'T' for Tabu and Time Dependent Travel Time

Johan W. Joubert

Department of Industrial and Systems Engineering, University of Pretoria, South Africa
johan.joubert@up.ac.za

Summary. Even in its most basic form, the Vehicle Routing Problem and its variants are notoriously hard to solve. More often artificially intelligent algorithms are employed to provide near-optimal solutions. To be classified as "intelligent", however, a solution strategy should be able to first analyze the environment in which the problem occurs, then solve it, and afterwards reflect on the solution process so as to improve future decision-making. Although reference is made to the entire context of the intelligence research project, this paper reports on the Tabu Search algorithm designed that catered for a problem with multiple soft time windows, a heterogeneous fleet, double scheduling, and time dependent travel time. An adaptive memory procedure was employed, initially populated with good initial feasible solutions, and the algorithm was tested on 60 problems based on established benchmark sets. The complex variant of the Vehicle Routing Problem required between 670 and 4762 seconds on a standard laptop computer, which is considered to be reasonable in the proposed application, and was consistent between different runs with an absolute mean deviation of 3.6%. The contribution is significant as it provides an algorithm that efficiently addresses a complex and practical application of the Vehicle Routing Problem. The algorithm can easily be extended to make use of multiple processors so as to reduce computational time.

1 Introduction

The Vehicle Routing Problem (VRP) has its aim to assign customer deliveries to vehicles, and determining the visiting order of each vehicle. The VRP is a well-researched problem in Operations Research literature. The problem is typically solved in engineering and logistics situations where the distribution of products are concerned – the main objective being the minimization of an objective function, typically distribution cost for individual carriers. In its basic form, the VRP can be described as the problem of assigning optimal delivery or collection routes from a depot to a number of geographically distributed customers, subject to problem-specific constraints. The most basic version of the VRP can be defined with $G = (V,E)$ being a directed

graph where $V = \{v_0, v_1, \ldots, v_n\}$ is a set of vertices with v_0 representing the depot where m identical vehicles, each with capacity Q, are located. The remaining vertices, denoted by $V \backslash \{v_0\}$, represents customers each having a non-negative demand q_i and a non-negative service time s_i [8]. The edge set connecting the vertices is given by $E = \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$. A distance matrix $C = \{c_{ij}\}$ is defined on E. In some contexts, c_{ij} can be interpreted as travel cost or travel distance. Hence, the terms distance, travel cost, and travel time are used interchangeably. The VRP consists of designing a set of m vehicle routes having a minimum total length such that

- each route starts and ends at the depot,
- each remaining vertex ($V \backslash \{v_0\}$) is visited exactly once by one vehicle,
- the total demand of a route does not exceed Q, and
- the total duration (including service and travel time) of a route does not exceed a preset limit L.

The VRP is a hard combinatorial (NP-hard) optimization problem for which several exact and approximate solution methods have been proposed (see Laporte [5] for a review). The area of application is wide, and specific variants of the VRP transform the basic problem to conform to application specific requirements. Additional constraints are introduced to adapt to business-specific requirements, and yields a hard combinatorial problem by restriction. The more complex a problem become, the more favorable the choice of approximate as opposed to exact algorithms due to the computational burden.

Metaheuristics are master strategies which use intelligent decision making techniques to guide algorithms to find global optimal solutions by temporarily allowing moves in the neighborhood that result in solutions with inferior objective function values compared to the incumbent solution. The majority of metaheuristics follow a similar process, although their specific naming conventions, analogies and detailed routines may vary.

Initialization – The process of finding an initial solution. Some metaheuristics, such as the *Genetic Algorithm* performs well with randomly generated initial solutions that need not be feasible, while the *Tabu Search* is highly sensitive to the quality of the initial solution.

Diversification – The mechanism that ensures that the underlying heuristics are searching a diversified neighborhood, and thus not getting trapped within local optima.

Intensification – A mechanism that ensures the heuristic starts *zooming in* towards a single solution. The most promising neighborhoods are identified and those areas of the solution space are searched more thoroughly.

The diversification and intensification can repeat indefinitely, and hence requires a stopping criteria to terminate the metaheuristic [8]. The longer the metaheuristic runs, the higher the probability of converging to the global optimum. The interested reader is referred to [7] for a recent and comprehensive review of various solution strategies. In a comparison of heuristics