementation known as Adaptive Simulated Annealing in which parameter values are changed in a systematic manner throughout the search process. Pilat and White [10] in their work used concepts from genetic algorithms in order to evolve solution parameters for a constituent ACO technique, ant colony system. This was developed as a "Meta ACS-TSP" that would run standard ACS within a genetic algorithm that evolves solution parameters (a computationally expensive exercise). Using this technique, they were able to suggest alternative good parameter values to Dorigo and Gambardella [5] for ACS and the TSP.

The approach adopted within uses standard mechanisms of ant colony system [5] to modify and to determine appropriate parameter values while problems are being solved. Therefore, it is conceptually simple to integrate this approach into an ant colony implementation (moreso than other search techniques, particularly iterative meta-heuristics). Another advantage is that ant based techniques *learn* appropriate values for particular problem instances (without the researcher/practitioner having to derive these manually). Additionally, it does not add a significant computational overhead to the native algorithm (unlike, for instance, that of Pilat and White [10]). The results show that good quality solutions are achieved for a range of TSP and QAP problem instances. The remainder of the paper is organised as follows. The extensions that allow ACS to evolve its own parameter values (using aspects of the native algorithm) are given in Section 2. Computational experiments, using benchmark TSP and QAP problem instances, are reported in Section 3. Finally, the future directions of this work and conclusions are outlined in Section 4.

2 Evolving ACO Parameter Values

ACS can use the same mechanics for generating solutions to evolve appropriate values for its parameters. Within ACS, the core parameters (apart from the number of ants) are as follows. These are introduced and described in greater detail by Dorigo and Gambardella [5].

- q_0: This parameter determines whether the greedy or probabilistic form of component selection equation is used by an ant at each step of the algorithm. A low value will more likely result in the use of the probabilistic form of the equation (and vice versa).
- ρ: The local pheromone updating factor.
- γ: The global pheromone updating factor.
- β: Is the relative importance placed on the visibility heuristic.

The standard ACS algorithm is augmented at each iteration by allowing each ant to select a value for each parameter before commencing the selection