Modern Approaches to the Rich Vehicle Routing Problem

Henry F. R. Fellows
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
fello056@morris.umn.edu

ABSTRACT

Henry: Autonomous navigation is cool and important. Routing, being a part of Autonomous navigation, is a problem. It is hard. Good routing saves a lot of money and time.

Keywords

Vehicle Routing Problem, Black-box optimization, Memetic Algorithms, Probability Collectives Henry: & more

1. INTRODUCTION

Henry: Routing is cool and important, also Amazon prime, Uber, self driving cars, and mail. The general case of routing is the traveling salesman problem. The traveling salesman problem asks for the shortest route which passes through each point once. Assuming that each pair of points is connected by a link, the number of potential routes is $0.5n!$, which grows extremely quickly, and is significant even for small values. For $n = 6$, the number of routes reaches $20,160$. The traveling salesman problem is in a class of problems known as NP-complete. Although it has not yet been proven, it is considered likely that there is no algorithm for NP-complete problems that allows them to be solved quickly. Fortunately, solutions of NP-complete problems are easy to verify. In practice, the traveling salesman problem is too abstract for real-world routing problems, and so a variant was proposed that better represents the challenges of routing vehicles.

1.1 Routing in the real world

Henry: Definition of ye olde Vehicle Routing Problem. Also there are a lot of variants; here are the ones I can find good sources about most common ones (a bit of sarcasm; they are actually the most researched ones, at least recently. A variant with dynamic nodes was very popular pre-2006.) Routing in the real world is often a complicated and messy affair. The vehicle has to have enough capacity for all of its deliveries, and often, customers expect that the vehicles arrive near a specified time. In some cases, the customer might not know how much of a product they need until the vehicle arrives. Dealing with these types of problems is the domain of the Vehicle Routing Problem.

2. THE VEHICLE ROUTING PROBLEM

Originally titled the Truck Dispatching Problem, the original formulation of the Vehicle Routing Problem (VRP) was created by G. B. Danzig and J. H. Ramser in 1959[?]. The premise of the problem is that each vehicle (or truck), has a limited capacity and the fleet must make deliveries to as many customers as possible, starting from a specific location known as a depot[?] Alternatively, the vehicles may have to make multiple deliveries in order to satisfy all of the customers. A measure that is also considered worth minimizing is the maximum tour length - the longest single route taken by one truck [?]. Danzig and Ramser note that if the total capacity of the vehicles is less than the total demand of the customers, the problem is mathematically identical to the traveling salesman problem. There has been a great deal of research done on the original VRP, and recent research focuses on versions of the VRP with different constraints or multiple constraints simultaneously. These variations of the Vehicle Routing Problem are collectively referred to as the Rich Vehicle Routing Problem[?] (RVRP). In the following sections of, we discuss prominent variants of the RVRP.

2.1 Decentralized Vehicle Routing Problem

Henry: It's agent flavored! Many problems in the rich vehicle routing category feature decentralized control, where each vehicle or depot is modeled as having an independent agent who makes decisions out of self-interest. However, pure self-interest can cause decisions that degrade the solutions of others. Imagine a vehicle which finds a route that fulfills a large number of deliveries extremely quickly; other vehicles might find that their routes force them to cross through this area where they cannot make any deliveries along the way. While it is highly efficient for the first vehicle, it is less efficient for all other vehicles. The ideal is to pursue optimal routes - both on the local (individual) scale, and on a global scale. Finding other methods of making decisions that allow for both autonomy and global utility is an ongoing research question.

2.2 Vehicle Routing Problem with Time Windows

Henry: Deliveries during business hours only!

2.3 Approaches

UMM CSci Senior Seminar Conference, December 2015 Morris, MN.

