An Investigation of an Ant-based Hyperheuristic for the Capacitated Vehicle Routing Problem

Zalilah Abd Aziz • Graham Kendall

1 Introduction

One of the aims of hyperheuristics is to develop more general systems that are able to solve a wider range of problems than is currently possible with more conventional search methodologies. Hyperheuristics can be defined as “heuristics to choose heuristics” (Burke et al. 2003), with their strength being in their ability to combine a number of simple heuristics to search for good solutions. These simple heuristics can be, for example, k-opt moves or simple local search moves that are problem dependent used by problem experts to construct solutions.

In this research, we intend to investigate ant-based hyperheuristic which are capable of managing a set of low-level heuristics utilising an ant algorithm (originally proposed by Dorigo et al. (1996)). A similar approach has previously been applied to two optimisation problems; the project presentation scheduling (Burke et al. 2005) and the travelling tournament problem (Chen et al. 2007). This research will extend these initial investigations and explore the effectiveness of the algorithm using the Capacitated Vehicle Routing Problem (CVRP).

2 Ant-based Hyperheuristics

In the proposed ant-based hyperheuristic, a problem is represented as a directed graph where the nodes represent low level heuristics. Initially, ants traverse the graph, carrying with them the initial solutions. To decide which heuristic (node) to visit next, the ants make the decision probabilistically.
The proposed ant-based hyperheuristic incorporates the following features:

1. **How ants make the decision as to which heuristic (node) to visit next?** The decision will be made based on the probability transition rule as defined in the Ant System (Dorigo et al. 1996), but suitably adjusted for a hyperheuristic framework.

2. **Pheromone value?** As the ants travel, they deposit a chemical substance called pheromone. In a standard Ant System, the amount of pheromone corresponds to the quality of the solution found by the ant. In this respect the hyperheuristic algorithm mirrors the original ant algorithm.

3. **Visibility?** Visibility information represents some form of heuristic information, which is combined with the pheromone value in order to decide which node (heuristic) to visit next. In the original ant algorithm, the heuristic information was the distance between two cities (as it was tackling the traveling salesman problem). As such, it remained static throughout the algorithm. In our proposed algorithm the heuristic information is how well two heuristics work together. This is measured as the computational time and, as such, it is now a dynamic value over the course of the algorithm.

3. **Problem Description : Capacitated Vehicle Routing Problem (CVRP)**

The Capacitated Vehicle Routing Problem (CVRP) is a version of the Vehicle Routing Problem (VRP) where a fleet of \( m \) vehicles with limited capacity \( Q \) has to visit a set of customers with specific demands. The sum of demands on any route should not exceed the vehicle capacity \( Q \). The objective is to minimize the total travel distance. The CVRP can be formulated as follows: a connected graph \( G = (V,E) \) of \( n+1 \) nodes, with a set of customers with demand \( q_i \) is represented as \( v_i \in V, i = 1\ldots n \) with \( v_0 \) as the depot.

The CVRP problem is to find a set of least cost vehicle routes so that the following constraints are satisfied:

1. Each vertex except the first vertex (depot), \( v_0 \), is served exactly once by exactly one vehicle.
2. Each vehicle’s route starts and ends at the depot.
3. The total demand does not exceed the vehicle capacity \( Q \) for every route.

4. **Low Level Heuristics**

Low level heuristics are held in the nodes of the graph. Usually, they represent simple local search neighborhoods such as swap, move or rules that were used by the user to construct the solutions. In the proposed hyperheuristic framework, low level heuristics will generate a new solution and we will maintain other information such as changes in the evaluation function and computational time taken to carry out the heuristic.

We have drawn from the literature (Braysy and Gendreau 2005), a set of low level heuristics for the CVRP. They are briefly described here:
1. **2-opt exchange operator** – replace two edges by two other edges and iterates until no further improvement is possible.
2. **Or-Opt operator** – replace three edges in the original tour by three new ones without modifying the orientation of the new route.
3. **2-opt* Operator** – combine two routes so that the last customers of a given route are introduced after the first customers of another route.
4. **Relocate Operator** – moves a random customer from its current route to a different one.
5. **Exchange Operator** – swaps two customers in different routes.
6. **Ejection-chain** – moves a customer from its current route and insert it into another route.
7. **Geni-Exchange Operator** – a customer is inserted between two customers on the nearest destination route.

5 Experimental Setup

In order to provide a sound foundation for our ant-based hyperheuristic, we have reproduced the original ant algorithm (Ant System). We have tested the algorithm on the Travelling Salesman Problem (TSP) and shown that it is not statistically significantly different from the original algorithm.

5.1 Ant-based Hyperheuristic

Our proposed algorithm is based on the ant algorithm of Dorigo et al. (1996) but suitably changed to cater for the dynamic heuristic information (i.e. CPU time of two heuristics) rather than the static information (distance between two cities) for the TSP implementation. The ant-based hyperheuristic is outlined in the following pseudocode:

1. \( t = 0 \) \( t \) is the journey counter. A journey is equivalent to the number of low level heuristics available in the algorithm.
2. \( \text{counter} = 0 \) user defined counter for how many cycles ant will make a journey.
3. \( \tau_{ij}(t) = c; \Delta \tau_{ij} = 0 \) \( c \) is an initial value of pheromone value given for every edge \((i,j)\)
4. \( \eta_{ij} = 1 \) visibility is initially given value 1
5. Set \( S_k \) and \( S_b \) as null \( S_k \) is each ant’s solution, \( S_b \) is best solution
6. Place \( m \) ants on \( n \) heuristics
7. For \( k = 1 \) to \( m \)
   - Place the starting node of the ants.
8. Repeat until ants complete a journey
   For \( k = 1 \) to \( m \) do
     - Select the next heuristic (node \( j \)) to apply based on pheromone and visibility
     - Move the \( k \)-th ant to heuristic \( j \)
     - Apply heuristic \( j \) to \( S_k \) to obtain new solution \( S'_k \)
     - If \( S'_k < S_b \)
       - \( S_b = S'_k \)
       - Update visibility \( \eta_{ij} \) visibility is updated after a heuristic is applied
     - \( t = t + 1 \)
   end for
9. \( \text{counter} ++ \) a journey is completed
Update pheromone value $\tau_{ij}$

\[ \text{counter} = 0 \]

If (any best solution is in this cycle)

For all ants $S_k = S_b$

Endif

9. If ( $t = t_{\text{max}}$)

Output best solution $S_b$ found

Stop

We will discuss the algorithm in more detail during the conference presentation.

5.2 Initial Solutions

The initial solutions for the CVRP have been generated using the sweep algorithm (Gillet and Miller 1974). The algorithm is briefly described as follows:

Step 1: The polar coordinates of each node are calculated and sorted in increasing order.

Step 2: Starting from an initial node, choose the next node that has the smallest angle. Assign that node to vehicle route $k$. The nodes are clustered such that the total capacity of each route does not exceed the capacity $Q$.

6. Future Work

We are in the process of adapting the original ant algorithm in order to produce a hyperheuristic version. Following this we will implement the heuristics presented in section 4. This will enable us to run a series of experiments to test the effectiveness of the proposed algorithm. At the conference we will present the full algorithmic details as well as the results when run on benchmark CVRP instances.

References