The Vehicle Routing Problem with Time Window Constraints

Eric McAlpine - ewm1001@sru.edu

Marc Sensenich - mas1029@sru.edu

Aaron Weckerly - ajw1002@sru.edu

Completion

The goal of this project was to implement the heuristics presented by Marius Solomon in “Algorithms for the vehicle routing and scheduling problems with time window constraints.” Based on the criteria defined for the project and the scope of the project this project is considered to be complete. The listing below outlines the heuristics implemented and their completion state:

* Savings Heuristic - Complete
* Savings with Time Windows Heuristic - Complete
* Time-oriented Nearest Neighbor Heuristic - Complete
* Insertion I Heuristic - Complete
* Insertion II Heuristic - Complete
* Insertion III Heuristic - Complete
* Sweep Heuristic - Complete

Abstract

The Vehicle Routing Problem (VRP) is a well-known scheduling problem which seeks to service a number of customers with a fleet of vehicles. The VRP is typically seen as a multiple objective problem; traditionally the objectives include minimizing the total route distance, duration, and vehicles utilized. An extension of the VRP is the Vehicle Routing Problem with Time Window Constraints (VRPTW). In the VRPTW, the objectives from the VRP are included on top of seeking to service all customers on the route within a set arrival and departure time. In our project we implement heuristics presented by Marcus Solomon in his paper “Algorithms for the Vehicle Routing and Scheduling Problem with Time Windows Constraints” in the Zeus framework which provides a mean to run Vehicle Routing Problems.We compare our implementation results against those found by Solomon and the best known results found for the VRPTW.

Problem Explanation

The Vehicle Routing Problem (VRP) is a combinatorial optimization and integer optimization problem that is seeking a solution that services a number of customers with a fleet of vehicles. The VRP seeks out the optimal route to service n customers with m vehicles initially located at a depot. The overall objective of the VRP is to reduce the total travel cost, which is a combination of distance travelled, total travel time, and the number of vehicles utilized. The issue with the traditional VRP is that it does not account for the complexities of the real world. Extensions of the VRP are created by placing additional restraints on the VRP, such as capacity and delivery deadlines. From these additional constraints comes the Vehicle Routing Problem with Time Windows (VRPTW) which extends the Capacitated Vehicle Routing Problem (CVRP).

With the VRPTW each customer now has a window of time in which a delivery must be delivered. This addition of time windows adds additional constraints that must be considered when compared to the VRP. The first constraint that must be in place is that all trucks only have a finite amount of space which can not be exceeded while building a route and the second that is crucial to the VRPTW is that all customers must be serviced before the close of their time window. Time windows also add another aspect to the total cost of the route, waiting time. For example, if the truck were to arrive before the customer’s time window were to open, the truck must wait for the time window to open; this waiting time is considered to be an additional cost. An objective that extends from having waiting time is to minimize the accrued waiting time on a route.

When developing solutions for the VRPTW there are more factors that must be accounted for. In an attempt to minimize time windows is focus on the temporal locality of the customer compared to the spatial locality of the customers. On routes that attempt to minimize the wait times the routes that emerge may not be as elegant as one may expect. In the benchmark data sets that are utilized, there may be customers that are place in cluster, however there may be large time window gaps between customers spatially close to others. In situations such as these one must account for the opportunity cost of waiting for the spatially closest customer over servicing additional customers that be within closer temporal locality.When heuristics are developed for the VRPTW one must account for these locality differences which may accrue more distance cost which has benefits that outweigh the cost of waiting for time windows to open.

Relation to Operating Systems

Vehicle Routing, as stated previously, is a scheduling problem. Scheduling jobs for processor time is a major function of the operating system in any modern, concurrent computer. Even with Moore’s law in place, stating that the number of transistors on a processor die will double around every eighteen months, jobs in need of being scheduled will almost always outnumber the number of processing units.

The time window aspect of the VRP can be compared to a Real-Time Operating System, one in which jobs need accomplished by a deadline. These processes, or customers, will only appear or be able to be serviced at a certain time. In this case, the customers have hard deadlines, which cannot be broken, or else the route, or CPU schedule, is invalid.

Operating Systems have a variety of different algorithms that can be implemented for practically every aspect of the machine, from CPU scheduling, to memory management. Deciding which algorithm works best has to be tested on real-world data on a system similar to the destination product. With the variety of heuristics we implemented, we have created the algorithms, and have been given a multitude of data sets to compare results on. Given this information, we can determine which algorithm works better for which category of data, similar to how an operating system can decide which algorithm works best for different types, or usage levels of systems.