In general the approaches used to solve the VRP are approximate methods; they intend to find 'good', not perfect solutions. The algorithms tend to use simple rules to guess solutions, and the process of narrowing down or perfecting these guesses is known as optimization. Henry: History, maybe?. Most modern methods of solving the VRP belong to the blackbox style of optimization; blackbox optimization is used when a problem does not have a formal algebraic model, or the model is too computationally expensive. The VRP does have a very good model, but it is extremely computationally expensive to use it, and so the natural choice is to use blackbox methods. The common feature of blackbox methods is the use of stochastic (random) elements, especially in selecting starting states. This makes blackbox methods into a double edged sword; they are often effective at solving otherwise intractable problems, but the solutions frequently fail to produce any insight into the problem. Stochastic decision making makes it hard to determine why the algorithm provides a specific solution.

Henry: Maybe: The preference for blackbox methods may be a reflection of the difficulty in finding/improving [non-guessing] algorithms (or, perhaps equivalently, of the inherent computational limitations of calculating a perfect answer to the VRP)

3. GENETIC AND MEMETIC ALGORITHMS

Henry: Of Memetic Algorithms: In general, the genetic algorithm improves the solution in large strokes, while the local optimization fine tunes the solutions generated by the GA. GAs are cool too, and the HGSADC is a pretty spiffy algorithm

3.1 Hybrid Genetic Search with Advanced Diversity Control

Adding time window constraints to customer demand and depot availability poses a significant challenge, especially in relation to the smaller proportion of solutions and the increased computational complexity caused by the increased search space. The opposing demands of temporal and spatial constraints is a particularly frustrating problem in the VRPTW.

The Hybrid Genetic Search with Advanced Diversity Control (HGSADC) address some of these challenges in the VRPTW, especially in route-duration constraints. The primary feature of the HGSADC is the different approach to diversity managment in it's population. HGSADC adds diversity to its objectives as a term to be optimized, which allows it to avoid dead-end solutions. The algorithm is considered the current state of the art in the multi-depot vehicle routing problem with time windows[**?**].

3.1.1 *Algorithm and mechanics*

HGSADC is a complex algorithm, and cannot be fully described in this paper. What follows is a summary of the algorithm and the more notable of its unique features.

```
Initialize population;
while number of interactions without improvement
$< lt_{NI}$, and time $< T_{max}$ do
    Select parent solutions $P_1$ and $P_2$;
    Create offspring $C$ from $P_1$ and $P_2$ (crossover);
    Educate $C$ (local search procedure);
    if $C$ infeasible then
        Insert $C$ into infeasible subpopulation;
        Repair with probability $P_r ep$;
    end
    if $C$ feasible then
        Insert $C$ into feasible subpopulation;
    end
    if maximum subpopulation size reached then
        Select survivors;
    end
    if best solution not improved for $It_{div}$ iterations
    then
        Diversify population;
    end
    Adjust penalty parameters for infeasibility;
    if number of iterations $= k \times It_{dec}$ where $k \in \mathbb{N}$
    then
        Henry: k made up of natural numbers?;
        Decompose the master problem;
        Use HGSADC on each subproblem;
        Reconstitute three solutions, and insert them in
        the population;
    end
end
return best feasible solution;
```

Henry: Algorithm-figure goes here eventually. HGSDAC evolves infeasible and feasible solutions as two separate subpopulations. Two parents are selected by Genetic operators Henry: ??? iteratively applied to the population, and are then combined into an offspring. A local search is then applied to the offspring, and then may be repaired Henry: explain or skip and placed into an appropriate subpopulation. If a subpopulation exceeds a maximum size, a survivor selection stage is triggered Henry: explain.... A diversification round may be started if there has been It_{div} iterations without improvement. Henry: to be cont, pending decisions on what to gloss over.

3.2 Adaptive Memetic Algorithms

Henry: This section needs to be replaced as of 3:44 Oct 13.

3.2.1 The actual algorithm

4. AGENT-BASED MODELS.

Agents are simply small decision making elements that typically have some internal state and the ability to make decisions based on this internal state. In optimization, a typical goal is for an agent to attempt to achieve some sort of optimal state. Agents (should) take actions to improve their state, and we describe this as the agent acting in its own interest. The function that returns the quantification of this local optimization is termed the local utility function, in contrast to the global, or system utility function. The global utility function describes how optimal the system is, which can be quite different from the local utility. Henry: Make up definition? Agents are very poorly defined. “A set of agents, with a function f(x) that change their internal state, based on external state. They may change external state as well.” We have chosen to discuss two major approaches in modeling individuals for the decentralized VRP. The first is agent-based models, the more traditional approach, and the second is probability collectives, a relatively recent approach that has a legacy in genetic algorithms.

