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

The Rich Vehicle Routing Problem is a class of problems that revolve around finding the most optimal route for a certain set of deliveries. Obtaining approximate solutions to these computationally hard problems is both economically valuable and academically interesting. This paper describes several recent approaches to popular variants of the Rich Vehicle Routing Problem such as Hybrid Genetic Search with Advanced Diversity Control, Probability Collectives, Distributed Reverse Vickrey Auctions, and Interactive Genetic Routing.

Keywords

Vehicle Routing Problem, Black-box optimization, Genetic Algorithms, Probability Collectives, Distributed Systems

1. INTRODUCTION

Routing is a deeply pervasive feature of the modern world. From the point-to-point navigation of everyday driving to the shipping of bulk freight, the task of creating those routes has been increasingly automated. The long term trend towards self-driving vehicles means that computers will soon control the entire process of transportation; aside from selecting a time of departure and destination, humans need no longer be involved. Software to create routes for school buses, mail deliveries and garbage trucks has been in use for decades; everything from Amazon to Delta uses these systems on a daily basis.

The exact impact of improvements in routing is difficult to measure, but in 2011, the external costs of traffic jams in the European Union were 1-2% of GDP [3]. In the United States, 57 tons of goods were moved for each person in 2012 [8]. A small change in routing efficiency has the potential to reduce some of the staggeringly large economic and environmental costs of transportation.

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 $\frac{1}{2}n!$, which grows extremely quickly, and is significant even for small values. For $n = 10$, the number of routes is $1,814,400$, and for $n = 15$, it reaches $653,837,184,000$. 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, and this property encourages approximate solutions; routes that satisfy all customers, but may not be the most optimal means of doing so.

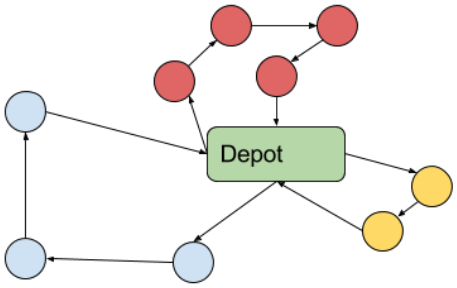

Figure 1: Example of the VRP

Routing in the real world is much more complex than the traveling salesman problem implies. There may be an expectation that the vehicle arrives near a specified time. New stops may be added during the trip, or the exact demand of each customer might not be known ahead of time.

Dealing with these types of problems is the domain of the Rich Vehicle Routing Problem, which extends the Vehicle Routing Problem described in the next section. After that, two major classes of approximate solutions to the Vehicle Routing Problem are discussed along with recent algorithms in these categories: genetic algorithms and agent-based algorithms. Finally, conclusions are drawn about the results of these algorithm.

2. RICH 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 [4]. 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 loca-

UMM CSci Senior Seminar Conference, December 2016 Morris, MN.

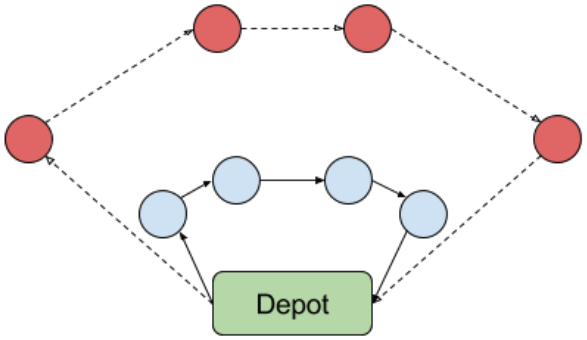

Figure 2: Example of poor global utility in DVRP

tion known as a depot as shown in Figure 1. The vehicles may have to make multiple trips in order to satisfy all of the customers. The primary metric is the number of trips taken to satisfy all customer demand. 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 a vehicle is greater than the total demand of the customers, the problem is mathematically identical to the traveling salesman problem, because then a single truck can serve all customers in a single route. 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 many constraints simultaneously [3]. These variations of the Vehicle Routing Problem are collectively referred to as the Rich Vehicle Routing Problem (RVRP) . In the following sections of, we define prominent variants of the RVRP.

2.1 Decentralized Vehicle Routing Problem

A common problem in rich vehicle routing literature is decentralized control, where each vehicle or depot is modeled as having an independent agent who makes decisions out of self-interest [3]. This is in contrast to the behavior of the traditional VRP, which has a single controlling entity that is aware of the entire state of the problem. The scheduling system that Delta Airlines uses to dispatch flights is an example of a centralized system; however, when the system crashed in august 2016, it forced the cancellation of hundreds of flights. A decentralized system is not as vulnerable to the failure of individual entities, which makes it useful in high-availability systems. In some cases, decentralized routing also makes sense when the vehicles are owned by independent owners that do not wish to maintain their own systems for routing.

