Ant Systems for a Dynamic TSP
Ants Caught in a Traffic Jam

Casper Joost Eyckelhof\(^1\) and Marko Snoek\(^2\)

\(^1\) Multigator vof
Enschede, The Netherlands
casper@eyckelhof.nl

\(^2\) Department of Computer Science, University of Twente
P.O. Box 217, 7500 AE, Enschede, The Netherlands
snoek@cs.utwente.nl

Abstract. In this paper we present a new Ants System approach to a dynamic Travelling Salesman Problem. Here the travel times between the cities are subject to change. To handle this dynamism several ways of adapting the pheromone matrix both locally and globally are considered. We show that the strategy of smoothing pheromone values only in the area containing a change leads to improved results.

1 Introduction

In nature, many systems consisting of very simple parts demonstrate a remarkable complexity as a whole. An example of this is ant colonies: every single ant just seems to walk around independently, however the colony itself is organised very well. This effect is also known as emergent behaviour. Ants' foraging behaviour shows that travel between nests and sources of food is highly optimised. Based on this phenomenon, Ant Colony Optimisation (ACO) algorithms have been developed \[10,2,5\], of which Ant System (AS) was the first \[4\]. These ACO algorithms have been successfully applied to a variety of combinatorial optimisation problems \[1\].

Like other metaheuristics ACO algorithms have proven their use for static problems, but less is known about their behaviour on dynamic problems. In dynamic problems the goal is not to find desirable solutions, but to track and/or find new desirable solutions.

After the initial emphasis on static problems, some of the focus is now shifting towards dynamic variants of combinatorial optimisation problems. Recently some research is being done on ACO for dynamic problems \[379\].

This paper gives an outline of research done as part of a graduation project at the University of Twente \[6\]. The main goals of that research were to test the hypothesis that ant systems can be applied successfully to dynamic problems and to adapt the original ant system algorithm in order to increase its performance on a particular dynamic problem.

In this paper we will show that both goals were reached: ant systems can be applied successfully to dynamic problems and enhancements to the original...
algorithm are proposed for a chosen dynamic problem. The problem we study in this research is based on the Travelling Salesman Problem (TSP). TSP is a well-known combinatorial optimisation problem and the original ant system algorithm was designed for TSP.

The paper is structured as follows. In §2 the TSP as well as the ant system for this problem are explained. In §3 we introduce our adaptation to TSP in order to create a dynamic problem and we will explain how ant systems can be modified to perform better on this dynamic problem. Next the test setup for the experiments is explained in §4. In §5 the experimental results are presented. We will finish with our conclusions in §6 and ideas for future work in §7.

2 The Ant System Approach to TSP

The Travelling Salesman Problem is a well-known problem among computer scientists and mathematicians. The task basically consists of finding the shortest tour through a number of cities, visiting every city exactly once. Formally, the symmetric TSP is defined as:

Given a set of \( n \) nodes and costs associated with each pair of nodes, find a closed tour of minimal total costs that contains each node exactly once, the cost associated with the node pairs \( \{i,j\} \) and \( \{j,i\} \) being equal.

It is also possible to discard that last condition and allow the distance from city \( i \) to city \( j \) to be different from the distance between city \( j \) and city \( i \). We refer to that case as the asymmetric travelling salesman problem. Our focus will be on the symmetric TSP.

The popularity of TSP probably comes from the fact that it is a very easy problem to understand and visualise, while it is very hard to solve. Many other problems in the \( NP \)-Hard class are not only hard to solve, but also hard to understand. With TSP, having \( n \) cities, there are \( (n-1)! \) possible solutions for asymmetric TSP and \( (n-1)!/2 \) possible solutions for symmetric TSP. For small instances it is no problem to generate all solutions and pick the shortest, but because the number of possible solutions ‘explodes’ when the number of cities increases. Within this domain heuristics that find acceptable solutions using an acceptable amount of resources are necessary.

In their book [2], Bonabeau et al. give a good explanation of Ant System (AS), the algorithm we will use as a basis for our own algorithm. It has become common practice to name the algorithms with their field of application, so this version of AS is called AS-TSP.

In AS-TSP \( m \) ants individually construct candidate solutions in an incremental fashion. The choice of the next city is based on two main components: pheromone trails and a heuristic value, called visibility in TSP. At the start all possible roads are initialised with a certain amount of pheromone: \( \tau_0 \). Then each ant constructs a solution by choosing the next city based on the observed

\[ 1 \] Using nomenclature corresponding to the metaphor of a travelling salesman we will use city for node, distance or travel time for cost, and road for node pair.