Chapter 9

Intelligent routing agents: birth or burial?

The primary research question, as stated in Section 1.3, that this thesis intends to answer is *whether it is feasible to develop a rational and intelligent agent to schedule a predefined variant of the Vehicle Routing Problem (VRP).*

9.1 Answering the research questions

In answering the question affirmatively, research highlights will be reviewed according to a number of secondary research questions stated in terms of the concept of an *intelligent agent.*

9.1.1 Sensory perception

In order to be classified as *intelligent*, an agent must have the ability to perceive its environment. This thesis postulates that benchmark data sets are skewed in the sense that they do not represent reality appropriately. Solution algorithms presented in literature are inherently problem specific, and are therefor lauded to be successful in solving selected benchmark problems.

In reality, however, decision makers do not have a reference to whether a problem's customers are clustered, semi-clustered, or randomly distributed. Fuzzy c-means clustering is used in this thesis to establish the level of clusteredness and level of cluster separation of a given data set, and thus providing the agent with an ability to recognize its environment.

Results of test sets indicated the Xie-Beni validation index to be best suited for measuring and provided noticeably different clustering results for the various problem sets.

9.1.2 Behavior generation

Two prominent metaheuristics, the Tabu Search (TS) and the Genetic Algorithm (GA), was developed to solve a complex variant of the VRP. Soft time windows, a heterogeneous fleet and multiple scheduling was incorporated in an environment with time-dependent travel times imposed on the network.

To ensure good performance by the TS, a good initial solution was required. A sequential route construction heuristic was developed for the complex environment. To ease the computational burden, the novel concept referred to as Time Window Compatibility (TWC) was introduced to eliminate obviously infeasible insertions, and to generate good seed customers. The TS itself incorporated an Adaptive Memory Procedure (AMP) that allowed the algorithm to benefit from the successes of evolutionary metaheuristics. A self-organizational component is introduced to ensure the algorithm is adaptable to changing environments.

During the development of the GA, two new precedence lists were proposed for the Merged Crossover (MX) operator. Results compared favorably in a thorough evaluation of various operators. However, the GA was competitive in neither the quality of the solution, nor the computational time required when measured against the TS.

9.1.3 Value judgement

A fairly standard set of costs, risks, benefits, and or penalties was employed to indicate to the intelligent agent which metaheuristic algorithm yielded solutions of higher quality. The objective function (or fitness function) minimized total time. Time was considered more important than a monetary value as it directly relates to both the time-dependent travel time that resulted in quickest routes, and the expected lateness at customers. The objective function heavily penalized both the number of vehicles in an attempt to improve utilization, and orphaned customers not included in the final solution.

9.1.4 World modeling

To update the agent's knowledge base, a neural network is trained to predict the best solution algorithm to use, given a problem set for which geographic dispersion, time window ratios, and demand characteristics are known.

The overall structure of the intelligent agent is provided in Algorithm 8.1. Once implemented in a real problem environment, the agent will be given an unknown data set with customer locations, time window requirements for each customer, as well as product demands. The customer characteristics are analyzed using the fuzzy c-means clustering, and the cluster results are simulated in the neural network, suggesting the best solution algorithm to use, as well as a predicted objective function value.

The agent uses the suggested solution algorithm to solve the problem at hand, and provide the user with the incumbent routes and schedules. The agent frequently retrains the neural network according to a dynamic update frequency parameter to ensure that the knowledge base is updated, and allowing the agent to adapt to the specific environment.

9.2 Critical observations and recommendations

Although the results from the initial solution algorithm, metaheuristics, as well as the cluster validation proved very successful and useful, some critical observations are valid.

Neural networks require rich data sets to allow the network to identify intricate non-linear relationships between inputs and outputs. The candidate expected the two metaheuristics, TS and GA, to be more competitive with each other. The contrary was however observed with the GA only competing in a single instance. It should be noted that the GA was not tested on all 60 problems due to the computational burden being in excess of 20 times that of the TS algorithm. The candidate therefor reverted to training the network with a data set not representative of both solution algorithms.

Fleischmann et al. (2004) used the rich data available to them trough the Intelligent Transportation System (ITS) in Berlin. The ITS provide real-time data on the congestion and travel speeds of roads, remotely-observed, in the city for different times of the day. Other contributions using real-time data readily available to them include Ghiani et al. (2003) and Giaglis et al. (2004). In the absence of such accurate real-time data, this thesis reverted to benchmark data sets and simulated network congestion, as did Ichoua et al. (2003). On the candidate's research agenda is the integration of the proposed routing agent with current South African initiatives to establish an ITS in selected metropolitan areas.

As the field of metaheuristics is well-researched, the majority of research opportunities sprouting from this thesis is on the topic of learning of an intelligent routing agent.

Learning structure As opposed to employing a neural network to represent the agent's world modeling ability, a Bayesian network could be investigated. Similar to neural networks, the Bayesian tree requires substantial data to be created.

Recurrent network The candidate proposed and tested a feed-forward neural network with back propagation. As an alternative, the recurrent network, or feed-back network, could be considered. It is unfortunately a network structure that is not well-understood and well researched due to its computational complexity. The benefit of the feed-back network structure is the short-term memory inherent in the network. Such an approach may resemble the thought processes of a human decision maker more accurately.

Update frequency Once the network is employed in the problem environment, it retrains itself according to an update frequency function. The function proposed in (8.1) initially retrains itself after every iteration to ensure the algorithm adapts rapidly to the new environment. As the network becomes more stable, the frequency decreases. A more complex update frequency function may be considered in environments that is *not* stable, as assumed in this thesis. The improved update frequency function may either reset the global iteration number, based on some criteria, to again update the network at every iteration. Alternatively, a completely new function may be proposed that dynamically adjusts based on environmental changes to reflect an ad hoc or short-term change in the environment.

Although the computational burden of the algorithms proposed in this thesis is not excessive, with exception of the GA results, significant improvement can be made through parallelization of the solution algorithm across multiple processors. The University of Pretoria has recently invested in *Velocity*: a computer cluster consisting of a master and 24 high specification processor nodes. Similar cluster configurations are becoming more readily available in industry as well, and are often shared by companies. Rochat and Taillard (1995) provide guidelines to extend the TS with AMP across multiple processors. Even the basic GA structure lends itself to parallel processing, and should be investigated for a speed-up in computational time.

A multi-processor environment will also benefit the neural network as one or more processors may be dedicated to actually solving the problem with the solution algorithm suggested by the agent, while other processors are dedicated to constantly retrain the neural network as new problem sets and targets become available.

9.3 Conclusion

The *not-so-artificially-intelligent-vehicle-routing-agent*^{TM} has been proved both feasible and viable through the careful integration of multiple Operations Research techniques, including cluster analysis, optimization and neural networks. With computational capacity becoming more accessible, the agent may become a generally accepted means to intelligently adapt routing algorithms to various, even unique, problem environments.

In establishing the agent's ability to execute solution algorithms, only two prominent metaheuristics were considered. The agent's structure lends itself to be generalized to include a multitude of solution algorithms, and even various similar solution algorithms with different parameter configurations.