Special Features

With the addition of the VRPTW package, all the classes have been adapted with time window information and have appropriately changed getters, setters, and clone functions. There is also the additional feature of batch processing problem sets. The team implemented a recursive function that will process the values by navigating the file structure and calling the main VRPTW function against each of the datasets

Detailed Report

To accommodate the time window information, all Zeus core classes have been inherited into the VRPTW package. The derived Nodes and Shipment classes contain a new data field for an early arrival time and a late arrival time. These have the appropriate getters and setters. Due to this change, the Attributes class had to be given a wait time data field, which in turn required the cost functions to adapt and contain a getter and setter for waiting time. The cost functions for total cost have been changed to include wait time in their calculation. Feasibility also takes the depot’s maximum allowed time into consideration when deciding if a given route is feasible.

The heuristics included vary on how they handle the time window information. Some rely on the cost functions to handle the time aspect of the problem, while others bring the time window into consideration when routing the shipment. The Savings Heuristics are supposed to create a route for each shipment, then combining them to save on overall cost until there are no feasible or cost-effective combinations. However, due to a miscommunication, it linearly inserts into a truck if it is more cost effective than it being in its own route. The Saving Time-Oriented takes wait time into consideration as discussed in Solomon’s paper, but it also fails to create individual routes to combine. The Insertion Criterion heuristics each take a set of criteria, based on distances, into consideration when assessing a possible route for a shipment insertion. Insertion I bases its criteria around minimizing the increase in distance and time between customers. Insertion II focuses more on minimizing the overall increase of the time and distance of a route. Insertion III build off of Insertion I and also accounts for the urgency of servicing a particular customer.

We have adapted Zeus’ file reading and writing from text files to excel files using Apache POI. Using this, we can input the data more cleanly by referencing a cell instead of knowing where the file pointer is. This also allows for comparison of existing information instead of simply writing over it. Using this, we have an excel sheet that contains our results for every algorithm on each data set, which is checked against the current result every time Zeus is run. Using these, we have Excel automatically calculate our best result for each data set, then Excel compares our best result to both Solomon’s best result and the best known result for each data set.

Caveats/Minefields

Both of the Savings heuristics do not follow the Solomon’s paper as well as they should. They are supposed to place every shipment into its own truck, then check if combining two routes will be cheaper than the existing routes, then repeat until there are no more efficient combinations. The Savings Time-Oriented simply looks more closely at the time windows when determining cost, so that a truck is not spending too much of its time waiting. At the moment, they merely select a shipment and place it in the first route that is cheaper than it being in its own route.

For a user looking to take on a project of this scope there are a few recommendations for knowledge that should be known:

* A thorough understanding of Object-Oriented principles
* A strong base understanding of the Zeus Framework
* A understanding of the objectives, constraints, and complexities of the Vehicle Routing Problem and the Vehicle Routing Problem with Time Windows

Comparison of Results Table

Our best results compared to best results. Best results used can be found at <http://w.cba.neu.edu/~msolomon/heuristi.htm>​

|  |  |  |
| --- | --- | --- |
| **Data** | **Our Best Known Distance** | **Our Best Known Trucks** |
| r101 | 14% | 42% |
| r102 | -9% | 18% |
| r103 | -51% | -46% |
| r104 | -113% | -167% |
| r105 | -21% | 7% |
| r106 | -26% | -25% |
| r107 | -83% | -100% |
| r108 | -137% | -178% |
| r109 | -35% | -27% |
| r110 | -53% | -60% |
| r111 | -39% | -30% |
| r112 | -76% | -100% |
| c101 | -47% | 0% |
| c102 | -53% | 0% |
| c103 | -53% | 0% |
| c104 | -54% | 0% |
| c105 | -46% | 0% |
| c106 | -46% | 0% |
| c107 | -44% | 0% |
| c108 | -45% | 0% |
| c109 | -53% | 0% |
| rc101 | -15% | 0% |
| rc102 | -54% | -67% |
| rc103 | -124% | -127% |
| rc104 | -171% | -200% |
| rc105 | -28% | -31% |
| rc106 | -53% | -55% |
| rc107 | -90% | -82% |
| rc108 | -112% | -110% |
| r201 | 23% | 25% |
| r202 | 24% | -67% |
| r203 | -32% | -133% |
| r204 | -39% | -300% |
| r205 | -1% | 0% |
| r206 | -6% | -67% |
| r207 | -12% | -250% |
| r208 | -58% | -300% |
| r209 | -11% | -33% |
| r210 | 0% | -33% |
| r211 | -2% | -100% |
| c201 | -39% | 0% |
| c202 | -45% | 0% |
| c203 | -35% | 0% |
| c204 | -43% | -33% |
| c205 | -40% | 0% |
| c206 | -40% | 0% |
| c207 | -40% | 0% |
| c208 | -40% | 0% |
| rc201 | 43% | 25% |
| rc202 | 26% | -67% |
| rc203 | -13% | -167% |
| rc204 | -89% | -200% |
| rc205 | 32% | 0% |
| rc206 | 19% | -33% |
| rc207 | 15% | -33% |
| rc208 | -22% | -67% |