A recurring problem in the DRVP and decentralized systems in general is that 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. Figure 2 is an example of this, where the blue (solid line) route 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 good methods of making decisions that allow for both autonomy and global utility is an ongoing research question.

2.2 VRP with Time Windows

Another variant of the VRP is the vehicle routing problem with time window (VRPTW), where the customer expects that the deliveries take place within a certain time interval [3]. Unlike the estimates provided by shipping services, these constraints are set by the customers. A simple example is the constraint that deliveries are only accepted during business hours. This adds the dimension of time to the problem – the time it takes the truck to traverse nodes and to complete the delivery must be considered. The VRPTW is one of the more popular extensions of the VRP because deliveries without implicit or explicit time constraints are rare. In practice, the time window constraint is rarely found without other constraints; industrial settings tend to consider the VRPTW to be the base problem for real-world settings.

2.3 Black Box Optimization

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 refining these guesses is known as optimization. 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 [1]. There is no method to solve the VRP without using the computationally expensive exact solution. The natural choice is to use blackbox methods which trade accuracy for speed. The common feature of these 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.

3. GENETIC ALGORITHMS

A genetic algorithm (GA) is a type of problem-solving that is inspired by natural selection. It is often used to find solutions for processes that are not well understood, but can be modeled well. In the case of the VRP, the problem is both known and modeled well, but the approximate solution given by a well-made genetic is often nearly as good, and much easier to compute than the exact solution.

The fundamental notion of a genetic algorithm is a *genetic representation*, which is a means of representing an individual. Individuals in VRP tend to be routes or sets of routes. By analogy, the representation is the genome of the individual. A common representation is a fixed-length array of bits, but other representations are possible. These genetic representations are manipulated by genetic operators that simulate various processes. A mutation operator provides the analogue to biological mutation by altering values in the genetic representation; for a fixed length array of bit, a common example is a function that flips bits randomly generator. The crossover operator represents reproduction - it combines traits from two (or more) individuals to produce a new individual. Genetic algorithms aim to improve their solutions, and they measure how 'good' a solution is by using a fitness function. Fitness functions are problem

Algorithm 1: Genetic Algorithm Pseudocode

```
Initialize population;
while iteration limit not reached do
    Select best individuals via fitness function;
    Generate new individuals using genetic operators;
    Build new population from old & new individuals;
end
return best solution;
```

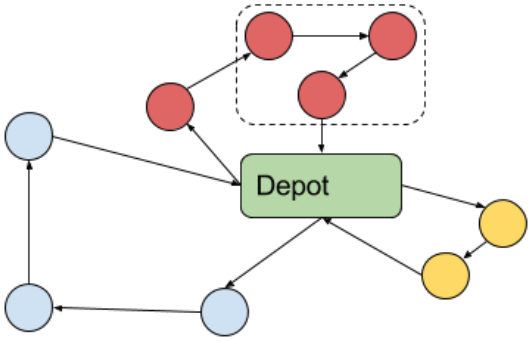

Figure 3: Example of selecting a good region

specific; in VRP, they measure if the individual solves the problem, and how good that solution is.

The general form of a GA begins with initializing the population. A selection of individuals with a random genetic representation is created. As illustrated in algorithm 1, the algorithm enters its main loop; the loop is usually terminated when an iteration limit is reached, but other conditions are sometimes used that are specific to the problem. In the loop, the individuals are scored using the fitness function and the best are set aside for the next stage. The crossover operator is then applied to this group, and mutation operator on the resulting individuals. Then, it evaluates each individual and replaces the worst members of the population with the best new individuals. This process is repeated until the termination condition is reached; this is frequently an iteration limit. The individual with the highest fitness in the population is then returned.

3.1 Interactive Genetic Routing

Humans tend to be very good at solving visual problems, and this extends to the routing if the problem is displayed appropriately. We have a talent for deciding whether a given route is good or bad by simply looking at a diagram. Recently, a number of platforms were created to allow developers to make use of human intelligence in solving computationally complex problems. Amazon's Mechanical Turk, an example of these platforms, allows the exploitation of human intuition by offering an API that makes it possible to integrate into software. Continuing the theme of using humans to do things humans are good at, S. Ismail, F. Legras, and G. Coppin [2] created a genetic algorithm that uses humans to determine how good a particular solution is.