4.1 Saleh et al. didn’t name their algorithm!!!

Henry: But seriously, why? The closest thing they came up with would be something like distributed reverse Vickrey auction Henry: I named it: “The Distributed RVA Routing Algorithm”

Saleh et al.[**?**] were examining a version of the DCVRP where each depot and customer are treated as agents. The authors were interested in designing a system that encouraged near-optimal solutions while still being completely self-interested. They pursued this by building mechanisms that encouraged depots to offer an accurate estimate of minimum costs while still allowing the cost functions of each depot to remain private.

4.1.1 The Distributed RVA Routing Algorithm

There is a set D of m agent depots; each depot, k, has N_k vehicles with total capacity q_k^i. Each vehicle is denoted $i_k^1 ... i_k^{N_k}$, and assigned to route r_i^k. The number of vehicles, capacity, and the cost function are private to each depot. The set of customers is C, and a single customer is z, which has a demand d_z that can be served only by one vehicle. This information is public to all depots.

For every customer j, we estimate the additional cost that results from adding j to the customers served by depot k, for each depot k such that $q_k^i \geq d_z$. The least additional cost dc_k is obtained using an approximate solution.

Then, find the depot k with the lowest dc_k, and add the customer to r_k, the route indicated by the method of estimating the least additional cost. The routes of depot k can be optimized by applying an exact solution to the VRP to the order of the list of nodes in each route. Finally, the capacity of depot k is updated to the sum of $q_k^i - d_j$.

Henry: Alternatively:

Each depot k submits a bid to each unassigned customer j where $q_k^i \geq d_z$. Bids consist of the estimated costs of servicing the customer. The customer chooses the lowest bid satisfying their demands and notifies k^*. The depot k^* may receive responses from other customers. From the given responses, the depot chooses the customer j^* with the lowest insertion cost (the customer whose addition to the vehicle’s route will cause the least change in the routing cost). Once the customer is assigned to a depot, the customer j^* submits payment to depot k^*. The payment is equal to the second lowest offer of the game. This, combined with the fact that bids are sealed, encourages depots to provide the lowest possible rate (because if they win, they are still paid more than the estimate). The completion of these steps is a single auction round, and many rounds may take place until every customer is assigned. Once the auction rounds are complete, the depot can apply other optimization strategies to it’s routes, or in the case of small routes, an exact solution.

4.2 Probability Collectives

Probability Collectives (PC) takes a different approach to selecting a good solution. Instead of attempting to evolve an exemplary individual in a population in the method of genetic or memetic algorithms, probability collectives selects optimal strategy for each agent[**?**]. A PC agent is a self-interested, learning individual that selects a strategy with the highest probability of optimizing the local and global objectives. In contrast to the Distributed Reverse Vickrey Auction, the agents willingly and freely share information about their strategies, and the cost function is global.

4.2.1 Mechanics of Probability Collectives

The general form of PC in the context of VRP is as follows, along with the flowchart below. Imagine you have N vehicles, each having a set X of strategies, of (potentially variable) length m. The strategies are most often routes in RVRP applications[**?**], but can sometime be as large as entire sets of routes or as small as individual actions. The set is represented as $X_i = \{X_i^{[1]}, X_i^{[2]}, ..., X_i^{[m]}\}$.