Comparison of our average results to the average results found in "Algorithms for the vehicle routing and scheduling problems with time window constraints"

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Set** | **Heuristc** | **Our Avg from Solomon Avg - SCH** | **Our Avg from Solomon Avg - Dist** | **Our Avg from Solomon Avg - WT** | **Our Avg from Solomon Avg - Truck** |
| C1 | SAV | -32% | -79% | -240% | 15% |
| C1 | SWT | -26% | -190% | -180% | -248% |
| C1 | I1 | -19% | -32% | -1101% | 0% |
| C1 | I2 | -26% | -60% | -1620% | 0% |
| C1 | I3 | -24% | -38% | -2846% | 0% |
| C1 | NN | -36% | -71% | -989% | 2% |
| C1 | S | -22% | -35% | -987% | 0% |
| R1 | SAV | -110% | -66% | -397% | -58% |
| R1 | SWT | -123% | -142% | -358% | -288% |
| R1 | I1 | -84% | -47% | -611% | -79% |
| R1 | I2 | -120% | -95% | -759% | -104% |
| R1 | I3 | -71% | -26% | -789% | -65% |
| R1 | NN | -102% | -88% | -440% | -61% |
| R1 | S | -66% | -20% | -490% | -17% |
| RC1 | I1 | -129% | -91% | -1182% | -129% |
| RC1 | I2 | -170% | -157% | -1416% | -180% |
| RC1 | I3 | -113% | -70% | -1277% | -129% |
| RC1 | NN | -134% | -123% | -740% | -96% |
| RC1 | S | -79% | -33% | -638% | -38% |
| R2 | I1 | -117% | -27% | -1934% | -65% |
| R2 | I2 | -174% | 9% | -2553% | -118% |
| R2 | I3 | -119% | -19% | -1713% | -58% |
| R2 | NN | -188% | 65% | -1683% | -122% |
| R2 | S | -135% | -8% | -2549% | -79% |
| C2 | I1 | -22% | -81% | -700% | -223% |
| C2 | I2 | -26% | -82% | -832% | -197% |
| C2 | I3 | -25% | -42% | -4509% | -186% |
| C2 | NN | -32% | -108% | -299% | -186% |
| C2 | S | -27% | -78% | -4736% | -233% |
| RC2 | I1 | -82% | -39% | -1123% | -38% |
| RC2 | I2 | -114% | -11% | -1135% | -77% |
| RC2 | I3 | -83% | -26% | -927% | -47% |
| RC2 | NN | -164% | 63% | -1101% | -115% |
| RC2 | S | -117% | -2% | -1306% | -66% |

References

Azi, Nabila, Michel Gendreau, and Jean-Yves Potvin. "An exact algorithm for a single-vehicle routing problem with time windows and multiple routes." European journal of operational research 178.3 (2007): 755-766.

El-Sherbeny, Nasser A. "Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods." Journal of King Saud University-Science 22.3 (2010): 123-131.

Golden, Bruce L., and Arjang A. Assad. "Vehicle routing with time-window constraints." American Journal of Mathematical and Management Sciences 6.3-4 (1986): 251-260.

Kontoravdis, George, and Jonathan F. Bard. "A GRASP for the vehicle routing problem with time windows." ORSA journal on Computing 7.1 (1995): 10-23.

Savelsbergh, Martin WP. "The vehicle routing problem with time windows: Minimizing route duration." ORSA journal on computing 4.2 (1992): 146-154.

Solomon, Marius M. "Algorithms for the vehicle routing and scheduling problems with time window constraints." Operations research 35.2 (1987): 254-265.