The interactive genetic algorithm, algorithm 2, uses humans to find good sequences of destinations. It begins by initializing the population with a set of random individuals.

Algorithm 2: Interactive Genetic Algorithm

```
Initialize population;
while iteration limit not reached do
    Select best individuals via fitness function;
    Generate new individuals using crossover;
    Humans to tag portions of new individual;
    Mutate individuals based on tagging;
    Build new population from old & new individuals;
end
return best feasible solution;
```

The individuals are sets of routes in this case. While the iteration cap has not been reached, the algorithm first creates a new group of individuals using a standard crossover operator. Then, humans are used to tag good or bad sections of the routes in each individual as shown in Figure 3. A good section should be a segment of the route that is likely to be part of the most optimal route; a bad section is a poor segment of a route. The tags influence the mutation operator; a 'good' section is unlikely to be modified, while a bad section is likely to be modified. This encourages preservation of good sections, while bad sections are removed. Conventional genetic algorithms are unable to gather or act on any information about the details of the individual aside from the overall fitness. The human involvement allows semi-intelligent decisions to be made about what parts of the individual should be changed. After the mutation operator, it evaluates each individual and replaces the worst members of the population with the best new individuals. The algorithm repeats until the iteration limit is reached and the individual with the best routes is returned.

Ismail, Legras, and Coppin don't provide any experimental results in the paper, but note that humans are relatively slow. In practice, humans would be best used in very large scale routing problems where the added time isn't as much of a concern, and the tagging procedure may not be called every iteration.

3.2 Hybrid Genetic Search with Advanced Diversity Control

Adding time window constraints to customer demand and depot availability poses a significant challenge. Time, distance, and speed are now important to the problem, and all of these factors must be accounted for. 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) addresses some of these challenges in the VRPTW, especially in route-duration constraints. The primary feature of the HGSADC is a different approach to diversity management in its population. HGSADC adds diversity to its objectives as a term to be optimized, which allows it to avoid dead-end solutions by recalling earlier solutions that do not encounter the same problem. The algorithm is considered the current state of the art in the multi-depot vehicle routing problem with time windows [11].

3.2.1 Algorithm and mechanics

HGSADC is a complex algorithm, and cannot be fully described in this paper. What follows is a summary of Algorithm 3 and the most notable features.

Algorithm 3: Hybrid Genetic Search with Advanced Diversity Control

```
Initialize populations at 4μ size;
while number of interactions without improvement
 < lt_NI, and time < T_max do
    Select parent solutions P_1 and P_2;
    Create child 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_rep;
    end
    if C feasible then
        Insert C into feasible subpopulation;
    end
    if maximum subpopulation size, μ, 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 modulo It_dec = 0 then
        Decompose the master problem;
        Use HGSADC on each subproblem;
        Reconstitute three solutions, and insert them in
         the population;
    end
end
return best feasible solution;
```

HGSDAC evolves infeasible and feasible solutions as two separate subpopulations. In most genetic algorithms, infeasible ('incorrect') solutions are removed from the population, but in the HGSADC, they are kept in a separate subpopulation. The rationale is that this allows the algorithm to create a reserve of genetic diversity that can be used to get the population out of dead ends. Two random individuals are pulled from the combined population and the best of them is chosen to be a parent. The crossover operator is an applied to two parents, and the resulting child is educated.

The education process is a limited local search that seeks to improve the child; if the child is still infeasible after this procedure, the repair procedure may be called. P_{rep} is the probability that the repair operator is called; repair is a very intensive local search. Finally, the child is placed into an appropriate subpopulation. If a subpopulation exceeds a maximum size, a survivor selection stage is triggered. A number of individuals is removed until the population returns to the nominal population size μ. An individual that is most similar in terms of fitness score and route to another individual is called a clone. Survivor selection iteratively removes a clone until the population size is returned to μ. The purpose of this style of population management is to maintain diversity. A diversification round may be started if there has been It_{div} iterations without improvement. The best third of each population is retained, and adds 4μ randomly created individuals to introduce new genetic material to the populations.

Finally, every It_{dec} iterations, the problem is decomposed into simplified subproblems and the HGSADC is run on those subproblems (without further decomposition). The initial population of subproblem solutions are created from the relevant genetic material of existing individuals in the populations of the original problem. The three best individuals of each subproblem are retained and merged to form three elite individuals who are added to the population of the original problem.

When the algorithm has finally hit the maximum time, the best solution from the feasible population is returned.