A strategy for vehicle i is denoted as $X_i^{[r]}$, where r is an identifier for that strategy. For each vehicle, assign a uniform probability, $1/m_i$ to all actions; the resulting probability of that strategy is signified by $q(X_i^{[r]})$. The vehicle then selects a random action r and a sampling of random actions from other vehicles. The resulting set, the ‘combined strategy set’, $Y_i^{[r]} = \{X_1^{[?]}, ..., X_i^{[r]}, ..., X_N^{[?]}\}$, represents a guess as to a potential future. Accordingly, for each of the strategy sets $Y_i^{[r]}$, compute the expected local utility - the measure of how good the solution is expected to be for the agent - using the following measure:

$$\text{Expected Utility of Agent } i^r = q_i^r \prod_{(i)} q(x_{(i)}^{[?]}) \cdot G(Y_i^{[r]}) \quad (1)$$

Where q_i^r represents the probability of action r for vehicle i, (i) is the set of all agents excluding i, and G is the function that computes global utility for a given set of strategies. G is problem specific, but in the RVRP it could be a measure of unused capacity, unvisited destinations, maximum tour length or various combinations of measurements.

After computing the local utility, the next step is to update the probability of each action for all vehicles as follows:

$$q(X_i^{[r]}) < -q(X_i^{[r]} - \alpha_{Step} \cdot q(X_i^{[r]}) \cdot k_{r\ update} \quad (2)$$

where k is the iteration, and

$$k_{r\ update} = \frac{contrib.\ of\ Agent\ i}{T} + S_i(q) + ln(q(X_i^{[r]})) \quad (3)$$

Here, Boltzmann's temperature T is a scalar that represents the importance of the of the contribution of agent i. Henry: This is only the literal interpretation of T. It might also be $T(k) = T_0/ln(k)$, but I can't find any good reference. The contribution of Agent i is then:

$$contrib.\ Agent\ i = exp.\ utility\ Agent\ i^r - exp.\ global\ utility \quad (4)$$

The $S_i(q)$ term is the entropy of every agent's probability distribution. Entropy in the context of information theory is the expected value of the information in a message. As it increases, the probability distribution more clearly distinguishes the contribution of each strategy toward optimizing the expected global utility. Entropy is computed by:

$$S_i(q) = -\sum_{r=1}^{m_i} q(X_i^{[r]}) \cdot ln(q(X_i^{[r]}) \quad (5)$$

To summarize, if a strategy r creates a larger contribution to the optimization of the objective than other strategies, the probability associated with r increases by a larger amount. This entire process is repeated until the probability distribution converges or the maximum number of iterations are completed. The strategy with the highest probability is then returned.

5. HUMAN ASSISTED ROUTING

Henry: 'Humans are really good at visually identifying good routes and tagging them accordingly. Also, server time is expensive and Amazon's Mechanical Turk is cheap.'

6. CONCLUSION

7. ACKNOWLEDGEMENTS

Henry: I'd like to thank caffeine, water, and stress; the raw elements that formed this paper. Harris L. Mayer's 1964 paper, "Opacity Calculations, Past and Future" was a wonderful source of thematic inspiration.

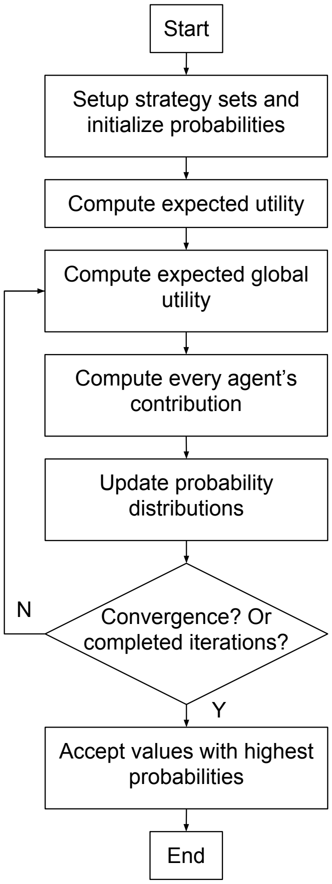
Start
Setup strategy sets and initialize probabilities
Compute expected utility
Compute expected global utility
Compute every agent's contribution
Update probability distributions
Convergence? Or completed iterations?
N
Y
Accept values with highest probabilities
End

Figure 1: Probability Collectives Algorithm. Henry: THIS FIGURE IS GIVING ME AN ULCER.