3.2.2 HGSADC Results

The HGSADC is extremely good at solving VRPTW. Algorithms are tested by measuring how close to optimal is the best solution the algorithm produced on a number of test data sets. Out of 465 instances of the VRPTW, the HGSADC improved or found the previous best solution 397 times. It strictly improved the best known solution 233 times. For certain types of instances, the HGSADC is the best known algorithm in literature. On the remaining instances, the HGSADC generates solutions of comparable quality to an earlier algorithm by Nagata et al. [7], and is less computationally efficient. However, it does not require as many hand tailored components for each instance, and performs better on problems with tighter time windows and shorter routes. The average standard deviation of the results is $0.2 - 0.4\%$, which shows that the algorithm is highly stable or consistent.

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 and potentially some of the global state. They typically represent and individual in a population. 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. A situation that is locally optimal for an agent, yet globally non-optimal is show in Figure 2. Here we discuss two major approaches in modeling individuals for the decentralized VRP. The first is a distributed reverse Vickrey auction, the more traditional approach which tends to avoid sharing information. Probability collectives, the second approach is a recent development from statistical physics and game theory that shares information.

4.1 Distributed reverse Vickrey auctions

Saleh et al. [9] examined a version of the decentralized VRP where each depot and customer are treated as agents, as opposed to the more common version that treats vehicles 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.

The core feature of the algorithm is a reverse Vickrey auction, which is a sealed bid reverse auction where competitors

offer a estimate of the cost of providing a service. The consumer selects the bidder that has the lowest cost, but pays them the amount of the second lowest bid. This encourages bidders to bid an honest estimate of cost; in the case that they win, they will always make a profit over their bid.

The algorithm is played as a number of rounds. In each round, each depot k submits a bid to each unassigned customer j where demand is less than or equal to remaining capacity. 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 multiple customers. From the given responses, the depot chooses a single 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 round, or the lowest if only one bid is submitted. 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 its routes, or in the case of small routes, an exact solution.

The distributed reverse Vickrey auction is an effective technique of solving the decentralized VRP in a competitive environment where information is not shared. The authors note that the performance of the algorithm is highly dependent on the choice of lower level routing algorithms, but overall, the DRVA is not as existing algorithms.

4.2 Probability Collectives

Probability Collectives (PC) take a different approach to selecting a good solution. Instead of attempting to evolve an exemplary individual in a population like genetic algorithms, probability collectives selects optimal strategy for each agent [6]. 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.

The general form of PC in the context of VRP is as follows, along with the flowchart below. It is important to note this process is repeated for every agent in the system. Imagine you have N agents, which can be depots or trucks, each having a set X of strategies, of (potentially variable) length m. The strategies are most often routes in RVRP applications [10]. X is sampled from the interval ψ_i with upper and lower bounds ψ^u and ψ^l, which is a non-strict subset of all possible strategies. A strategy for agent i is denoted as $X_i^{[r]}$, where r is an identifier for that strategy. The set is represented as $X_i = \{X_i^{[1]}, X_i^{[2]}, ..., X_i^{[m]}\}$.

For each agent, assign a uniform probability, $1/m_i$ to all actions; the resulting probability of that strategy being selected is signified by $q(X_i^{[r]})$. The agents then selects a random action r and a sampling of random actions from other agents. 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 using the following measure:

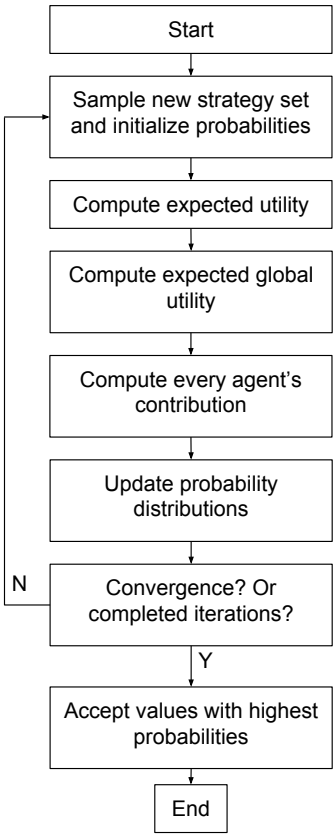

Figure 4: Probability Collectives Algorithm.

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

Here 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 agents as follows:

$$q(X_i^{[r]}) \leftarrow q(X_i^{[r]}) - \alpha_{step} \cdot q(X_i^{[r]}) \cdot O_k^r \quad (2)$$

where k is the iteration, α_{step} is a constant that controls the amount of change each step.

$$O_k^r = \frac{Contrib.\ of\ Agent\ i}{T_k} + S_i(q) + ln(q(X_i^{[r]})) \quad (3)$$

Here, temperature T_k is a scalar that represethe nts the relative importance of the of the contribution of agent i over time. It is computed as $T_{(k+1)} = T_k - \alpha_T \cdot T_k$, and T_0 is the initial temperature. The initial value is unimportant - it must simply be "sufficently high" [5]. α_T controls the rate of temperature change, and the values are often chosen experimentally. The contribution of Agent i is then:

$$Contrib.\ Agent\ i = Utility\ Agent\ i^r - Global\ Utility \quad (4)$$

The $S_i(q)$ term is the entropy of the combined strategy set. In information theory, Entropy is the average value of the information in a message. Probability collectives treats the combined strategy set as a message in order to find the set with the highest Entropy. As it increases, the probability distribution more clearly distinguishes the contribution of each strategy toward optimizing the expected global utility. When it reaches a maximum, it represents the set that has the most information about the probabilities of each strategy. Entropy is computed by:

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

which is the sum of every strategy $q(X_i^{[r]})$ times the natural logarithm of itself. At this point, the algorithm checks to see if it will continue updating the global utility or it will terminate. If the probabilities of the combined strategy set have not changed by a constant amount, σ, or if the number of iterations (k) have reached a maximum, the strategy with the highest probabilities will be returned. Otherwise, the algorithm will continue by narrowing and re-centering the sampling interval ψ_i. The exact process for adjusting the sampling interval can be found in Kulkarni et al.[5].After ψ is narrowed then a new strategy set is sampled, and the process of refining those probabilities is started anew. Once every agent has returned a final strategy, each agent can then pursue those strategies.

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.

4.2.1 Probability Collectives Results

There are no specific results comparing Probability Collectives to other DRVP-solving algorithms. Probability Collectives do have some noteworthy trends; they outperform GA in avoiding dead end solutions and in long term optimization [5]. Vasirani and Ossowski [10] do show that the choice of global utility estimate has a profound effect on how well the algorithm performs. Global utility estimates that do not take the utility of the agent invoking the function are preferred over functions that take all agents into consideration.

5. CONCLUSION

The vehicle routing problem is an important problem in real world logistics, and as in the field of optimization. The most performant algorithm in RVRP is HGSADC, which is dominant within the scope of VRPTW. Adding human intuition or systems emulating it to optimization systems may be important in improving routing systems. In decentralized RVRP variants, the specific constraints of each problem limit the comparability of each algorithm. The distributed reverse Vickrey auction is conservative with the amount of information available to each individual; probability collectives The distributed reverse Vickrey algorithm is within several percent of the best known solutions. Probability collectives is not as good in a straightforward sense, but it handles noisy and imperfect agent behaviors better than other distributed systems. Decentralized means of solving the VRP are not as well-developed as centralized methods, but they show good promise for solving near-future routing problems.

Acknowledgements

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 literary inspiration. Finally, this paper owes a great deal to Nic McPhee and Kirbie Dramdahl for their thoughtful feedback and advice.

6. REFERENCES

[1] S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury. Simulation optimization: a review of algorithms and applications. *4OR*, 12(4):301–333, 2014.

[2] S. Ben Ismail, F. Legras, and G. Coppin. A new interactive evolutionary algorithm for the vehicle routing problem. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '12, pages 661–668, New York, NY, USA, 2012. ACM.

[3] J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan. Rich vehicle routing problem: Survey. *ACM Comput. Surv.*, 47(2):32:1–32:28, Dec. 2014.

[4] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[5] A. J. Kulkarni, A. Abraham, and K. Tai. *Probability collectives*. Springer International Publishing, 1st edition, 2015.

[6] A. J. Kulkarni and K. Tai. Probability collectives for decentralized, distributed optimization: A collective intelligence approach. In *2008 IEEE International Conference on Systems, Man and Cybernetics*, pages 1271–1275, Oct 2008.

[7] Y. Nagata, O. Bräysy, and W. Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 37(4):724 – 737, 2010.

[8] U. S. D. of Transportation. Transportation statistics annual report 2012. www.rita.dot.gov/bts/.

[9] M. Saleh, A. Soeanu, S. Ray, M. Debbabi, J. Berger, and A. Boukhtouta. Mechanism design for decentralized vehicle routing problem. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 749–754, New York, NY, USA, 2012. ACM.

[10] M. Vasirani and S. Ossowski. Decentralized coordination strategies for the vehicle routing problem. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 130–131, New York, NY, USA, 2008. ACM.

[11] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers and Operations Research*, 40(1):475 – 489, 2013.